

Content

1. Welcome ..... 1

2. Introduction to Internet Direct ..... 2

    2.1. Introduction to Internet Direct ..... 2

    2.2. Internet Direct Credits ..... 2

    2.3. Internet Direct Sponsors ..... 4

    2.4. Technical Support ..... 5

    2.5. What Is Internet Direct? ..... 6

3. Symbol Reference ..... 8

    3.1. Classes ..... 8

        3.1.1. EldAcceptWaitCannotBeModifiedWhileServerIsActive ..... 8

        3.1.2. EldAlreadyConnected ..... 8

        3.1.3. EldCanNotChangeTarget ..... 8

        3.1.4. EldCanNotCreateMessagePart ..... 9

        3.1.5. EldClosedSocket ..... 9

        3.1.6. EldConnClosedGracefully ..... 9

        3.1.7. EldCorruptServicesFile ..... 10

        3.1.8. EldCouldNotBindSocket ..... 10

        3.1.9. EldDnsResolverError ..... 10

        3.1.10. EldEldTunnelConnectToMasterFailed ..... 11

        3.1.11. EldException ..... 11

        3.1.12. EldFailedToRetreiveTimeZoneInfo ..... 11

        3.1.13. EldFTPFileAlreadyExists ..... 12

        3.1.14. EldHTTPCannotSwitchSessionStateWhenActive ..... 12

        3.1.15. EldHTTPErrorParsingCommand ..... 12

        3.1.16. EldHTTPHeaderAlreadyWritten ..... 13

        3.1.17. EldHTTPServerError ..... 13

        3.1.18. EldHTTPUnsupportedAuthorisationScheme ..... 13

3.1.19. EldIcmpException ..... 14

3.1.20. EldInterceptPropInvalid ..... 14

3.1.21. EldInterceptProplsNil ..... 14

3.1.22. EldInvalidServiceName ..... 15

3.1.23. EldInvalidSocket ..... 15

3.1.24. EldLoginException ..... 15

3.1.25. EldMaxLoginAttempt ..... 16

3.1.26. EldMessageException ..... 16

3.1.27. EldMoreThanOneTldAntiFreeze ..... 16

3.1.28. EldNNTTPConnectionRefused ..... 17

3.1.29. EldNNTPException ..... 17

3.1.30. EldNNTPNoOnNewGroupsList ..... 17

3.1.31. EldNNTPNoOnNewNewsList ..... 18

3.1.32. EldNNTPNoOnNewsgroupList ..... 18

3.1.33. EldNNTPStringListNotInitialized ..... 18

3.1.34. EldNoBindingsSpecified ..... 19

3.1.35. EldNoDataToRead ..... 19

3.1.36. EldNoExecuteSpecified ..... 19

3.1.37. EldNoOnAuthentication ..... 20

3.1.38. EldNotAllBytesSent ..... 20

3.1.39. EldNotEnoughDataInBuffer ..... 20

3.1.40. EldObjectTypeNotSupported ..... 21

3.1.41. EldOpenSSLError ..... 21

3.1.42. EldOpenSSLLoadError ..... 21

3.1.43. EldOSSLAcceptError ..... 22

3.1.44. EldOSSLConnectError ..... 22

3.1.45. EldOSSLCouldNotLoadSSLLibrary ..... 22

3.1.46. EldOSSLCreatingContextError ..... 23

3.1.47. EldOSSLDataBindingError ..... 23

3.1.48. EldOSSLGetMethodError ..... 23

3.1.49. EldOSSLLoadingCertError ..... 24

3.1.50. EldOSSLLoadingKeyError ..... 24

3.1.51. EldOSSLLoadingRootCertError ..... 24

3.1.52. EldOSSLMoDeNotSet ..... 25

3.1.53. EldOSSLSettingCipherError ..... 25

3.1.54. EldPackageSizeTooBig ..... 25

3.1.55. EldProtocolReplyError ..... 26

3.1.56. EldResponseError ..... 26

3.1.57. EldSetSizeExceeded ..... 26

3.1.58. EldSilentException ..... 26

3.1.59. EldSocketError ..... 27

3.1.60. EldSocketHandleError ..... 27

3.1.61. EldSocksAuthError ..... 27

3.1.62. EldSocksAuthMethodError ..... 28

3.1.63. EldSocksError ..... 28

3.1.64. EldSocksRequestFailed ..... 29

3.1.65. EldSocksRequestIdentFailed ..... 29

3.1.66. EldSocksRequestServerFailed ..... 29

3.1.67. EldSocksServerAddressError ..... 30

3.1.68. EldSocksServerCommandError ..... 30

3.1.69. EldSocksServerConnectionRefusedError ..... 30

3.1.70. EldSocksServerGeneralError ..... 31

3.1.71. EldSocksServerHostUnreachableError ..... 31

3.1.72. EldSocksServerNetUnreachableError ..... 31

3.1.73. EldSocksServerPermissionError ..... 32

3.1.74. EldSocksServerRespondError ..... 32

3.1.75. EldSocksServerTTLExpiredError ..... 32

3.1.76. EldSocksUnknownError ..... 33

3.1.77. EldStackCanNotLoadWinsock ..... 33

3.1.78. EldStackError ..... 33

3.1.79. EldStackInitializationFailed ..... 34

3.1.80. EldStackSetSizeExceeded ..... 34

3.1.81. EldTableNotFound ..... 34

3.1.82. EldTCPConnectionError ..... 34

3.1.83. EldTCPServerError ..... 35

3.1.84. EldTelnetClientConnectError ..... 35

3.1.85. EldTelnetError ..... 36

3.1.86. EldTelnetServerException ..... 36

3.1.87. EldTelnetServerOnDataAvailableIsNil ..... 36

3.1.88. EldTextInvalidCount ..... 36

3.1.89. EldTFTPAccessViolation ..... 37

3.1.90. EldTFTPAllocationExceeded ..... 37

3.1.91. EldTFTPException ..... 37

3.1.92. EldTFTPFileAlreadyExists ..... 38

3.1.93. EldTFTPFileNotFound ..... 38

3.1.94. EldTFTPIllegalOperation ..... 38

3.1.95. EldTFTPNoSuchUser ..... 38

3.1.96. EldTFTPOptionNegotiationFailed ..... 39

3.1.97. EldTFTPUnknownTransferID ..... 39

3.1.98. EldThreadClassNotSpecified ..... 39

3.1.99. EldThreadMgrError ..... 40

3.1.100. EldTunnelConnectToMasterFailed ..... 40

3.1.101. EldTunnelCRCFailed ..... 40

3.1.102. EldTunnelCustomMessageInterpretationFailure ..... 41

3.1.103. EldTunnelDontAllowConnections ..... 41

3.1.104. EldTunnelException ..... 41

3.1.105. EldTunnelInterpretationOfMessageFailed ..... 42

3.1.106. EldTunnelMessageHandlingFailed ..... 42

3.1.107. EldTunnelMessageTypeRecognitionError ..... 42

3.1.108. EldTunnelTransformError ..... 43

3.1.109. EldTunnelTransformErrorBeforeSend ..... 43

3.1.110. EldUDPException ..... 43

3.1.111. EldUDPReceiveErrorZeroBytes ..... 44

3.1.112. EldUDPServerException ..... 44

3.1.113. MClientThread ..... 44

3.1.114. TARRecord ..... 45

3.1.115. TClientData ..... 45

3.1.116. THInfoRecord ..... 45

3.1.117. Tid3To4Coder ..... 46

3.1.118. TidAntiFreeze ..... 46

3.1.119. TidAntiFreezeBase ..... 47

3.1.120. TidASCIICoder ..... 47

3.1.121. TidAttachment ..... 47

3.1.122. TidBase64Decoder ..... 48

3.1.123. TidBase64Encoder ..... 48

3.1.124. TldBaseComponent ..... 51

3.1.125. TldBuffer ..... 51

3.1.126. TldCardAddressItem ..... 52

3.1.127. TldCardPhoneNumber ..... 52

3.1.128. TldChargenServer ..... 52

3.1.129. TldCoder ..... 52

3.1.130. TldCoderCollection ..... 53

3.1.131. TldCoderCRC16 ..... 53

3.1.132. TldCoderItem ..... 54

3.1.133. TldCoderMD2 ..... 54

3.1.134. TldCoderMD4 ..... 55

3.1.135. TldCoderMD5 ..... 55

3.1.136. TldComponent ..... 57

3.1.137. TldConnectionIntercept ..... 57

3.1.138. TldConnectionInterceptOpenSSL ..... 57

3.1.139. TldCookie ..... 58

3.1.140. TldCookieCollection ..... 58

3.1.141. TldDateTimeStamp ..... 59

3.1.142. TldDayTime ..... 59

3.1.143. TldDayTimeServer ..... 60

3.1.144. TldDICTServer ..... 60

3.1.145. TldDISCARDServer ..... 61

3.1.146. TldDNSHeader ..... 61

3.1.147. TldDNSQuestionList ..... 62

3.1.148. TldDNSResolver ..... 62

3.1.149. TldDNSResourceItem ..... 63

3.1.150. TldDNSResourceList ..... 63

3.1.151. TldEcho ..... 63

3.1.152. TldECHOSErver ..... 64

3.1.153. TldEMailAddressItem ..... 64

3.1.154. TldEMailAddressList ..... 65

3.1.155. TldFinger ..... 65

3.1.156. TldFingerServer ..... 66

3.1.157. TldFTP ..... 66

3.1.158. TldGopher ..... 66

3.1.159. TldGopherMenu ..... 67

3.1.160. TldGopherMenuItem ..... 68

3.1.161. TldGopherServer ..... 68

3.1.162. TldHeaderInfo ..... 69

3.1.163. TldHeaderList ..... 69

3.1.164. TldHostNameServer ..... 70

3.1.165. TldHTTP ..... 70

3.1.166. TldHTTPRequestInfo ..... 71

3.1.167. TldHTTPResponseInfo ..... 71

3.1.168. TldHTTPServer ..... 72

3.1.169. TldHTTPSession ..... 72

3.1.170. TldHTTPSessionList ..... 73

3.1.171. TldIcmpClient ..... 73

3.1.172. TldIMAP4Server ..... 74

3.1.173. TldIMFDecoder ..... 74

3.1.174. TldIMFUUDecoder ..... 75

3.1.175. TldIPWatch ..... 75

3.1.176. TldIPWatchThread ..... 76

3.1.177. TldIRCServer ..... 76

3.1.178. TldListenerThread ..... 78

3.1.179. TldLogBase ..... 79

3.1.180. TldLogDebug ..... 79

3.1.181. TldMappedPortTCP ..... 80

3.1.182. TldMappedPortTCPData ..... 80

3.1.183. TldMessage ..... 80

3.1.184. TldMessageClient ..... 81

3.1.185. TldMessagePart ..... 82

3.1.186. TldMessageParts ..... 82

3.1.187. TldMimeTable ..... 82

3.1.188. TldNetworkCalculator ..... 83

3.1.189. TldNNTP ..... 83

3.1.190. TldNNTPServer ..... 84

3.1.191. TldPeerThread ..... 85

3.1.192. TldPOP3 ..... 85

3.1.193. TldQOTD ..... 86

3.1.194. TIdQOTDServer ..... 86

3.1.195. TIdQuotedPrintableDecoder ..... 87

3.1.196. TIdQuotedPrintableEncoder ..... 87

3.1.197. TIdRawBase ..... 88

3.1.198. TIdRawClient ..... 89

3.1.199. TIdServerIntercept ..... 89

3.1.200. TIdServerInterceptOpenSSL ..... 90

3.1.201. TIdSimpleServer ..... 90

3.1.202. TIdSMTP ..... 91

3.1.203. TIdSNTP ..... 91

3.1.204. TIdSocketHandle ..... 92

3.1.205. TIdSocketHandles ..... 92

3.1.206. TIdSSLConnectionIntercept ..... 93

3.1.207. TIdSSLContext ..... 93

3.1.208. TIdSSLOptions ..... 93

3.1.209. TIdSSLServerIntercept ..... 94

3.1.210. TIdSSLSocket ..... 94

3.1.211. TIdStack ..... 95

3.1.212. TIdStackVersion ..... 95

3.1.213. TIdStackVersionWinsock ..... 95

3.1.214. TIdStackWinsock ..... 96

3.1.215. TIdTCPCClient ..... 96

3.1.216. TIdTCPConnection ..... 96

3.1.217. TIdTCPServer ..... 97

3.1.218. TIdTCPServerConnection ..... 97

3.1.219. TIdTelnet ..... 98

3.1.220. TIdTelnetReadThread ..... 98

3.1.221. TIdTelnetServer ..... 99

3.1.222. TIdText ..... 100

3.1.223. TIdThread ..... 100

3.1.224. TIdThreadMgr ..... 101

3.1.225. TIdThreadMgrDefault ..... 101

3.1.226. TIdThreadMgrPool ..... 102

3.1.227. TIdTime ..... 103

3.1.228. TIdTimeServer ..... 103

3.1.229. TIdTrivialFTP ..... 104

3.1.230. TIdTrivialFTPServer ..... 105

3.1.231. TIdTunnelMaster ..... 105

3.1.232. TIdTunnelSlave ..... 106

3.1.233. TIdUDPBase ..... 107

3.1.234. TIdUDPClient ..... 108

3.1.235. TIdUDPListenerThread ..... 108

3.1.236. TIdUDPServer ..... 109

3.1.237. TIdURI ..... 109

3.1.238. TIdUUDecoder ..... 109

3.1.239. TIdUUEncoder ..... 110

3.1.240. TIdVCard ..... 111

3.1.241. TIdVCardAddresses ..... 112

3.1.242. TIdVCardBusinessInfo ..... 112

3.1.243. TIdVCardEMailAddresses ..... 112

3.1.244. TIdVCardEMailItem ..... 113

3.1.245. TIdVCardEmbeddedObject ..... 113

3.1.246. TIdVCardGeog ..... 113

3.1.247. TIdVCardMailingLabelItem ..... 114

3.1.248. TIdVCardMailingLabels ..... 114

3.1.249. TIdVCardName ..... 114

3.1.250. TIdVCardTelephones ..... 114

3.1.251. TIdWhois ..... 115

3.1.252. TIdWholsServer ..... 115

3.1.253. TIdX509 ..... 115

3.1.254. TIdX509Name ..... 116

3.1.255. TIdXXDecoder ..... 116

3.1.256. TIdXXEncoder ..... 116

3.1.257. TIpProperty ..... 118

3.1.258. TLogger ..... 119

3.1.259. TMInfoRecord ..... 119

3.1.260. TMRecord ..... 119

3.1.261. TMXRecord ..... 120

3.1.262. TNameRecord ..... 120

3.1.263. TPTRRecord ..... 120

3.1.264. TQuestionItem .....	121
3.1.265. TReceiver .....	121
3.1.266. TSender .....	121
3.1.267. TSlaveData .....	122
3.1.268. TSlaveThread .....	122
3.1.269. TSOARecord .....	123
3.1.270. TSocksInfo .....	123
3.1.271. TTelnetData .....	123
3.1.272. TWKSRecord .....	124
3.2. Records / Structs .....	124
3.2.1. _TRANSMIT_FILE_BUFFERS .....	124
3.2.2. CardinalRec .....	124
3.2.3. HiLoBytes .....	125
3.2.4. HiLoWords .....	125
3.2.5. hostent .....	125
3.2.6. in_addr .....	126
3.2.7. linger .....	126
3.2.8. netent .....	127
3.2.9. protoent .....	127
3.2.10. servent .....	127
3.2.11. sockaddr_in .....	128
3.2.12. sockproto .....	128
3.2.13. SunB .....	128
3.2.14. SunW .....	129
3.2.15. TByteArray .....	129
3.2.16. TEVP_MD .....	130
3.2.17. TFDSet .....	130
3.2.18. THInfo .....	130
3.2.19. TIdArpHdr .....	131
3.2.20. TIdCardinalBytes .....	131
3.2.21. TIdDnsHdr .....	132
3.2.22. TIdEtherAddr .....	132
3.2.23. TIdEthernetHdr .....	132
3.2.24. TIdHeader .....	133

3.2.25. TIdIcmpEcho .....	133
3.2.26. TIdIcmpFrag .....	134
3.2.27. TIdIcmpHdr .....	134
3.2.28. TIdIcmpTs .....	134
3.2.29. TIdIgmpHdr .....	135
3.2.30. TIdInAddr .....	135
3.2.31. TIdIpHdr .....	135
3.2.32. TIdIpOptions .....	136
3.2.33. TIdRipHdr .....	136
3.2.34. TIdSocksRequest .....	137
3.2.35. TIdSocksResponse .....	137
3.2.36. TIdSunB .....	138
3.2.37. TIdSunW .....	138
3.2.38. TIdTcpHdr .....	139
3.2.39. TIdTcpOptions .....	139
3.2.40. TIdUdpHdr .....	139
3.2.41. timeval .....	140
3.2.42. TIMFCoderUsage .....	140
3.2.43. TipStruct .....	140
3.2.44. TLinger .....	141
3.2.45. TLR .....	142
3.2.46. TMInfo .....	142
3.2.47. TMX .....	143
3.2.48. TNTPGram .....	143
3.2.49. TPeerInfo .....	155
3.2.50. TQWord .....	155
3.2.51. TRdata .....	156
3.2.52. TReplyStatus .....	157
3.2.53. TSOA .....	157
3.2.54. TULong .....	158
3.2.55. TWKS .....	159
3.2.56. TWorkInfo .....	161
3.2.57. WordRec .....	161
3.3. Functions .....	162

3.3.1. AnsiSameText ..... 162

3.3.2. Base64Encode ..... 162

3.3.3. CommaSeperatedToStringList ..... 163

3.3.4. CopyFileTo ..... 163

3.3.5. CurrentProcessId ..... 164

3.3.6. DateTimeToGmtOffSetStr ..... 164

3.3.7. DateTimeToInternetStr ..... 165

3.3.8. DebugOutput ..... 166

3.3.9. Decode2022JP ..... 166

3.3.10. DecodeAddress ..... 167

3.3.11. DecodeAddresses ..... 167

3.3.12. DecodeHeader ..... 168

3.3.13. Encode2022JP ..... 168

3.3.14. EncodeAddress ..... 169

3.3.15. EncodeAddressItem ..... 169

3.3.16. EncodeHeader ..... 170

3.3.17. FD\_CLR ..... 171

3.3.18. FD\_ISSET ..... 171

3.3.19. FD\_SET ..... 172

3.3.20. FD\_ZERO ..... 172

3.3.21. Fetch ..... 172

3.3.22. FileSizeByName ..... 173

3.3.23. FreeAndNil ..... 173

3.3.24. GetMIMETypeFromFile ..... 174

3.3.25. GetQClassStr ..... 174

3.3.26. GetQTypeStr ..... 175

3.3.27. GetSystemLocale ..... 175

3.3.28. GetTickCount ..... 176

3.3.29. GmtOffsetStrToDateTime ..... 177

3.3.30. GMTToLocalDateTime ..... 177

3.3.31. IdPorts ..... 178

3.3.32. IdRawBuildArp ..... 178

3.3.33. IdRawBuildDns ..... 179

3.3.34. IdRawBuildEthernet ..... 179

3.3.35. IdRawBuildIcmpEcho ..... 179

3.3.36. IdRawBuildIcmpMask ..... 179

3.3.37. IdRawBuildIcmpRedirect ..... 180

3.3.38. IdRawBuildIcmpTimeExceed ..... 180

3.3.39. IdRawBuildIcmpTimestamp ..... 180

3.3.40. IdRawBuildIcmpUnreach ..... 180

3.3.41. IdRawBuildIcmp ..... 181

3.3.42. IdRawBuildIp ..... 181

3.3.43. IdRawBuildRip ..... 181

3.3.44. IdRawBuildTcp ..... 181

3.3.45. IdRawBuildUdp ..... 182

3.3.46. IncludeTrailingBackSlash ..... 182

3.3.47. IncQWord ..... 182

3.3.48. InfoCallback ..... 183

3.3.49. InitializeMime ..... 183

3.3.50. InMainThread ..... 184

3.3.51. IntToBin ..... 184

3.3.52. IsCurrentThread ..... 185

3.3.53. IsNumeric ..... 185

3.3.54. LoadWinsock ..... 186

3.3.55. LogicalAnd ..... 186

3.3.56. MakeAckPkt ..... 187

3.3.57. MakeTempFilename ..... 187

3.3.58. Max ..... 188

3.3.59. Min ..... 188

3.3.60. OffsetFromUTC ..... 189

3.3.61. ParseNewsGroup ..... 189

3.3.62. ParseURI ..... 190

3.3.63. ParseXOVER ..... 191

3.3.64. PasswordCallback ..... 192

3.3.65. PosInStrArray ..... 193

3.3.66. RegisterCoderClass ..... 194

3.3.67. ReturnMIMEType ..... 195

3.3.68. RightStr ..... 195

3.3.69. ROL ..... 196

3.3.70. ROR ..... 196

3.3.71. RPos .....	197
3.3.72. SendError .....	198
3.3.73. SendError .....	199
3.3.74. SendError .....	199
3.3.75. SendError .....	200
3.3.76. SetLocalTime .....	201
3.3.77. SetThreadPriority .....	202
3.3.78. Sleep .....	202
3.3.79. StrInternetToDateTime .....	203
3.3.80. StrToCard .....	203
3.3.81. StrToDay .....	204
3.3.82. StrToMonth .....	204
3.3.83. StrToWord .....	205
3.3.84. TimeZoneBias .....	205
3.3.85. UnloadWinsock .....	206
3.3.86. UpCaseFirst .....	206
3.3.87. URLDecode .....	206
3.3.88. URLEncode .....	207
3.3.89. VerifyCallback .....	208
3.3.90. WinsockLoaded .....	208
3.3.91. WordToStr .....	208
3.3.92. WSAGetAsyncBuflen .....	209
3.3.93. WSAGetAsyncError .....	209
3.3.94. WSAGetSelectError .....	209
3.3.95. WSAGetSelectEvent .....	209
3.3.96. WSAMakeSelectReply .....	210
3.3.97. WSAMakeSyncReply .....	210
3.4. Types .....	210
3.4.1. CIdCoder .....	210
3.4.2. CSET .....	210
3.4.3. PFDSet .....	211
3.4.4. PHostEnt .....	211
3.4.5. PIdArpHdr .....	211
3.4.6. PIdBase64Decoder .....	211

3.4.7. PIdBase64Encoder .....	211
3.4.8. PIdCoder .....	212
3.4.9. PIdCoderItem .....	212
3.4.10. PIdDnsHdr .....	212
3.4.11. PIdEthernetHdr .....	212
3.4.12. PIdIcmpEcho .....	213
3.4.13. PIdIcmpFrag .....	213
3.4.14. PIdIcmpHdr .....	213
3.4.15. PIdIcmpTs .....	213
3.4.16. PIdIgmPHdr .....	213
3.4.17. PIdInAddr .....	214
3.4.18. PIdIpHdr .....	214
3.4.19. PIdRipHdr .....	214
3.4.20. PIdTcpHdr .....	214
3.4.21. PIdUdpHdr .....	214
3.4.22. PIdUUDecoder .....	215
3.4.23. PIdUUEncoder .....	215
3.4.24. PIdXXDecoder .....	215
3.4.25. PIdXXEncoder .....	215
3.4.26. PIMFCoderUsage .....	215
3.4.27. PInAddr .....	216
3.4.28. PLinger .....	216
3.4.29. PNetEnt .....	216
3.4.30. PProtoEnt .....	216
3.4.31. PServEnt .....	216
3.4.32. PSOCKADDR .....	217
3.4.33. PSockAddrIn .....	217
3.4.34. PSockProto .....	217
3.4.35. PTimeVal .....	217
3.4.36. PTransmitFileBuffers .....	217
3.4.37. T__WSAFDIsSetProc .....	218
3.4.38. T128BitRecord .....	218
3.4.39. T160BitRecord .....	218
3.4.40. T16x4LongWordRecord .....	218
3.4.41. T384BitRecord .....	218

3.4.42. T4x4LongWordRecord ..... 219

3.4.43. T4x4x4LongWordRecord ..... 219

3.4.44. T64BitRecord ..... 219

3.4.45. TAcceptExProc ..... 219

3.4.46. TAcceptProc ..... 220

3.4.47. TAccessFileEvent ..... 220

3.4.48. TAuthenticationEvent ..... 220

3.4.49. TAuthenticationType ..... 221

3.4.50. TBeforeClientConnectEvent ..... 221

3.4.51. TBindProc ..... 222

3.4.52. TCallbackEvent ..... 222

3.4.53. TCharBuf ..... 222

3.4.54. TCharSet ..... 223

3.4.55. TClassIdException ..... 223

3.4.56. TClientEvent ..... 223

3.4.57. TClosesocketProc ..... 224

3.4.58. TCommandEvent ..... 224

3.4.59. TConnectionResult ..... 225

3.4.60. TConnectProc ..... 225

3.4.61. TDataEvent ..... 225

3.4.62. TDays ..... 226

3.4.63. TDoByIdEvent ..... 226

3.4.64. TDoByNoEvent ..... 226

3.4.65. TEventNewNewsList ..... 227

3.4.66. TEventNewsgroupList ..... 227

3.4.67. TEventStreaming ..... 228

3.4.68. TGetAcceptExSockaddrsProc ..... 228

3.4.69. TGetEvent ..... 228

3.4.70. TGetHostByAddrProc ..... 229

3.4.71. TGetHostByNameProc ..... 229

3.4.72. TGetHostNameProc ..... 229

3.4.73. TGetPeerNameProc ..... 229

3.4.74. TGetProtoByNameProc ..... 230

3.4.75. TgetProtoByNumberProc ..... 230

3.4.76. TGetServByNameProc ..... 230

3.4.77. TGetServByPortProc ..... 230

3.4.78. TGetSockNameProc ..... 230

3.4.79. TGetSockOptProc ..... 231

3.4.80. TGroupEvent ..... 231

3.4.81. THostEnt ..... 231

3.4.82. THostNameGetEvent ..... 231

3.4.83. THostNameOneParmEvent ..... 232

3.4.84. THtonIProc ..... 232

3.4.85. THtonsProc ..... 233

3.4.86. TICMPDataBuffer ..... 233

3.4.87. TIdCardAddressAttributes ..... 233

3.4.88. TIdDICTAuthEvent ..... 234

3.4.89. TIdDICTDefineEvent ..... 234

3.4.90. TIdDICTGetEvent ..... 235

3.4.91. TIdDICTMatchEvent ..... 235

3.4.92. TIdDICTOtherEvent ..... 236

3.4.93. TIdDICTShowEvent ..... 236

3.4.94. TIdExceptionEvent ..... 237

3.4.95. TIdFingerGetEvent ..... 237

3.4.96. TIdFTPTransferType ..... 238

3.4.97. TIdGopherMenuEvent ..... 238

3.4.98. TIdHTTPGetEvent ..... 238

3.4.99. TIdHTTPMethod ..... 239

3.4.100. TIdHTTPOnRedirectEvent ..... 239

3.4.101. TIdHTTPOtherEvent ..... 240

3.4.102. TIdHTTPProtocolVersion ..... 240

3.4.103. TIdIrcFiveParmEvent ..... 241

3.4.104. TIdIrcGetEvent ..... 241

3.4.105. TIdIrcOneParmEvent ..... 241

3.4.106. TIdIrcOtherEvent ..... 242

3.4.107. TIdIrcServerEvent ..... 242

3.4.108. TIdIrcThreeParmEvent ..... 242

3.4.109. TIdIrcTwoParmEvent ..... 243

3.4.110. TIdIrcUserEvent ..... 243

3.4.111. TIdLinger ..... 243



3.4.112. TIdLogDebugTarget ..... 243

3.4.113. TIdMessageEvent ..... 244

3.4.114. TIdMessagePartClass ..... 244

3.4.115. TIdMessagePriority ..... 244

3.4.116. TIdNetTime ..... 245

3.4.117. TIdPhoneAttributes ..... 245

3.4.118. TIdPID ..... 246

3.4.119. TIdQOTDGetEvent ..... 246

3.4.120. TIdServeFile ..... 247

3.4.121. TIdServerThreadEvent ..... 247

3.4.122. TIdSSLAction ..... 247

3.4.123. TIdSSLErrorMode ..... 248

3.4.124. TIdSSLMode ..... 248

3.4.125. TIdSSLVerifyMode ..... 248

3.4.126. TIdSSLVerifyModeSet ..... 248

3.4.127. TIdSSLVersion ..... 249

3.4.128. TIdStackSocketHandle ..... 249

3.4.129. TIdStatisticsOperation ..... 249

3.4.130. TIdStatus ..... 249

3.4.131. TIdStatusEvent ..... 250

3.4.132. TIdStringMessageEvent ..... 251

3.4.133. TIdTelnetNegotiateEvent ..... 251

3.4.134. TIdTFTPMode ..... 251

3.4.135. TIdThreadClass ..... 252

3.4.136. TIdThreadStopMode ..... 252

3.4.137. TIdVCardEmailType ..... 253

3.4.138. TInAddr ..... 253

3.4.139. TInet\_AddrProc ..... 253

3.4.140. TInet\_NtoaProc ..... 254

3.4.141. TIntStringEvent ..... 254

3.4.142. TIoctlSocketProc ..... 254

3.4.143. TListenProc ..... 254

3.4.144. TLogItemEvent ..... 255

3.4.145. TModeSetResult ..... 255

3.4.146. TModeType ..... 256

3.4.147. TMonths ..... 256

3.4.148. TNetEnt ..... 257

3.4.149. TNetworkClass ..... 257

3.4.150. TNewsEvent ..... 258

3.4.151. TNewsTransportEvent ..... 258

3.4.152. TNohlProc ..... 258

3.4.153. TNohsProc ..... 259

3.4.154. TOnGetMessagePartStream ..... 259

3.4.155. TOnReplyEvent ..... 259

3.4.156. TOnSessionEndEvent ..... 259

3.4.157. TOnSessionStartEvent ..... 260

3.4.158. TOnTelnetCommand ..... 260

3.4.159. TOtherEvent ..... 261

3.4.160. TPasswordEvent ..... 261

3.4.161. TPlusRequestEvent ..... 261

3.4.162. TPosProc ..... 262

3.4.163. TProceduralEvent ..... 262

3.4.164. TProtoEnt ..... 262

3.4.165. TRANSMIT\_FILE\_BUFFERS ..... 263

3.4.166. TRecvFromProc ..... 263

3.4.167. TRecvProc ..... 263

3.4.168. TReplyStatusTypes ..... 263

3.4.169. TRequestedRecord ..... 264

3.4.170. TRequestedRecords ..... 264

3.4.171. TRequestEvent ..... 264

3.4.172. TSelectProc ..... 265

3.4.173. TSendMsgEvent ..... 265

3.4.174. TSendMsgEventC ..... 265

3.4.175. TSendProc ..... 265

3.4.176. TSendToProc ..... 265

3.4.177. TSendTrnEvent ..... 266

3.4.178. TSendTrnEventC ..... 266

3.4.179. TServEnt ..... 266

3.4.180. TSetSockOptProc ..... 266

3.4.181. TShutDownProc ..... 267

3.4.182. TSocketAddr ..... 267

3.4.183. TSocket ..... 267

3.4.184. TSocketProc ..... 267

3.4.185. TSocketProto ..... 267

3.4.186. TSocksAuthentication ..... 268

3.4.187. TSocksVersion ..... 268

3.4.188. TStringEvent ..... 268

3.4.189. TTelnetCommand ..... 269

3.4.190. TThreadPriority ..... 269

3.4.191. TTimeVal ..... 269

3.4.192. TTnDataAvail ..... 270

3.4.193. TTnState ..... 270

3.4.194. TTransfer ..... 271

3.4.195. TTransferCompleteEvent ..... 271

3.4.196. TTransmitFileBuffers ..... 271

3.4.197. TTransmitFileProc ..... 272

3.4.198. TTunnelEvent ..... 272

3.4.199. TTunnelEventC ..... 272

3.4.200. TUDPReadEvent ..... 272

3.4.201. TVerifyPeerEvent ..... 273

3.4.202. TWKSBits ..... 273

3.4.203. TWorkBeginEvent ..... 273

3.4.204. TWorkEndEvent ..... 274

3.4.205. TWorkEvent ..... 274

3.4.206. TWorkMode ..... 274

3.4.207. TWSAAsyncGetHostByAddrProc ..... 275

3.4.208. TWSAAsyncGetHostByNameProc ..... 275

3.4.209. TWSAAsyncGetProtoByNameProc ..... 275

3.4.210. TWSAAsyncGetProtoByNumberProc ..... 276

3.4.211. TWSAAsyncGetServByNameProc ..... 276

3.4.212. TWSAAsyncGetServByPortProc ..... 276

3.4.213. TWSAAsyncSelectProc ..... 276

3.4.214. TWSACancelAsyncRequestProc ..... 276

3.4.215. TWSACancelBlockingCallProc ..... 277

3.4.216. TWSACleanupProc ..... 277

3.4.217. TWSADData ..... 277

3.4.218. TWSAGetLastErrorProc ..... 277

3.4.219. TWSAIsBlockingProc ..... 277

3.4.220. TWSARecvExProc ..... 278

3.4.221. TWSASetBlockingHookProc ..... 278

3.4.222. TWSASetLastErrorProc ..... 278

3.4.223. TWSAStartupProc ..... 278

3.4.224. TWSAUnhookBlockingHookProc ..... 278

3.4.225. u\_char ..... 279

3.4.226. u\_int ..... 279

3.4.227. u\_long ..... 279

3.4.228. u\_short ..... 279

3.4.229. WordStr ..... 279

3.5. Variables ..... 280

3.5.1. \_\_WSAFDIsSet ..... 280

3.5.2. Accept ..... 280

3.5.3. AcceptEx ..... 280

3.5.4. Bind ..... 280

3.5.5. CloseSocket ..... 281

3.5.6. CoderCollective ..... 281

3.5.7. Connect ..... 281

3.5.8. GAntiFreeze ..... 281

3.5.9. GetAcceptExSockaddrs ..... 282

3.5.10. GetHostByAddr ..... 282

3.5.11. GetHostByName ..... 282

3.5.12. GetHostName ..... 283

3.5.13. GetPeerName ..... 283

3.5.14. GetProtoByname ..... 283

3.5.15. GetProtoByNumber ..... 283

3.5.16. GetServByName ..... 283

3.5.17. GetServByPort ..... 284

3.5.18. GetSockName ..... 284

3.5.19. GetSockOpt ..... 284

3.5.20. GOffsetFromUTC ..... 284

3.5.21. GServeFileProc ..... 285

3.5.22. GStack ..... 285

3.5.23. GSystemLocale ..... 285

3.5.24. GTimeZoneBias ..... 286

3.5.25. Htonl ..... 286

3.5.26. Htons ..... 286

3.5.27. Id\_SO\_False ..... 286

3.5.28. Id\_SO\_True ..... 287

3.5.29. IndyPos ..... 287

3.5.30. Inet\_Addr ..... 287

3.5.31. Inet\_Ntoa ..... 287

3.5.32. IoctlSocket ..... 288

3.5.33. Listen ..... 288

3.5.34. MIMEMediaType ..... 288

3.5.35. Ntohl ..... 288

3.5.36. Ntohs ..... 289

3.5.37. Recv ..... 289

3.5.38. RecvFrom ..... 289

3.5.39. Select ..... 289

3.5.40. Send ..... 289

3.5.41. SendTo ..... 290

3.5.42. SetSockOpt ..... 290

3.5.43. ShutDown ..... 290

3.5.44. Socket ..... 290

3.5.45. TransmitFile ..... 291

3.5.46. WSAAsyncGetHostByAddr ..... 291

3.5.47. WSAAsyncGetHostByName ..... 291

3.5.48. WSAAsyncGetProtoByName ..... 291

3.5.49. WSAAsyncGetProtoByNumber ..... 292

3.5.50. WSAAsyncGetServByName ..... 292

3.5.51. WSAAsyncGetServByPort ..... 292

3.5.52. WSAAsyncSelect ..... 292

3.5.53. WSACancelAsyncRequest ..... 293

3.5.54. WSACancelBlockingCall ..... 293

3.5.55. WSACleanup ..... 293

3.5.56. WSAGetLastError ..... 293

3.5.57. WSAsIsBlocking ..... 294

3.5.58. WSARecvEx ..... 294

3.5.59. WSASetBlockingHook ..... 294

3.5.60. WSASetLastError ..... 294

3.5.61. WSASStartup ..... 295

3.5.62. WSAUnhookBlockingHook ..... 295

3.6. Constants ..... 295

3.6.1. AF\_APPLETALK ..... 295

3.6.2. AF\_BAN ..... 295

3.6.3. AF\_CCITT ..... 296

3.6.4. AF\_CHAOS ..... 296

3.6.5. AF\_DATAKIT ..... 296

3.6.6. AF\_DECnet ..... 296

3.6.7. AF\_DLI ..... 296

3.6.8. AF\_ECMA ..... 297

3.6.9. AF\_FIREFOX ..... 297

3.6.10. AF\_HYLINK ..... 297

3.6.11. AF\_IMPLINK ..... 297

3.6.12. AF\_INET ..... 298

3.6.13. AF\_IPX ..... 298

3.6.14. AF\_ISO ..... 298

3.6.15. AF\_LAT ..... 298

3.6.16. AF\_MAX ..... 298

3.6.17. AF\_NETBIOS ..... 299

3.6.18. AF\_NS ..... 299

3.6.19. AF\_OSI ..... 299

3.6.20. AF\_PUP ..... 299

3.6.21. AF\_SNA ..... 300

3.6.22. AF\_UNIX ..... 300

3.6.23. AF\_UNKNOWN1 ..... 300

3.6.24. AF\_UNSPEC ..... 300

3.6.25. AF\_VOICEVIEW ..... 300

3.6.26. BACKSPACE ..... 301

3.6.27. base64\_tbl ..... 301

3.6.28. Base64CodeTable ..... 301

3.6.29. BUFFERLEN ..... 302

3.6.30. BytesReadType ..... 302

3.6.31. BytesWriteType ..... 302

3.6.32. cA ..... 303

3.6.33. cAABit ..... 303

3.6.34. cAAMask ..... 303

3.6.35. cAXFR ..... 303

3.6.36. cCH ..... 304

3.6.37. cCS ..... 304

3.6.38. CHAR0 ..... 304

3.6.39. CHAR32 ..... 305

3.6.40. cHINFO ..... 305

3.6.41. cHS ..... 305

3.6.42. cIN ..... 305

3.6.43. cMAILA ..... 306

3.6.44. cMAILB ..... 306

3.6.45. cMB ..... 306

3.6.46. cMD ..... 306

3.6.47. cMF ..... 307

3.6.48. cMG ..... 307

3.6.49. cMINFO ..... 307

3.6.50. cMR ..... 308

3.6.51. cMX ..... 308

3.6.52. CN\_CODED\_DATA ..... 308

3.6.53. CN\_CODING\_ENDED ..... 308

3.6.54. CN\_CODING\_STARTED ..... 309

3.6.55. CN\_DATA\_END\_FOUND ..... 309

3.6.56. CN\_DATA\_START\_FOUND ..... 309

3.6.57. CN\_IMF\_BODY\_PART\_END ..... 310

3.6.58. CN\_IMF\_BODY\_START ..... 310

3.6.59. CN\_IMF\_CODER\_START ..... 310

3.6.60. CN\_IMF\_DATA\_END ..... 311

3.6.61. CN\_IMF\_END\_MULTIPART ..... 311

3.6.62. CN\_IMF\_HEAD\_VALUE ..... 311

3.6.63. CN\_IMF\_NEW\_FILENAME ..... 311

3.6.64. CN\_IMF\_NEW\_MULTIPART ..... 312

3.6.65. CN\_NEW\_FILENAME ..... 312

3.6.66. CN\_UU\_BEGIN\_FOUND ..... 312

3.6.67. CN\_UU\_CODER\_START ..... 313

3.6.68. CN\_UU\_END\_FOUND ..... 313

3.6.69. CN\_UU\_LAST\_CHAR\_FOUND ..... 313

3.6.70. CN\_UU\_NEW\_FILENAME ..... 313

3.6.71. CN\_UU\_PRIVILEGE\_ERROR ..... 314

3.6.72. CN\_UU\_PRIVILEGE\_FOUND ..... 314

3.6.73. CN\_UU\_TABLE\_BEGIN\_ABORT ..... 314

3.6.74. CN\_UU\_TABLE\_CHANGED ..... 315

3.6.75. CN\_UU\_TABLE\_FOUND ..... 315

3.6.76. cName ..... 315

3.6.77. cNS ..... 315

3.6.78. cNULL ..... 316

3.6.79. CompressedBytesType ..... 316

3.6.80. CompressionRatioType ..... 316

3.6.81. ConstBoundary ..... 317

3.6.82. ConstContentDisposition ..... 317

3.6.83. ConstContentMD5 ..... 317

3.6.84. ConstContentTransferEncoding ..... 317

3.6.85. ConstContentType ..... 318

3.6.86. ConstFileName ..... 318

3.6.87. ConstIMFBoundaryEnd ..... 318

3.6.88. ConstIMFMessageStart ..... 319

3.6.89. ConstIMFStart ..... 319

3.6.90. ConstName ..... 319

3.6.91. cOpCodeBits ..... 319

3.6.92. cOpCodeMask ..... 320

3.6.93. cOPCodeStrs ..... 320

3.6.94. CP\_FALLBACK ..... 320

3.6.95. CP\_IMF ..... 321

3.6.96. CP\_STANDARD ..... 321

3.6.97. cPTR ..... 321

3.6.98. cQClassStr ..... 321

3.6.99. cQRBit ..... 322

3.6.100. cQRMask ..... 322

3.6.101. CR ..... 322

3.6.102. cRABit ..... 323

3.6.103. cRAMask ..... 323

3.6.104. cRCodeBits ..... 323

3.6.105. cRCodeFormatErr ..... 323

3.6.106. cRCodeMask ..... 324

3.6.107. cRCodeNameErr ..... 324

3.6.108. cRCodeNoError ..... 324

3.6.109. cRCodeNotImplemented ..... 325

3.6.110. cRCodeRefused ..... 325

3.6.111. cRCodeServerErr ..... 325

3.6.112. cRCodeStrs ..... 325

3.6.113. cRDBit ..... 326

3.6.114. cRDMask ..... 326

3.6.115. cReslQuery ..... 326

3.6.116. cResQuery ..... 327

3.6.117. cResStatus ..... 327

3.6.118. cSOA ..... 327

3.6.119. csSPECIALS ..... 328

3.6.120. cStar ..... 328

3.6.121. CT\_Creation ..... 328

3.6.122. CT\_Realisation ..... 328

3.6.123. cTCBit ..... 329

3.6.124. cTCMask ..... 329

3.6.125. CTL3To4 ..... 329

3.6.126. cTXT ..... 330

3.6.127. cWKS ..... 330

3.6.128. DEF\_PACKET\_SIZE ..... 330

3.6.129. EADDRINUSE ..... 330

3.6.130. EADDRNOTAVAIL ..... 331

3.6.131. EAFNOSUPPORT ..... 331

3.6.132. EALREADY ..... 331

3.6.133. ECONNABORTED ..... 331

3.6.134. ECONNREFUSED ..... 331

3.6.135. ECONNRESET ..... 332

3.6.136. EDESTADDRREQ ..... 332

3.6.137. EDQUOT ..... 332

3.6.138. EHOSTDOWN ..... 332

3.6.139. EHOSTUNREACH ..... 333

3.6.140. EINPROGRESS ..... 333

3.6.141. EISCONN ..... 333

3.6.142. ELOOP ..... 333

3.6.143. EMSGSIZE ..... 333

3.6.144. ENAMETOOLONG ..... 334

3.6.145. ENETDOWN ..... 334

3.6.146. ENETRESET ..... 334

3.6.147. ENETUNREACH ..... 334

3.6.148. ENOBUFS ..... 335

3.6.149. ENOPROTOOPT ..... 335

3.6.150. ENOTCONN ..... 335

3.6.151. ENOTEMPTY ..... 335

3.6.152. ENOTSOCK ..... 335

3.6.153. EOL ..... 336

3.6.154. EOPNOTSUPP ..... 336

3.6.155. EPFNOSUPPORT ..... 336

3.6.156. EPROCLIM ..... 336

3.6.157. EPROTONOSUPPORT ..... 337

3.6.158. EPROTOTYPE ..... 337

3.6.159. EREMOTE ..... 337

3.6.160. ErrAccessViolation ..... 337

3.6.161. ErrAllocationExceeded ..... 338

3.6.162. ErrFileAlreadyExists ..... 338

3.6.163. ErrFileNotFound ..... 338

3.6.164. ErrIllegalOperation ..... 338

3.6.165. ErrNoSuchUser ..... 339

3.6.166. ErrOptionNegotiationFailed ..... 339

3.6.167. ErrUndefined ..... 339

3.6.168. ErrUnknownTransferID ..... 340

3.6.169. ESHUTDOWN ..... 340

3.6.170. ESOCKTNOSUPPORT ..... 340

3.6.171. ESTALE ..... 340

3.6.172. ETIMEDOUT ..... 341

3.6.173. ETOOMANYREFS ..... 341

3.6.174. EUSERS ..... 341

3.6.175. EWouldBlock ..... 341

3.6.176. FD\_ACCEPT ..... 341

3.6.177. FD\_CLOSE ..... 342

3.6.178. FD\_CONNECT ..... 342

3.6.179. FD\_OOB ..... 342

3.6.180. FD\_READ ..... 342

3.6.181. FD\_SETSIZE ..... 343

3.6.182. FD\_WRITE ..... 343

3.6.183. FIOASYNC ..... 343

3.6.184. FIONBIO ..... 343

3.6.185. FIONREAD ..... 344

3.6.186. GContentType ..... 344

3.6.187. GFContentLength ..... 344

3.6.188. GFMaxAge ..... 345

3.6.189. GFRequestedBlockSize ..... 345

3.6.190. GFTTL ..... 345

3.6.191. GLoginAttempts ..... 345

3.6.192. GPathSep ..... 346

3.6.193. GReceiveTimeout ..... 346

3.6.194. GRecvBufferSizeDefault ..... 346

3.6.195. GResponseNo ..... 347

3.6.196. GSendBufferSizeDefault ..... 347

3.6.197. GServerSoftware ..... 347

3.6.198. gsldProductName ..... 348

3.6.199. gsldVersion ..... 348

3.6.200. GTransferMode ..... 348

3.6.201. HalfCodeTable ..... 348

3.6.202. hdrsize ..... 349

3.6.203. HOST\_NOT\_FOUND ..... 349

3.6.204. ICMP\_MIN ..... 349

3.6.205. Id\_ARP\_HSIZE ..... 350

3.6.206. Id\_ARPHRD\_ETHER ..... 350

3.6.207. Id\_ARPOP\_INVREPLY ..... 350

3.6.208. Id\_ARPOP\_INVREQUEST ..... 350

3.6.209. Id\_ARPOP\_REPLY ..... 350

3.6.210. Id\_ARPOP\_REQUEST ..... 351

3.6.211. Id\_ARPOP\_REVREPLY ..... 351

3.6.212. Id\_ARPOP\_REVREQUEST ..... 351

3.6.213. ID\_Default\_TIdAntiFreezeBase\_Active ..... 351

3.6.214. ID\_Default\_TIdAntiFreezeBase\_ApplicationHasPriority ..... 352

3.6.215. ID\_Default\_TIdAntiFreezeBase\_IdleTimeOut ..... 352

3.6.216. ID\_Default\_TIdAntiFreezeBase\_OnlyWhenIdle ..... 352

3.6.217. Id\_DNS\_HSIZE ..... 353

3.6.218. Id\_ETHER\_HSIZE ..... 353

3.6.219. Id\_ETHER\_ADDR\_LEN ..... 353

3.6.220. Id\_ETHERTYPE\_ARP ..... 353

3.6.221. Id\_ETHERTYPE\_IP ..... 353

3.6.222. Id\_ETHERTYPE\_LOOPBACK ..... 354

3.6.223. Id\_ETHERTYPE\_PUP ..... 354

3.6.224. Id\_ETHERTYPE\_REVARP ..... 354

3.6.225. Id\_ETHERTYPE\_VLAN ..... 354

3.6.226. Id\_ICMP\_ECHO ..... 355

3.6.227. Id\_ICMP\_ECHO\_HSIZE ..... 355

3.6.228. Id\_ICMP\_ECHOREPLY ..... 355

3.6.229. Id\_ICMP\_HSIZE ..... 355

3.6.230. Id\_ICMP\_IREQ ..... 355

3.6.231. Id\_ICMP\_IREQREPLY ..... 356

3.6.232. Id\_ICMP\_MASK\_HSIZE ..... 356

3.6.233. Id\_ICMP\_MASKREPLY ..... 356

3.6.234. Id\_ICMP\_MASKREQ ..... 356

3.6.235. Id\_ICMP\_PARAMPROB ..... 357

3.6.236. Id\_ICMP\_PARAMPROB\_OPTABSENT ..... 357

3.6.237. Id\_ICMP\_REDIRECT ..... 357

3.6.238. Id\_ICMP\_REDIRECT\_HOST ..... 357

3.6.239. Id\_ICMP\_REDIRECT\_HSIZE ..... 357

3.6.240. Id\_ICMP\_REDIRECT\_NET ..... 358

3.6.241. Id\_ICMP\_REDIRECT\_TOSHOST ..... 358

3.6.242. Id\_ICMP\_REDIRECT\_TOSNET ..... 358

3.6.243. Id\_ICMP\_ROUTERADVERT ..... 358

3.6.244. Id\_ICMP\_ROUTERSOLICIT ..... 359

3.6.245. Id\_ICMP\_SOURCEQUENCH ..... 359

3.6.246. Id\_ICMP\_TIMXCEED\_HSIZE ..... 359

3.6.247. Id\_ICMP\_TIMXCEED ..... 359

3.6.248. Id\_ICMP\_TIMXCEED\_INTRANS ..... 359

3.6.249. Id\_ICMP\_TIMXCEED\_REASS ..... 360

3.6.250. Id\_ICMP\_TS\_HSIZE ..... 360

3.6.251. Id\_ICMP\_TSTAMP ..... 360

3.6.252. Id\_ICMP\_TSTAMPREPLY ..... 360

3.6.253. Id\_ICMP\_UNREACH ..... 361

3.6.254. Id\_ICMP\_UNREACH\_FILTER\_PROHIB ..... 361

3.6.255. Id\_ICMP\_UNREACH\_HOST ..... 361

3.6.256. Id\_ICMP\_UNREACH\_HOST\_PRECEDENCE ..... 361

3.6.257. Id\_ICMP\_UNREACH\_HOST\_PROHIB ..... 361

3.6.258. Id\_ICMP\_UNREACH\_HOST\_UNKNOWN ..... 362

3.6.259. Id\_ICMP\_UNREACH\_HSIZE ..... 362

3.6.260. Id\_ICMP\_UNREACH\_ISOLATED ..... 362

3.6.261. Id\_ICMP\_UNREACH\_NEEDFRAG ..... 362

3.6.262. Id\_ICMP\_UNREACH\_NET ..... 363

3.6.263. Id\_ICMP\_UNREACH\_NET\_PROHIB ..... 363

3.6.264. Id\_ICMP\_UNREACH\_NET\_UNKNOWN ..... 363

3.6.265. Id\_ICMP\_UNREACH\_PORT ..... 363

3.6.266. Id\_ICMP\_UNREACH\_PRECEDENCE\_CUTOFF ..... 363

3.6.267. Id\_ICMP\_UNREACH\_PROTOCOL ..... 364

3.6.268. Id\_ICMP\_UNREACH\_SRCFAIL ..... 364

3.6.269. Id\_ICMP\_UNREACH\_TOSHOST ..... 364

3.6.270. Id\_ICMP\_UNREACH\_TOSNET ..... 364

3.6.271. Id\_IGMP\_HSIZE ..... 365

3.6.272. Id\_IGMP\_LEAVE\_GROUP ..... 365

3.6.273. Id\_IGMP\_MEMBERSHIP\_QUERY ..... 365

3.6.274. Id\_IGMP\_V1\_MEMBERSHIP\_REPORT ..... 365

3.6.275. Id\_IGMP\_V2\_MEMBERSHIP\_REPORT ..... 365

3.6.276. Id\_INADDR\_ANY ..... 366

3.6.277. Id\_INADDR\_NONE ..... 366

3.6.278. Id\_INVALID\_SOCKET ..... 366

3.6.279. Id\_IP\_DF ..... 366

3.6.280. Id\_IP\_HSIZE ..... 367

3.6.281. Id\_IP\_MAXPACKET ..... 367

3.6.282. Id\_IP\_MF ..... 367

3.6.283. Id\_IP\_OFFMASK ..... 367

3.6.284. Id\_IP\_RF ..... 367

3.6.285. Id\_IP\_TTL ..... 368

3.6.286. Id\_IPPROTO\_ICMP ..... 368

3.6.287. Id\_IPPROTO\_IGMP ..... 368

3.6.288. Id\_IPPROTO\_IP ..... 368

3.6.289. Id\_IPPROTO\_MAX ..... 369

3.6.290. Id\_IPPROTO\_RAW ..... 369

3.6.291. Id\_IPPROTO\_TCP ..... 369

3.6.292. Id\_IPPROTO\_UDP ..... 369

3.6.293. ID\_LOGBASE\_Active ..... 369

3.6.294. ID\_LOGBASE\_LogTime ..... 370

3.6.295. ID\_MAPPED\_PORT\_TCP\_PORT ..... 370

3.6.296. Id\_MAX\_IPOPTLEN ..... 370

3.6.297. ID\_MSG\_NODECODE ..... 371

3.6.298. ID\_MSG\_PRIORITY ..... 371

3.6.299. ID\_MSG\_USENOWFORDATE ..... 371

3.6.300. ID\_NC\_MASK\_LENGTH ..... 372

3.6.301. ID\_NETWORKCLASS ..... 372

3.6.302. Id\_PF\_INET ..... 372

3.6.303. Id\_RIP\_HSIZE ..... 373

3.6.304. Id\_RIPCMD\_MAX ..... 373

3.6.305. Id\_RIPCMD\_POLL ..... 373

3.6.306. Id\_RIPCMD\_POLLENTRY ..... 373

3.6.307. Id\_RIPCMD\_REQUEST ..... 373

3.6.308. Id\_RIPCMD\_RESPONSE ..... 374

3.6.309. Id\_RIPCMD\_TRACEOFF ..... 374

3.6.310. Id\_RIPCMD\_TRACEON ..... 374

3.6.311. Id\_RIPVER\_0 ..... 374

3.6.312. Id\_RIPVER\_1 ..... 375

3.6.313. Id\_RIPVER\_2 ..... 375

3.6.314. Id\_SD\_Both ..... 375

3.6.315. Id\_SD\_Recv ..... 375

3.6.316. Id\_SD\_Send ..... 375

3.6.317. ID\_SIMPLE\_SERVER\_BOUND\_PORT ..... 376

3.6.318. Id\_SO\_BROADCAST ..... 376

3.6.319. Id\_SO\_DEBUG ..... 376

3.6.320. Id\_SO\_DONTROUTE ..... 376

3.6.321. Id\_SO\_KEEPAIVE ..... 377

3.6.322. Id\_SO\_LINGER ..... 377

3.6.323. Id\_SO\_OOBLINE ..... 377

3.6.324. Id\_SO\_RCVBUF ..... 377

3.6.325. Id\_SO\_RCVTIMEO ..... 377

3.6.326. Id\_SO\_REUSEADDR ..... 378

3.6.327. Id\_SO\_SNDBUF ..... 378

3.6.328. Id\_SO\_SNDTIMEO ..... 378

3.6.329. Id SOCK\_DGRAM ..... 378

3.6.330. Id SOCK\_RAW ..... 379

3.6.331. Id SOCK\_STREAM ..... 379

3.6.332. Id\_SOCKET\_ERROR ..... 379

3.6.333. ID\_SOCKS\_AUTH ..... 379

3.6.334. ID\_SOCKS\_PORT ..... 380

3.6.335. ID\_SOCKS\_VER ..... 380

3.6.336. Id\_SOL\_SOCKET ..... 380

3.6.337. Id\_TCP\_ACK ..... 380

3.6.338. Id\_TCP\_FIN ..... 381

3.6.339. Id\_TCP\_HSIZE ..... 381

3.6.340. Id\_TCP\_NODELAY ..... 381

3.6.341. Id\_TCP\_PUSH ..... 381

3.6.342. Id\_TCP\_RST ..... 381

3.6.343. Id\_TCP\_SYN ..... 382

3.6.344. Id\_TCP\_URG ..... 382

3.6.345. Id\_TId\_HTTPAutoStartSession ..... 382

3.6.346. Id\_TId\_HTTPServer\_ParseParams ..... 382

3.6.347. Id\_TId\_HTTPServer\_SessionState ..... 383

3.6.348. Id\_TId\_HTTPSessionTimeOut ..... 383

3.6.349. Id\_TIdFinger\_VerboseOutput ..... 383

3.6.350. Id\_TIdFTP\_Passive ..... 384

3.6.351. Id\_TIdFTP\_TransferType ..... 384

3.6.352. Id\_TIdGopherServer\_TruncateLength ..... 384

3.6.353. Id\_TIdGopherServer\_TruncateUserFriendly ..... 385

3.6.354. Id\_TIdHTTP\_HandleRedirects ..... 385

3.6.355. Id\_TIdHTTP\_ProtocolVersion ..... 385

3.6.356. Id\_TIdHTTP\_RedirectMax ..... 386

3.6.357. Id\_TIDICMP\_ReceiveTimeout ..... 386

3.6.358. ID\_TIDLOGDEBUG\_TARGET ..... 386

3.6.359. Id\_TIdRawBase\_BufferSize ..... 387

3.6.360. Id\_TIdRawBase\_Port ..... 387

3.6.361. ID\_TIDSMTP\_AUTH\_TYPE ..... 387

3.6.362. ID\_UDP\_BUFFERSIZE ..... 387

3.6.363. Id\_UDP\_HSIZE ..... 388

3.6.364. Id\_WSAEACCES ..... 388

3.6.365. Id\_WSAEADDRINUSE ..... 388

3.6.366. Id\_WSAEADDRNOTAVAIL ..... 388

3.6.367. Id\_WSAEAFNOSUPPORT ..... 389

3.6.368. Id\_WSAEALREADY ..... 389

3.6.369. Id\_WSAEBADF ..... 389

3.6.370. Id\_WSAECONNABORTED ..... 389

3.6.371. Id\_WSAECONNREFUSED ..... 389

3.6.372. Id\_WSAECONNRESET ..... 390

3.6.373. Id\_WSAEDESTADDRREQ ..... 390

3.6.374. Id\_WSAEFAULT ..... 390

3.6.375. Id\_WSAEHOSTDOWN ..... 390

3.6.376. Id\_WSAEHOSTUNREACH ..... 391



3.6.377. Id\_WSAEINPROGRESS ..... 391

3.6.378. Id\_WSAEINTR ..... 391

3.6.379. Id\_WSAEINVAL ..... 391

3.6.380. Id\_WSAEISCONN ..... 391

3.6.381. Id\_WSAELOOP ..... 392

3.6.382. Id\_WSAEMFILE ..... 392

3.6.383. Id\_WSAEMSGSIZE ..... 392

3.6.384. Id\_WSAENAMETOOLONG ..... 392

3.6.385. Id\_WSAENETDOWN ..... 393

3.6.386. Id\_WSAENETRESET ..... 393

3.6.387. Id\_WSAENETUNREACH ..... 393

3.6.388. Id\_WSAENOBUFFS ..... 393

3.6.389. Id\_WSAENOPROTOPT ..... 393

3.6.390. Id\_WSAENOTCONN ..... 394

3.6.391. Id\_WSAENOTEMPTY ..... 394

3.6.392. Id\_WSAENOTSOCK ..... 394

3.6.393. Id\_WSAEOPNOTSUPP ..... 394

3.6.394. Id\_WSAEFPNOSUPPORT ..... 395

3.6.395. Id\_WSAEPROTONOSUPPORT ..... 395

3.6.396. Id\_WSAEPROTOTYPE ..... 395

3.6.397. Id\_WSAESHUTDOWN ..... 395

3.6.398. Id\_WSAESOCKTNOSUPPORT ..... 395

3.6.399. Id\_WSAETIMEDOUT ..... 396

3.6.400. Id\_WSAETOOMANYREFS ..... 396

3.6.401. Id\_WSAEWOULDBLOCK ..... 396

3.6.402. IdBeatsInDay ..... 396

3.6.403. IdDayNames ..... 397

3.6.404. IdDayShortNames ..... 397

3.6.405. IdDaysInCentury ..... 397

3.6.406. IdDaysInFourYears ..... 397

3.6.407. IdDaysInLeapCentury ..... 398

3.6.408. IdDaysInLeapYear ..... 398

3.6.409. IdDaysInLeapYearCycle ..... 398

3.6.410. IdDaysInMonth ..... 398

3.6.411. IdDaysInShortLeapYearCycle ..... 399

3.6.412. IdDaysInShortNonLeapYearCycle ..... 399

3.6.413. IdDaysInWeek ..... 399

3.6.414. IdDaysInYear ..... 399

3.6.415. IdDNSResolver\_ReceiveTimeout ..... 400

3.6.416. iDEFAULTPACKETSIZE ..... 400

3.6.417. iDEFAULTREPLYBUFSIZE ..... 400

3.6.418. IdGopherItem\_Binary ..... 401

3.6.419. IdGopherItem\_BinDOS ..... 401

3.6.420. IdGopherItem\_BinHex ..... 401

3.6.421. IdGopherItem\_CSO ..... 401

3.6.422. IdGopherItem\_Directory ..... 401

3.6.423. IdGopherItem\_Document ..... 402

3.6.424. IdGopherItem\_Error ..... 402

3.6.425. IdGopherItem\_GIF ..... 402

3.6.426. IdGopherItem\_HTML ..... 402

3.6.427. IdGopherItem\_Image ..... 402

3.6.428. IdGopherItem\_Image2 ..... 403

3.6.429. IdGopherItem\_Information ..... 403

3.6.430. IdGopherItem\_MIME ..... 403

3.6.431. IdGopherItem\_Movie ..... 403

3.6.432. IdGopherItem\_Redundant ..... 404

3.6.433. IdGopherItem\_Search ..... 404

3.6.434. IdGopherItem\_Sound ..... 404

3.6.435. IdGopherItem\_Sound2 ..... 404

3.6.436. IdGopherItem\_Telnet ..... 405

3.6.437. IdGopherItem\_TN3270 ..... 405

3.6.438. IdGopherItem\_UUE ..... 405

3.6.439. IdGopherPlusAbstract ..... 405

3.6.440. IdGopherPlusAdmin ..... 406

3.6.441. IdGopherPlusAsk ..... 406

3.6.442. IdGopherPlusAskFileName ..... 406

3.6.443. IdGopherPlusAskLong ..... 406

3.6.444. IdGopherPlusAskPassword ..... 407

3.6.445. IdGopherPlusChoose ..... 407

3.6.446. IdGopherPlusChooseFile ..... 407

3.6.447. IdGopherPlusData\_BeginSign ..... 408

3.6.448. IdGopherPlusData\_EndSign ..... 408

3.6.449. IdGopherPlusData\_ErrorBeginSign ..... 408

3.6.450. IdGopherPlusData\_ErrorUnknownSize ..... 409

3.6.451. IdGopherPlusData\_UnknownSize ..... 409

3.6.452. IdGopherPlusDirectoryInformation ..... 409

3.6.453. IdGopherPlusError\_ItemMoved ..... 409

3.6.454. IdGopherPlusError\_NotAvailable ..... 410

3.6.455. IdGopherPlusError\_TryLater ..... 410

3.6.456. IdGopherPlusIndicator ..... 410

3.6.457. IdGopherPlusInfo ..... 411

3.6.458. IdGopherPlusInformation ..... 411

3.6.459. IdGopherPlusSelect ..... 411

3.6.460. IdGopherPlusViews ..... 411

3.6.461. IdHoursInDay ..... 412

3.6.462. IdHoursInHalfDay ..... 412

3.6.463. IdMillisecondsInDay ..... 412

3.6.464. IdMillisecondsInHour ..... 412

3.6.465. IdMillisecondsInMinute ..... 413

3.6.466. IdMilliSecondsInSecond ..... 413

3.6.467. IdMillisecondsInWeek ..... 413

3.6.468. IdMinutesInHour ..... 413

3.6.469. IdMonthNames ..... 414

3.6.470. IdMonthShortNames ..... 414

3.6.471. IdMonthsInYear ..... 414

3.6.472. IdPORT\_AUTH ..... 414

3.6.473. IdPORT\_CHARGEN ..... 415

3.6.474. IdPORT\_DAYTIME ..... 415

3.6.475. IdPORT\_DICT ..... 415

3.6.476. IdPORT\_DISCARD ..... 416

3.6.477. IdPORT\_DOMAIN ..... 416

3.6.478. IdPORT\_ECHO ..... 416

3.6.479. IdPORT\_FINGER ..... 417

3.6.480. IdPORT\_FTP ..... 417

3.6.481. IdPORT\_GOPHER ..... 417

3.6.482. IdPORT\_HOSTNAME ..... 418

3.6.483. IdPORT\_HTTP ..... 418

3.6.484. IdPORT\_IMAP4 ..... 418

3.6.485. IdPORT\_IRC ..... 419

3.6.486. IdPORT\_LPD ..... 419

3.6.487. IdPORT\_NETSTAT ..... 419

3.6.488. IdPORT\_NNTP ..... 419

3.6.489. IdPORT\_POP2 ..... 420

3.6.490. IdPORT\_POP3 ..... 420

3.6.491. IdPORT\_QOTD ..... 420

3.6.492. IdPORT\_SMTP ..... 421

3.6.493. IdPORT\_SNTP ..... 421

3.6.494. IdPORT\_SSL ..... 421

3.6.495. IdPORT\_SYSTAT ..... 422

3.6.496. IdPORT\_TELNET ..... 422

3.6.497. IdPORT\_TFTP ..... 422

3.6.498. IdPORT\_TIME ..... 422

3.6.499. IdPORT\_WHOIS ..... 423

3.6.500. IdSecondsInDay ..... 423

3.6.501. IdSecondsInHalfDay ..... 423

3.6.502. IdSecondsInHour ..... 424

3.6.503. IdSecondsInLeapYear ..... 424

3.6.504. IdSecondsInMinute ..... 424

3.6.505. IdSecondsInWeek ..... 424

3.6.506. IdSecondsInYear ..... 425

3.6.507. IdStati ..... 425

3.6.508. IdTimeoutDefault ..... 425

3.6.509. IdTimeoutInfinite ..... 426

3.6.510. IdYearsInCentury ..... 426

3.6.511. IdYearsInLeapYearCycle ..... 426

3.6.512. IdYearsInShortLeapYearCycle ..... 426

3.6.513. IMAPCommands ..... 427

3.6.514. IMPLINK\_HIGHEXPER ..... 427

3.6.515. IMPLINK\_IP ..... 427

3.6.516. IMPLINK\_LOWEXPER ..... 427

3.6.517. INVALID\_SOCKET ..... 428

3.6.518. IOC\_IN ..... 428

3.6.519. IOC\_INOUT ..... 428

3.6.520. IOC\_OUT ..... 428

3.6.521. IOC\_VOID ..... 428

3.6.522. IOCPARM\_MASK ..... 429

3.6.523. IP\_ADD\_MEMBERSHIP ..... 429

3.6.524. IP\_DEFAULT\_MULTICAST\_LOOP ..... 429

3.6.525. IP\_DEFAULT\_MULTICAST\_TTL ..... 429

3.6.526. IP\_DONTFRAGMENT ..... 430

3.6.527. IP\_DROP\_MEMBERSHIP ..... 430

3.6.528. IP\_MAX\_MEMBERSHIPS ..... 430

3.6.529. IP\_MULTICAST\_IF ..... 430

3.6.530. IP\_MULTICAST\_LOOP ..... 430

3.6.531. IP\_MULTICAST\_TTL ..... 431

3.6.532. IP\_OPTIONS ..... 431

3.6.533. IP\_TOS ..... 431

3.6.534. IP\_TTL ..... 431

3.6.535. IP\_WATCH\_ACTIVE ..... 432

3.6.536. IP\_WATCH\_HIST\_ENABLED ..... 432

3.6.537. IP\_WATCH\_HIST\_FILENAME ..... 432

3.6.538. IP\_WATCH\_HIST\_MAX ..... 433

3.6.539. IP\_WATCH\_INTERVAL ..... 433

3.6.540. IPPORT\_RESERVED ..... 433

3.6.541. IPPROTO\_GGP ..... 433

3.6.542. IPPROTO\_ICMP ..... 434

3.6.543. IPPROTO\_IDP ..... 434

3.6.544. IPPROTO\_IGMP ..... 434

3.6.545. IPPROTO\_IP ..... 434

3.6.546. IPPROTO\_MAX ..... 434

3.6.547. IPPROTO\_ND ..... 435

3.6.548. IPPROTO\_PUP ..... 435

3.6.549. IPPROTO\_RAW ..... 435

3.6.550. IPPROTO\_TCP ..... 435

3.6.551. IPPROTO\_UDP ..... 436

3.6.552. kana\_tbl ..... 436

3.6.553. KnownCommands ..... 436

3.6.554. LF ..... 437

3.6.555. MAX\_PACKET\_SIZE ..... 437

3.6.556. MAXGETHOSTSTRUCT ..... 437

3.6.557. MaxMIMEBinToASCIIType ..... 437

3.6.558. MaxMIMECompressType ..... 438

3.6.559. MaxMIMEEncType ..... 438

3.6.560. MaxMIMEMessageDigestType ..... 438

3.6.561. MaxMIMESubTypes ..... 438

3.6.562. MaxMIMETYPE ..... 439

3.6.563. maxPriv ..... 439

3.6.564. MaxWord ..... 439

3.6.565. MIME7Bit ..... 439

3.6.566. MIMEEncBase64 ..... 440

3.6.567. MIMEEncNISTSHA ..... 440

3.6.568. MIMEEncRLECompress ..... 440

3.6.569. MIMEEncRSAMD2 ..... 440

3.6.570. MIMEEncRSAMD4 ..... 440

3.6.571. MIMEEncRSAMD5 ..... 441

3.6.572. MIMEEncUUEncode ..... 441

3.6.573. MIMEEncXXEncode ..... 441

3.6.574. MIMEFullApplicationOctetStream ..... 441

3.6.575. MIMEGenericText ..... 441

3.6.576. MIMESplit ..... 442

3.6.577. MIMESubMacBinHex40 ..... 442

3.6.578. MIMESubOctetStream ..... 442

3.6.579. MIMETYPEApplication ..... 442

3.6.580. MIMETYPEAudio ..... 442

3.6.581. MIMETYPEImage ..... 443

3.6.582. MIMETYPEMessage ..... 443

3.6.583. MIMETYPEMultipart ..... 443

3.6.584. MIMETYPEText ..... 443

3.6.585. MIMETYPEVideo ..... 443

3.6.586. MIMEXVal ..... 444

3.6.587. minPriv ..... 444

3.6.588. MSG\_DONTROUTE ..... 444

3.6.589. MSG\_MAXIOVLEN ..... 444

3.6.590. MSG\_OOB ..... 445

3.6.591. MSG\_PARTIAL ..... 445

3.6.592. MSG\_PEEK ..... 445

3.6.593. MultiPartAlternativeBoundary ..... 445

3.6.594. MultiPartBoundary ..... 446

3.6.595. MultiPartRelatedBoundary ..... 446

3.6.596. NO\_ADDRESS ..... 446

3.6.597. NO\_DATA ..... 447

3.6.598. NO\_RECOVERY ..... 447

3.6.599. NTPMaxInt ..... 447

3.6.600. NumberOfClientsType ..... 447

3.6.601. NumberOfConnectionsType ..... 448

3.6.602. NumberOfPacketsType ..... 448

3.6.603. NumberOfServicesType ..... 448

3.6.604. NumberOfSlavesType ..... 449

3.6.605. PF\_APPLETALK ..... 449

3.6.606. PF\_BAN ..... 449

3.6.607. PF\_CCITT ..... 449

3.6.608. PF\_CHAOS ..... 450

3.6.609. PF\_DATAKIT ..... 450

3.6.610. PF\_DECnet ..... 450

3.6.611. PF\_DLI ..... 450

3.6.612. PF\_ECMA ..... 450

3.6.613. PF\_FIREFOX ..... 451

3.6.614. PF\_HYLINK ..... 451

3.6.615. PF\_IMPLINK ..... 451

3.6.616. PF\_INET ..... 451

3.6.617. PF\_IPX ..... 452

3.6.618. PF\_ISO ..... 452

3.6.619. PF\_LAT ..... 452

3.6.620. PF\_MAX ..... 452

3.6.621. PF\_NS ..... 452

3.6.622. PF\_OSI ..... 453

3.6.623. PF\_PUP ..... 453

3.6.624. PF\_SNA ..... 453

3.6.625. PF\_UNIX ..... 453

3.6.626. PF\_UNKNOWN1 ..... 454

3.6.627. PF\_UNSPEC ..... 454

3.6.628. PF\_VOICEVIEW ..... 454

3.6.629. RSAAboutBoxCompName ..... 454

3.6.630. RSAAboutBoxCopyright ..... 454

3.6.631. RSAAboutBoxIndyWebsite ..... 455

3.6.632. RSAAboutBoxPleaseVisit ..... 455

3.6.633. RSAAboutBoxVersion ..... 455

3.6.634. RSAAboutCreditsCoCoordinator ..... 455

3.6.635. RSAAboutCreditsCoordinator ..... 456

3.6.636. RSAAboutFormCaption ..... 456

3.6.637. RSAAboutMenuItemName ..... 456

3.6.638. RSAcceptWaitCannotBeModifiedWhileServerIsActive ..... 456

3.6.639. RSAlreadyConnected ..... 457

3.6.640. RSByteIndexOutOfBounds ..... 457

3.6.641. RSCannotAllocateSocket ..... 457

3.6.642. RSCannotChangeDebugTargetAtWhileActive ..... 457

3.6.643. RSCMDNotRecognized ..... 458

3.6.644. RSCodeNoError ..... 458

3.6.645. RSCodeQueryFormat ..... 458

3.6.646. RSCodeQueryName ..... 458

3.6.647. RSCodeQueryNotImplemented ..... 458

3.6.648. RSCodeQueryQueryRefused ..... 459

3.6.649. RSCodeQueryServer ..... 459

3.6.650. RSCodeQueryUnknownError ..... 459

3.6.651. RSCoderNoTableEntryNotFound ..... 459

3.6.652. RSConnectionClosedGracefully ..... 460

3.6.653. RSCorruptServicesFile ..... 460

3.6.654. RSCouldNotBindSocket ..... 460

3.6.655. RSCouldNotLoad ..... 460

3.6.656. RSDestinationFileAlreadyExists ..... 461

3.6.657. RSDNSMailAObsolete ..... 461

3.6.658. RSDNSMailBNotImplemented ..... 461

3.6.659. RSDNSMDISObsolete ..... 461

3.6.660. RSDNSMFIsObsolete ..... 461

3.6.661. RSFailedTimeZonelInfo ..... 462

3.6.662. RSFTPUnknownHost ..... 462

3.6.663. RSGopherNotGopherPlus ..... 462

3.6.664. RSGopherServerNoProgramCode ..... 462

3.6.665. RSHTTPAccepted ..... 463

3.6.666. RSHTTPBadGateway ..... 463

3.6.667. RSHTTPBadRequest ..... 463

3.6.668. RSHTTPCannotSwitchSessionStateWhenActive ..... 463

3.6.669. RSHTTPChunkStarted ..... 464

3.6.670. RSHTTPConflict ..... 464

3.6.671. RSHTTPContinue ..... 464

3.6.672. RSHTTPCreated ..... 464

3.6.673. RSHTTPErrorParsingCommand ..... 464

3.6.674. RSHTTPForbidden ..... 465

3.6.675. RSHTTPGatewayTimeout ..... 465

3.6.676. RSHTTPGone ..... 465

3.6.677. RSHTTPHeaderAlreadyWritten ..... 465

3.6.678. RSHTTPHTTPVersionNotSupported ..... 466

3.6.679. RSHTTPInternalServerError ..... 466

3.6.680. RSHTTPLengthRequired ..... 466

3.6.681. RSHTTPMethodNotAllowed ..... 466

3.6.682. RSHTTPMovedPermanently ..... 466

3.6.683. RSHTTPMovedTemporarily ..... 467

3.6.684. RSHTTPNoContent ..... 467

3.6.685. RSHTTPNonAuthoritativeInformation ..... 467

3.6.686. RSHTTPNotAcceptable ..... 467

3.6.687. RSHTTPNotFound ..... 468

3.6.688. RSHTTPNotImplemented ..... 468

3.6.689. RSHTTPNotModified ..... 468

3.6.690. RSHTTPOK ..... 468

3.6.691. RSHTTTPartialContent ..... 468

3.6.692. RSHTTTPreconditionFailed ..... 469

3.6.693. RSHTTTProxyAuthenticationRequired ..... 469

3.6.694. RSHTTTPRequestEntityTooLong ..... 469

3.6.695. RSHTTTPRequestTimeout ..... 469

3.6.696. RSHTTTPRequestURITooLong ..... 470

3.6.697. RSHTTTPResetContent ..... 470

3.6.698. RSHTTTPSeeOther ..... 470

3.6.699. RSHTTTPServiceUnavailable ..... 470

3.6.700. RSHTTTPSwitchingProtocols ..... 470

3.6.701. RSHTTTPUnauthorized ..... 471

3.6.702. RSHTTTPUnknownResponseCode ..... 471

3.6.703. RSHTTTPUnsupportedAuthorisationScheme ..... 471

3.6.704. RSHTTTPUnsupportedMediaType ..... 471

3.6.705. RSHTTTPUseProxy ..... 472

3.6.706. RSICMPNonEchoResponse ..... 472

3.6.707. RSICMPNotEnoughBytes ..... 472

3.6.708. RSICMPReceiveError0 ..... 472

3.6.709. RSICMPWrongDestination ..... 472

3.6.710. RSIdNoDataToRead ..... 473

3.6.711. RSInterceptPropInvalid ..... 473

3.6.712. RSInterceptProplsNil ..... 473

3.6.713. RSInvalidServiceName ..... 473

3.6.714. RSLPDAbortJob ..... 474

3.6.715. RSLPDClosingConnection ..... 474

3.6.716. RSLPDConnectTo ..... 474

3.6.717. RSLPDControlFileSaved ..... 474

3.6.718. RSLPDDataFileSaved ..... 474

3.6.719. RSLPDDirectoryDoesNotExist ..... 475

3.6.720. RSLPDNoQueuesDefined ..... 475

3.6.721. RSLPDQueueStatus ..... 475

3.6.722. RSLPDReceiveControlFile ..... 475

3.6.723. RSLPDReceiveDataFile ..... 476

3.6.724. RSLPDServerActive ..... 476

3.6.725. RSLPDServerStartTitle ..... 476

3.6.726. RSLPDUnknownQueue ..... 476

3.6.727. RSMsgClientEncodingAttachment ..... 476

3.6.728. RSMsgClientEncodingText ..... 477

3.6.729. RSMsgCmpEdtrBodyText ..... 477

3.6.730. RSMsgCmpEdtrExtraHead ..... 477

3.6.731. RSMsgCmpEdtrNew ..... 477

3.6.732. RSNETCALCInvalidNetworkMask ..... 478

3.6.733. RSNETCALCInvalidValueLength ..... 478

3.6.734. RSNETCALConfirmLongIPList ..... 478

3.6.735. RSNETCALInvalidIPString ..... 478

3.6.736. RSNNTTPConnectionRefused ..... 479

3.6.737. RSNNTTPNoOnNewGroupsList ..... 479

3.6.738. RSNNTTPNoOnNewNewsList ..... 479

3.6.739. RSNNTTPNoOnNewsgroupList ..... 479

3.6.740. RSNNTTPServerGoodBye ..... 480

3.6.741. RSNNTTPServerNotRecognized ..... 480

3.6.742. RSNNTTPStringListNotInitialized ..... 480

3.6.743. RSNoBindingsSpecified ..... 480

3.6.744. RSNoExecuteSpecified ..... 480

3.6.745. RSNotAllBytesSent ..... 481

3.6.746. RSNotEnoughDataInBuffer ..... 481

3.6.747. RSOBJECTTypeNotSupported ..... 481

3.6.748. RSONExecuteNotAssigned ..... 481

3.6.749. RSOnlyOneAntiFreeze ..... 482

3.6.750. RSOSslCertificateLookup ..... 482

3.6.751. RSOSslConnectionDropped ..... 482

3.6.752. RSOSslCouldNotLoadSSLLibrary ..... 482

3.6.753. RSOSslInternal ..... 483

3.6.754. RSOSslModeNotSet ..... 483

3.6.755. RSOSslStatusString ..... 483

3.6.756. RSPackageSizeTooBig ..... 483

3.6.757. RSPOP3FieldNotSpecified ..... 483

3.6.758. RSQueryInvalidHeaderID ..... 484

3.6.759. RSQueryInvalidPacketSize ..... 484

3.6.760. RSQueryInvalidQueryCount ..... 484

3.6.761. RSQueryLessThanFour ..... 484

3.6.762. RSQueryLessThanTwelve ..... 485

3.6.763. RSQueryPackReceivedTooSmall ..... 485

3.6.764. RSSRawReceiveError0 ..... 485

3.6.765. RSSSetSizeExceeded ..... 485

3.6.766. RSSocksAuthError ..... 486

3.6.767. RSSocksAuthMethodError ..... 486

3.6.768. RSSocksRequestFailed ..... 486

3.6.769. RSSocksRequestIdentFailed ..... 486

3.6.770. RSSocksRequestServerFailed ..... 487

3.6.771. RSSocksServerAddressError ..... 487

3.6.772. RSSocksServerCommandError ..... 487

3.6.773. RSSocksServerConnectionRefusedError ..... 487

3.6.774. RSSocksServerGeneralError ..... 488

3.6.775. RSSocksServerHostUnreachableError ..... 488

3.6.776. RSSocksServerNetUnreachableError ..... 488

3.6.777. RSSocksServerPermissionError ..... 488

3.6.778. RSSocksServerRespondError ..... 488

3.6.779. RSSocksServerTTLExpiredError ..... 489

3.6.780. RSSocksUnknownError ..... 489

3.6.781. RSSSLAcceptError ..... 489

3.6.782. RSSSLConnectError ..... 489

3.6.783. RSSSLCreatingContextError ..... 490

3.6.784. RSSSLDataBindingError ..... 490

3.6.785. RSSSLGetMethodError ..... 490

3.6.786. RSSSLLoadingCertError ..... 490

3.6.787. RSSSLLoadingKeyError ..... 490

3.6.788. RSSSLLoadingRootCertError ..... 491

3.6.789. RSSSLSettingChiperError ..... 491

3.6.790. RSStackEACCES ..... 491

3.6.791. RSStackEADDRINUSE ..... 491

3.6.792. RSStackEADDRNOTAVAIL ..... 492

3.6.793. RSStackEAFNOSUPPORT ..... 492

3.6.794. RSStackEALREADY ..... 492

3.6.795. RSStackEBADF ..... 492

3.6.796. RSStackECONNABORTED ..... 493

3.6.797. RSStackECONNREFUSED ..... 493

3.6.798. RSStackECONNRESET ..... 493

3.6.799. RSStackEDESTADDRREQ ..... 493

3.6.800. RSStackEDQUOT ..... 493

3.6.801. RSStackEFAULT ..... 494

3.6.802. RSStackEHOSTDOWN ..... 494

3.6.803. RSStackEHOSTUNREACH ..... 494

3.6.804. RSStackEINPROGRESS ..... 494

3.6.805. RSStackEINTR ..... 495

3.6.806. RSStackEINVAL ..... 495

3.6.807. RSStackEISCONN ..... 495

3.6.808. RSStackELOOP ..... 495

3.6.809. RSStackEMFILE ..... 495

3.6.810. RSStackEMSGSIZE ..... 496

3.6.811. RSStackENAMETOOLONG ..... 496

3.6.812. RSStackENETDOWN ..... 496

3.6.813. RSStackENETRESET ..... 496

3.6.814. RSStackENETUNREACH ..... 497

3.6.815. RSStackENOBUFS ..... 497

3.6.816. RSStackENOPROTOOPT ..... 497

3.6.817. RSStackENOTCONN ..... 497

3.6.818. RSStackENOTEMPTY ..... 497

3.6.819. RSStackENOTSOCK ..... 498

3.6.820. RSStackEOPNOTSUPP ..... 498

3.6.821. RSStackEPFNOSUPPORT ..... 498

3.6.822. RSStackEPROCLIM ..... 498

3.6.823. RSStackEPROTONOSUPPORT ..... 499

3.6.824. RSStackEPROTOTYPE ..... 499

3.6.825. RSStackEREMOTE ..... 499

3.6.826. RSStackError ..... 499

3.6.827. RSStackESHUTDOWN ..... 499

3.6.828. RSStackESOCKTNOSUPPORT ..... 500

3.6.829. RSStackESTALE ..... 500

3.6.830. RSStackETIMEDOUT ..... 500

3.6.831. RSStackETOOMANYREFS ..... 500

3.6.832. RSStackEUSERS ..... 501

3.6.833. RSStackEWOULDBLOCK ..... 501

3.6.834. RSStackHOST\_NOT\_FOUND ..... 501

3.6.835. RSStackNO\_DATA ..... 501

3.6.836. RSStackNO\_RECOVERY ..... 501

3.6.837. RSStackNOTINITIALISED ..... 502

3.6.838. RSStackSYSNOTREADY ..... 502

3.6.839. RSStackTRY\_AGAIN ..... 502

3.6.840. RSStackVERNOTSUPPORTED ..... 502

3.6.841. RSStatusConnected ..... 503

3.6.842. RSStatusConnecting ..... 503

3.6.843. RSStatusDisconnected ..... 503

3.6.844. RSStatusDisconnecting ..... 503

3.6.845. RSStatusResolving ..... 503

3.6.846. RSStatusText ..... 504

3.6.847. RSTELNETCLIConnectError ..... 504

3.6.848. RSTELNETCLIReadError ..... 504

3.6.849. RSTELNETSRVInvalidLogin ..... 504

3.6.850. RSTELNETSRVMaxloginAttempt ..... 505

3.6.851. RSTELNETSRVNoAuthHandler ..... 505

3.6.852. RSTELNETSRVOnDataAvailableIsNil ..... 505

3.6.853. RSTELNETSRVPasswordPrompt ..... 505

3.6.854. RSTELNETSRVUsernamePrompt ..... 506

3.6.855. RSTELNETSRVWelcomeString ..... 506

3.6.856. RSTFTPAccessDenied ..... 506

3.6.857. RSTFTPDiskFull ..... 506

3.6.858. RSTFTPFileNotFound ..... 507

3.6.859. RSTFTPUnexpectedOp ..... 507

3.6.860. RSTFTPUnsupportedTrxMode ..... 507

3.6.861. RSThreadClassNotSpecified ..... 507

3.6.862. RSTIdMessagePartCreate ..... 507

3.6.863. RSTIdTextInvalidCount ..... 508

3.6.864. RSTimeOut ..... 508

3.6.865. RSTunnelConnectMsg ..... 508

3.6.866. RSTunnelConnectToMasterFailed ..... 508

3.6.867. RSTunnelCRCFailed ..... 509

3.6.868. RSTunnelDisconnectMsg ..... 509

3.6.869. RSTunnelDontAllowConnections ..... 509

3.6.870. RSTunnelGetByteRange ..... 509

3.6.871. RSTunnelMessageCustomInterpretError ..... 510

3.6.872. RSTunnelMessageHandlingError ..... 510

3.6.873. RSTunnelMessageInterpretError ..... 510

3.6.874. RSTunnelMessageTypeError ..... 510

3.6.875. RSTunnelTransformError ..... 511

3.6.876. RSTunnelTransformErrorBS ..... 511

3.6.877. RSUDPReceiveError0 ..... 511

3.6.878. RSWinsockInitializationError ..... 511

3.6.879. RSWSockStack ..... 511

3.6.880. sBlockSize ..... 512

3.6.881. sj1\_tbl ..... 512

3.6.882. sj2\_tbl ..... 512

3.6.883. SO\_ACCEPTCONN ..... 513

3.6.884. SO\_BROADCAST ..... 514

3.6.885. SO\_CONNDATA ..... 514

3.6.886. SO\_CONNDATALEN ..... 514

3.6.887. SO\_CONNECT\_TIME ..... 514

3.6.888. SO\_CONNOPT ..... 514

3.6.889. SO\_CONNOPTLEN ..... 515

3.6.890. SO\_DEBUG ..... 515

3.6.891. SO\_DISCDATA ..... 515

3.6.892. SO\_DISCDATALEN ..... 515

3.6.893. SO\_DISCOPT ..... 516

3.6.894. SO\_DISCOPTLEN ..... 516

3.6.895. SO\_DONTLINGER ..... 516

3.6.896. SO\_DONTROUTE ..... 516

3.6.897. SO\_ERROR ..... 516

3.6.898. SO\_KEEPALIVE ..... 517

3.6.899. SO\_LINGER ..... 517

3.6.900. SO\_MAXDG ..... 517

3.6.901. SO\_MAXPATHDG ..... 517

3.6.902. SO\_OOBLINE ..... 518

3.6.903. SO\_OPENTYPE ..... 518

3.6.904. SO\_RCVBUF ..... 518

3.6.905. SO\_RCVLOWAT ..... 518

3.6.906. SO\_RCVTIMEO ..... 518

3.6.907. SO\_REUSEADDR ..... 519

3.6.908. SO\_SNDBUF ..... 519

3.6.909. SO\_SNDLOWAT ..... 519

3.6.910. SO\_SNDTIMEO ..... 519

3.6.911. SO\_SYNCHRONOUS\_ALERT ..... 520

3.6.912. SO\_SYNCHRONOUS\_NONALERT ..... 520

3.6.913. SO\_TYPE ..... 520

3.6.914. SO\_UPDATE\_ACCEPT\_CONTEXT ..... 520

3.6.915. SO\_USELOOPBACK ..... 520

3.6.916. SOCK\_DGRAM ..... 521

3.6.917. SOCK\_RAW ..... 521

3.6.918. SOCK\_RDM ..... 521

3.6.919. SOCK\_SEQPACKET ..... 521

3.6.920. SOCK\_STREAM ..... 522

3.6.921. SOCKET\_ERROR ..... 522

3.6.922. SOL\_SOCKET ..... 522

3.6.923. SOMAXCONN ..... 522

3.6.924. TAB ..... 522

3.6.925. TCP\_BSDURGENT ..... 523

3.6.926. TCP\_NODELAY ..... 523

3.6.927. TF\_DISCONNECT ..... 523

3.6.928. TF\_REUSE\_SOCKET ..... 523

3.6.929. TF\_WRITE\_BEHIND ..... 524

3.6.930. TFTP\_ACK ..... 524

3.6.931. TFTP\_DATA ..... 524

3.6.932. TFTP\_ERROR ..... 524

3.6.933. TFTP\_OACK ..... 525

3.6.934. TFTP\_RRQ ..... 525

3.6.935. TFTP\_WRQ ..... 525

3.6.936. tmConnect ..... 525



3.6.937. tmCustom ..... 526

3.6.938. tmData ..... 526

3.6.939. tmDisconnect ..... 526

3.6.940. tmError ..... 526

3.6.941. TNC\_AO ..... 527

3.6.942. TNC\_AYT ..... 527

3.6.943. TNC\_BREAK ..... 527

3.6.944. TNC\_DATA\_MARK ..... 528

3.6.945. TNC\_DO ..... 528

3.6.946. TNC\_DONT ..... 528

3.6.947. TNC\_EC ..... 528

3.6.948. TNC\_EL ..... 529

3.6.949. TNC\_EOR ..... 529

3.6.950. TNC\_GA ..... 529

3.6.951. TNC\_IAC ..... 530

3.6.952. TNC\_IP ..... 530

3.6.953. TNC\_NOP ..... 530

3.6.954. TNC\_SB ..... 530

3.6.955. TNC\_SE ..... 531

3.6.956. TNC\_WILL ..... 531

3.6.957. TNC\_WONT ..... 531

3.6.958. TNO\_3270REGIME ..... 532

3.6.959. TNO\_AMSN ..... 532

3.6.960. TNO\_AUTH ..... 532

3.6.961. TNO\_BINARY ..... 532

3.6.962. TNO\_BYTE\_MACRO ..... 533

3.6.963. TNO\_DET ..... 533

3.6.964. TNO\_EA ..... 533

3.6.965. TNO\_ECHO ..... 534

3.6.966. TNO\_ENCRYPT ..... 534

3.6.967. TNO\_EOL ..... 534

3.6.968. TNO\_EOR ..... 534

3.6.969. TNO\_LINEMODE ..... 535

3.6.970. TNO\_LOGOUT ..... 535

3.6.971. TNO\_NAWS ..... 535

3.6.972. TNO\_OCRD ..... 536

3.6.973. TNO\_OFD ..... 536

3.6.974. TNO\_OHTD ..... 536

3.6.975. TNO\_OHTS ..... 536

3.6.976. TNO\_OLD ..... 537

3.6.977. TNO\_OLW ..... 537

3.6.978. TNO\_OM ..... 537

3.6.979. TNO\_OPS ..... 538

3.6.980. TNO\_OVT ..... 538

3.6.981. TNO\_OVTD ..... 538

3.6.982. TNO\_RCTE ..... 538

3.6.983. TNO\_RECONNECT ..... 539

3.6.984. TNO\_RFLOW ..... 539

3.6.985. TNO\_SGA ..... 539

3.6.986. TNO\_SL ..... 540

3.6.987. TNO\_STATUS ..... 540

3.6.988. TNO\_SUPDUP ..... 540

3.6.989. TNO\_SUPDUP\_OUTPUT ..... 540

3.6.990. TNO\_TACACS\_ID ..... 541

3.6.991. TNO\_TERM\_SPEED ..... 541

3.6.992. TNO\_TERMTYPE ..... 541

3.6.993. TNO\_TIMING\_MARK ..... 542

3.6.994. TNO\_TLN ..... 542

3.6.995. TNO\_X3PAD ..... 542

3.6.996. TNO\_XDISPLOC ..... 542

3.6.997. TNOS\_NAME ..... 543

3.6.998. TNOS\_REPLY ..... 543

3.6.999. TNOS\_TERM\_IS ..... 543

3.6.1000. TNOS\_TERMTYPE\_SEND ..... 543

3.6.1001. TRY\_AGAIN ..... 544

3.6.1002. TZ\_ADT ..... 544

3.6.1003. TZ\_AHST ..... 544

3.6.1004. TZ\_AST ..... 544

3.6.1005. TZ\_AT ..... 545

3.6.1006. TZ\_BST ..... 545

3.6.1007. TZ\_BT ..... 545

3.6.1008. TZ\_CAT ..... 545

3.6.1009. TZ\_CCT ..... 545

3.6.1010. TZ\_CDT ..... 546

3.6.1011. TZ\_CET ..... 546

3.6.1012. TZ\_CST ..... 546

3.6.1013. TZ\_EADT ..... 546

3.6.1014. TZ\_EAST ..... 547

3.6.1015. TZ\_EDT ..... 547

3.6.1016. TZ\_EET ..... 547

3.6.1017. TZ\_EST ..... 547

3.6.1018. TZ\_FST ..... 547

3.6.1019. TZ\_FWT ..... 548

3.6.1020. TZ\_GMT ..... 548

3.6.1021. TZ\_GST ..... 548

3.6.1022. TZ\_HDT ..... 548

3.6.1023. TZ\_HST ..... 549

3.6.1024. TZ\_IDLE ..... 549

3.6.1025. TZ\_IDLW ..... 549

3.6.1026. TZ\_JST ..... 549

3.6.1027. TZ\_MDT ..... 549

3.6.1028. TZ\_MEST ..... 550

3.6.1029. TZ\_MESZ ..... 550

3.6.1030. TZ\_MET ..... 550

3.6.1031. TZ\_MEWT ..... 550

3.6.1032. TZ\_MST ..... 551

3.6.1033. TZ\_NT ..... 551

3.6.1034. TZ\_NZDT ..... 551

3.6.1035. TZ\_NZST ..... 551

3.6.1036. TZ\_NZT ..... 551

3.6.1037. TZ\_PDT ..... 552

3.6.1038. TZ\_PST ..... 552

3.6.1039. TZ\_SST ..... 552

3.6.1040. TZ\_SWT ..... 552

3.6.1041. TZ\_UT ..... 553

3.6.1042. TZ.UTC ..... 553

3.6.1043. TZ\_WADT ..... 553

3.6.1044. TZ\_WAST ..... 553

3.6.1045. TZ\_WAT ..... 553

3.6.1046. TZ\_WET ..... 554

3.6.1047. TZ\_YDT ..... 554

3.6.1048. TZ\_YST ..... 554

3.6.1049. TZ\_ZP4 ..... 554

3.6.1050. TZ\_ZP5 ..... 555

3.6.1051. TZ\_ZP6 ..... 555

3.6.1052. TZM\_A ..... 555

3.6.1053. TZM\_Alpha ..... 555

3.6.1054. TZM\_B ..... 555

3.6.1055. TZM\_Bravo ..... 556

3.6.1056. TZM\_C ..... 556

3.6.1057. TZM\_Charlie ..... 556

3.6.1058. TZM\_D ..... 556

3.6.1059. TZM\_Delta ..... 557

3.6.1060. TZM\_E ..... 557

3.6.1061. TZM\_Echo ..... 557

3.6.1062. TZM\_F ..... 557

3.6.1063. TZM\_Foxtrot ..... 557

3.6.1064. TZM\_G ..... 558

3.6.1065. TZM\_Golf ..... 558

3.6.1066. TZM\_H ..... 558

3.6.1067. TZM\_Hotel ..... 558

3.6.1068. TZM\_J ..... 559

3.6.1069. TZM\_Juliet ..... 559

3.6.1070. TZM\_K ..... 559

3.6.1071. TZM\_Kilo ..... 559

3.6.1072. TZM\_L ..... 559

3.6.1073. TZM\_Lima ..... 560

3.6.1074. TZM\_M ..... 560

3.6.1075. TZM\_Mike ..... 560

3.6.1076. TZM\_N ..... 560

3.6.1077. TZM\_November ..... 561

3.6.1078. TZM\_O ..... 561

3.6.1079. TZM\_Oscar ..... 561

3.6.1080. TZM\_P ..... 561

3.6.1081. TZM\_Papa ..... 561

3.6.1082. TZM\_Q ..... 562

3.6.1083. TZM\_Quebec ..... 562

3.6.1084. TZM\_R ..... 562

3.6.1085. TZM\_Romeo ..... 562

3.6.1086. TZM\_S ..... 563

3.6.1087. TZM\_Sierra ..... 563

3.6.1088. TZM\_T ..... 563

3.6.1089. TZM\_Tango ..... 563

3.6.1090. TZM\_U ..... 563

3.6.1091. TZM\_Uniform ..... 564

3.6.1092. TZM\_V ..... 564

3.6.1093. TZM\_Victor ..... 564

3.6.1094. TZM\_W ..... 564

3.6.1095. TZM\_Whiskey ..... 565

3.6.1096. TZM\_X ..... 565

3.6.1097. TZM\_XRay ..... 565

3.6.1098. TZM\_Y ..... 565

3.6.1099. TZM\_Yankee ..... 565

3.6.1100. TZM\_Z ..... 566

3.6.1101. TZM\_Zulu ..... 566

3.6.1102. UUBegin ..... 566

3.6.1103. UUBEGINFound ..... 566

3.6.1104. UUCodeTable ..... 567

3.6.1105. UUDataStarted ..... 567

3.6.1106. UUEnd ..... 567

3.6.1107. UUENDFound ..... 568

3.6.1108. UUErrIncompletePrivilege ..... 568

3.6.1109. UUErrIncompletePrivilege2 ..... 568

3.6.1110. UUErrorDataEndWithoutEND ..... 569

3.6.1111. UUErrorNoBEGINAfterTABLE ..... 569

3.6.1112. UUErrorPivillageNotNumeric ..... 569

3.6.1113. UUErrTableNotAtEnd ..... 569

3.6.1114. UUInitialLength ..... 570

3.6.1115. UULastCharFound ..... 570

3.6.1116. UUPrivilegeFound ..... 570

3.6.1117. UUStarted ..... 571

3.6.1118. UUTable ..... 571

3.6.1119. UUTableBeenRead ..... 571

3.6.1120. UUTableBegun ..... 571

3.6.1121. UUTableOneLine ..... 572

3.6.1122. vkana\_tbl ..... 572

3.6.1123. WSABASEERR ..... 572

3.6.1124. WSAEACCES ..... 573

3.6.1125. WSAEADDRINUSE ..... 573

3.6.1126. WSAEADDRNOTAVAIL ..... 573

3.6.1127. WSAEAFNOSUPPORT ..... 573

3.6.1128. WSAEALREADY ..... 573

3.6.1129. WSAEBADF ..... 574

3.6.1130. WSAECONNABORTED ..... 574

3.6.1131. WSAECONNREFUSED ..... 574

3.6.1132. WSAECONNRESET ..... 574

3.6.1133. WSAEDESTADDRREQ ..... 575

3.6.1134. WSAEDISCON ..... 575

3.6.1135. WSAEDQUOT ..... 575

3.6.1136. WSAEFAULT ..... 575

3.6.1137. WSAEHOSTDOWN ..... 575

3.6.1138. WSAEHOSTUNREACH ..... 576

3.6.1139. WSAEINPROGRESS ..... 576

3.6.1140. WSAEINTR ..... 576

3.6.1141. WSAEINVAL ..... 576

3.6.1142. WSAEISCONN ..... 577

3.6.1143. WSAELOOP ..... 577

3.6.1144. WSAEMFILE ..... 577

3.6.1145. WSAEMSGSIZE ..... 577

3.6.1146. WSAENAMETOOLONG ..... 577

3.6.1147. WSAENETDOWN .....	578
3.6.1148. WSAENETRESET .....	578
3.6.1149. WSAENETUNREACH .....	578
3.6.1150. WSAENOBUFFS .....	578
3.6.1151. WSAENOPROTOOPT .....	579
3.6.1152. WSAENOTCONN .....	579
3.6.1153. WSAENOTEMPTY .....	579
3.6.1154. WSAENOTSOCK .....	579
3.6.1155. WSAEOPNOTSUPP .....	579
3.6.1156. WSAEPFNOSUPPORT .....	580
3.6.1157. WSAEPROCLIM .....	580
3.6.1158. WSAEPROTONOSUPPORT .....	580
3.6.1159. WSAEPROTOTYPE .....	580
3.6.1160. WSAEREMOTE .....	581
3.6.1161. WSAESHUTDOWN .....	581
3.6.1162. WSAESOCKTNOSUPPORT .....	581
3.6.1163. WSAESTALE .....	581
3.6.1164. WSAETIMEDOUT .....	581
3.6.1165. WSAETOOMANYREFS .....	582
3.6.1166. WSAEUSERS .....	582
3.6.1167. WSAEWOULDBLOCK .....	582
3.6.1168. WSAHOST_NOT_FOUND .....	582
3.6.1169. WSANO_ADDRESS .....	583
3.6.1170. WSANO_DATA .....	583
3.6.1171. WSANO_RECOVERY .....	583
3.6.1172. WSANOTINITIALISED .....	583
3.6.1173. WSASYSNOTREADY .....	583
3.6.1174. WSATRY_AGAIN .....	584
3.6.1175. WSAVERNOTSUPPORTED .....	584
3.6.1176. wsErr .....	584
3.6.1177. wsOk .....	584
3.6.1178. XXCodeTable .....	585
3.7. Units .....	585
3.7.1. IdAntiFreeze.pas .....	585

3.7.2. IdAntiFreezeBase.pas .....	585
3.7.3. IdBaseComponent.pas .....	586
3.7.4. IdChargenServer.pas .....	586
3.7.5. IdCoder.pas .....	586
3.7.6. IdCoder3To4.pas .....	586
3.7.7. IdCoderIMF.pas .....	587
3.7.8. IdCoderMessageDigest.pas .....	587
3.7.9. IdCoderText.pas .....	587
3.7.10. IdCompilerDefines.inc .....	587
3.7.11. IdComponent.pas .....	587
3.7.12. IdDateTimeStamp.pas .....	588
3.7.13. IdDayTime.pas .....	588
3.7.14. IdDayTimeServer.pas .....	588
3.7.15. IdDICTServer.pas .....	588
3.7.16. IdDiscardServer.pas .....	589
3.7.17. IdDNSResolver.pas .....	589
3.7.18. IdEcho.pas .....	589
3.7.19. IdEchoServer.pas .....	589
3.7.20. IdEmailAddress.pas .....	590
3.7.21. IdException.pas .....	590
3.7.22. IdFinger.pas .....	590
3.7.23. IdFingerServer.pas .....	590
3.7.24. IdFTP.pas .....	591
3.7.25. IdGlobal.pas .....	591
3.7.26. IdGopher.pas .....	591
3.7.27. IdGopherConsts.pas .....	591
3.7.28. IdGopherServer.pas .....	592
3.7.29. IdHeaderCoder.pas .....	592
3.7.30. IdHeaderList.pas .....	592
3.7.31. IdHostnameServer.pas .....	592
3.7.32. IdHTTP.pas .....	593
3.7.33. IdHTTPServer.pas .....	593
3.7.34. IdIcmpClient.pas .....	593
3.7.35. IdIMAP4Server.pas .....	593
3.7.36. IdIntercept.pas .....	594

3.7.37. IdIPWatch.pas ..... 594

3.7.38. IdIrcServer.pas ..... 594

3.7.39. IdLogBase.pas ..... 594

3.7.40. IdLogDebug.pas ..... 595

3.7.41. IdMappedPortTCP.pas ..... 595

3.7.42. IdMessage.pas ..... 595

3.7.43. IdMessageClient.pas ..... 595

3.7.44. IdMIMETypes.pas ..... 596

3.7.45. IdNetworkCalculator.pas ..... 596

3.7.46. IdNNTP.pas ..... 596

3.7.47. IdNNTPServer.pas ..... 596

3.7.48. IdPOP3.pas ..... 597

3.7.49. IdQotd.pas ..... 597

3.7.50. IdQotdServer.pas ..... 597

3.7.51. IdRawBase.pas ..... 597

3.7.52. IdRawClient.pas ..... 598

3.7.53. IdRawFunctions.pas ..... 598

3.7.54. IdRawHeaders.pas ..... 598

3.7.55. IdResourceStrings.pas ..... 598

3.7.56. IdSimpleServer.pas ..... 598

3.7.57. IdSMTP.pas ..... 599

3.7.58. IdSNTP.pas ..... 599

3.7.59. IdSocketHandle.pas ..... 599

3.7.60. IdSocks.pas ..... 599

3.7.61. IdSSLIntercept.pas ..... 600

3.7.62. IdSSLOpenSSL.pas ..... 600

3.7.63. IdStack.pas ..... 600

3.7.64. IdStackConsts.pas ..... 600

3.7.65. IdStackWinsock.pas ..... 600

3.7.66. IdTCPClient.pas ..... 601

3.7.67. IdTCPConnection.pas ..... 601

3.7.68. IdTCPServer.pas ..... 601

3.7.69. IdTelnet.pas ..... 601

3.7.70. IdTelnetServer.pas ..... 602

3.7.71. IdThread.pas ..... 602

3.7.72. IdThreadMgr.pas ..... 602

3.7.73. IdThreadMgrDefault.pas ..... 602

3.7.74. IdThreadMgrPool.pas ..... 603

3.7.75. IdTime.pas ..... 603

3.7.76. IdTimeServer.pas ..... 603

3.7.77. IdTrivialFTP.pas ..... 603

3.7.78. IdTrivialFTPBase.pas ..... 604

3.7.79. IdTrivialFTPServer.pas ..... 604

3.7.80. IdTunnelCommon.pas ..... 604

3.7.81. IdTunnelMaster.pas ..... 604

3.7.82. IdTunnelSlave.pas ..... 605

3.7.83. IdUDPBase.pas ..... 605

3.7.84. IdUDPClient.pas ..... 605

3.7.85. IdUDPServer.pas ..... 605

3.7.86. IdURI.pas ..... 606

3.7.87. IdVCard.pas ..... 606

3.7.88. IdWhois.pas ..... 606

3.7.89. IdWholsServer.pas ..... 606

3.7.90. IdWinsock.pas ..... 607

Index ..... 609

# 1. Welcome

Hello and welcome to the Internet Direct (Indy) documentation.

Internet Direct (Indy) is an open source internet component suite offering popular internet protocols. Indy is a development library based entirely on blocking sockets, and is available for the Borland Delphi, C++ Builder, and Kylix development platforms.

# 2. Introduction to Internet Direct

## 2.1. Introduction to Internet Direct

This section includes topics to familiarize members of the Open Source community with the Internet Direct (Indy) project and organization.

## 2.2. Internet Direct Credits

### Indy Project Organizer

- Chad Hower (Kudzu)

### Mercury Team

The Mercury Team is the Indy "Steering Committee", and makes all the executive decisions affecting the project.

- Chairperson: Chad Z. Hower (Kudzu)
- Vice-Chairperson: Hadi Hariri

Mercury Team Members (Alphabetically by last name):

- Mark Holmes
- Gregor Ibic
- Jim Gunkel
- Stephane Grobety
- Andrew Mee
- Rune Moberg
- J. Peter Mugaas
- Allen O'Neill
- Don Siders
- Charles Stack

**Indy Core Team (aka the "Indy Pit Crew")**

The Indy Core Team is the team that actively works on the core. The Core Team works both on its own as well as with members of the Dev Team. All changes that are merged into the Indy core are done via this Team.

Members of the Indy Core Team are chosen from members of the Dev Team from time to time. If you want to become a part of the core team, join the Dev Team and get active!

- Chairperson: Chad Z. Hower (Kudzu)
- Vice-Chairperson: Hadi Hariri

Indy Core Team Members (Alphabetically by last name):

- Mark Holmes
- Gregor Ibic
- Jim Gunkel
- Stephane Grobety
- Andrew Mee
- Rune Moberg
- J. Peter Mugaas
- Allen O'Neill
- Don Siders
- Charles Stack

**Indy Dev Team**

The Indy Dev Team is for those interested in contributing and/or discussing issues for the current Indy core. The Dev Team has read only access to the current core as well as direct access to members of the Core Team. Any one interested in developing for the Indy core should start here. This team is also open to lurkers and learners. Contributions are not required.

- Chairperson: Chad Z. Hower (Kudzu)
- Vice-Chairperson: Hadi Hariri

**Indy Demo Team**

The Demo Team is for those interested in contributing and/or discussing issues for the current Indy Demos.

- Chairperson: Allen O'Neill

**Indy Docs Team**

The Indy Docs Team is a volunteer effort to provide documentation for the Internet Direct (Indy) component suite. The Docs Team is for those interested in contributing and/or discussing issues for the Indy Documentation.

- Chairperson: Don Siders

Indy Docs Team Members (Alphabetically by last name):

- Chad Hower
- Stephane Grobety
- Hadi Hariri
- J. Peter Mugaas

**Indy Distribution Team**

The Indy Distribution Team is the team that prepares Indy source code and help files releases for distribution to the Open Source community.

- Chairperson: J. Peter Mugaas

**Related Topic Groups**

**2.3. Internet Direct Sponsors**

**Nevrona Designs**

Nevrona Designs is the primary Corporate sponsor for the Open Source Internet Direct project. Nevrona Designs is a producer of tools for Object Pascal and C++ developers that

include:

- Rave Visual and Code-Based Reporting Tool
- ReportPrinter Pro
- ND-IntraWeb
- IndyPro
- AdHocery
- ND-Patterns
- ND-Source

Nevrona Designs also provides World-Wide-Web and FTP hosting for the Internet Direct project.

**ToolsFactory**

ToolsFactory is the developer of the Doc-O-Matic application, used by the Indy Docs project to produce the documentation for Internet Direct.

Doc-O-Matic is a source code documentation system that makes documentation of source code very easy. It provides an intuitive interface for managing documentation projects and it creates powerful documentation in various different formats such as HTML, HTMLHelp, Windows Help, RTF and PDF (PDF is not supported at the moment).

Doc-O-Matic currently supports C++ and Object Pascal.

2.4. Technical Support

**Where can I get technical support?**

Indy technical support is available from several sources. Consult the Indy web site at <http://www.nevrona.com/indy> for updated Documentation, FAQs, and related Indy Articles.

Free technical support is also available in the Borland developer community by accessing the Borland Newsgroups at <news://newsgroups.borland.com>. Team Indy monitors the *borland.public.delphi.internet.winsock* and *borland.public.cppbuilder.internet* newsgroups on a regular basis. This also allows the Open Source community to benefit from your posts. Paid Priority technical support is also available from Nevrona Design in their IndyPro technical

support subscription services. If you are interested in professional, timely technical support for Indy, Nevrona Design provides several annual subscription plans to help you get over the bumps and pitfalls of Internet programming.

**Problem and Bug Reporting**

Problems and Bugs Reports should be directed to the Indy Bug List.

2.5. What Is Internet Direct?

**What Is Indy?**

Internet Direct (Indy) is an open source internet component suite comprised of popular internet protocols. Both client and server implementations are included as well as full source code and comprehensive demos.

Indy is an open source internet development library for the Borland Delphi, C++ Builder, and Kylix product lines, and is based entirely on blocking sockets. The Indy Client components are easy to use because you write your transactions in a sequence. The Indy Servers are multithreaded.

**The Indy Licenses**

Indy is dual licensed. You can review which license better suits your needs, and use that license. You can even change your mind later if you have previously chosen one.

**The Indy BSD License**

The Indy BSD license is a very no nonsense license that allows you to do almost anything you want with Indy, provided you provide proper attribution. The Indy BSD License can be found at <http://www.nevrona.com/Indy/BSDLicense.html>.

**The Indy MPL License**

To make it easier and consistent for JEDI users, we also offer an MPL v1.1 (Mozilla Public License). The Indy MPL License can be found at <http://www.mozilla.org/MPL/MPL-1.1.html>. Mozilla is the Open Source initiative formulated by Netscape for the next generation of their



web browsers. Netscape states "We believe this license satisfies the Debian Free Software Guidelines which provide a commonly accepted definition of "free software," much like other free software licenses such as GPL or BSD."

Project JEDI's implementation of the MPL allows developers to use its code in their applications ("Larger Work") regardless of whether the intended distribution will be in the public domain or as commercial applications, as long as the license conditions are met. For a more detailed explanation, an annotated version of the MPL is available.

### 3. Symbol Reference

#### 3.1. Classes

##### 3.1.1. EldAcceptWaitCannotBeModifiedWhileServerIsActive

Exception raised when the accept wait value is modified for an active server.

```
EIdAcceptWaitCannotBeModifiedWhileServerIsActive =  
class(EIdTCPServerError)
```

**Description**

EldAcceptWaitCannotBeModifiedWhileServerIsActive is an EIdTCPServerError *EIdTCPServerError* exception descendant raised when the TIdTCPServer.AcceptWait property is modified while the server is Active.

##### 3.1.2. EldAlreadyConnected

Exception raised when connecting using an already open connection.

```
EIdAlreadyConnected = class(EIdException)
```

**Description**

EIdAlreadyConnected is an exception raised when you attempt to make a connection with an Indy component and there is already an active connection.

##### 3.1.3. EldCanNotChangeTarget

```
EIdCanNotChangeTarget = class(EIdException)
```

**Description**

This exception class is raised if you attempt to set the TIdLogDebug.Target property while the TIdLogDebug.Active property is true. This exception is only raised in the Borland IDE and when the component is loading from a stream.

### 3.1.4. EldCanNotCreateMessagePart

Indicates an error in the TldMessagePart *↗TldMessagePart* constructor.

```
EIdCanNotCreateMessagePart = class(EIdMessageException)
```

**Description**

EIdCanNotCreateMessagePart is an EldMessageException *↗EldMessageException* descendant used to identify an error in the constructor for a TldMessagePart *↗TldMessagePart* instance. This exception generally indicates that the constructor has been called with the abstract base class TldMessagePart *↗TldMessagePart* instead of a TldText *↗TldText* or TldAttachment *↗TldAttachment* descendant.

### 3.1.5. EldClosedSocket

Exception for writing to a closed TCP socket.

```
EIdClosedSocket = class(EIdException)
```

**Description**

EIdClosedSocket is the exception raised when an application attempts to write to a closed TCP socket.

### 3.1.6. EldConnClosedGracefully

Exception raised when a connection has close gracefully.

```
EIdConnClosedGracefully = class(EIdSilentException)
```

**Description**

This exception class is raised when a connection has been closed gracefully. Because it descends from EldSilentException *↗EldSilentException*, it behaves like the VCL EAbort exception. We recommend that you set your IDE to ignore those exceptions by clicking

Tools|Debugger Options..|Language Exceptions|Add and entering "EldSilentException".

### 3.1.7. EldCorruptServicesFile

This exception class is raised when the SERVICES file is corrupt.

```
EIdCorruptServicesFile = class(EIdException)
```

**Description**

This exception class is raised by IdPorts *↗IdPorts* function when it could not read the services file. The services file is usually located in different places depending upon the Operating System:

- BSD Unix - /etc/
- Windows 95, 98, and Me - Windows directory
- Windows NT - Windowssystem32driversetc

### 3.1.8. EldCouldNotBindSocket

Exception when binding a socket descriptor.

```
EIdCouldNotBindSocket = class(EIdSocketHandleError)
```

**Description**

EIdCouldNotBindSocket is an EldSocketHandleError *↗EldSocketHandleError* descendant raised when TldSocketHandle.Bind encounters an error while binding the protocol family, address, and port number for the socket descriptor.

### 3.1.9. EldDnsResolverError

Exception raised to indicate TldDNSResolver *↗TldDNSResolver* errors.

```
EIdDnsResolverError = class(EIdException)
```

**Description**

EldDnsResolverError is an exception raised by TIdDNSResolver *↗TIdDNSResolver* so that it can only respond to those errors.

**3.1.10. EldEldTunnelConnectToMasterFailed**

EIdEldTunnelConnectToMasterFailed = **class**(EIdTunnelException)

**Description**

The text for this class has been generated automatically. This means that it is not documented.

**3.1.11. EldException**

Base Exception class for Indy.

EIdException = **class**(Exception)

**Description**

EIdException is an Exception that is the Base Exception class for all Indy Exception descendants. Indy does not raise the standard Exception class.

**3.1.12. EldFailedToRetreiveTimeZoneInfo**

This exception class is raised when the Time Zone could not be retrieved.

EIdFailedToRetreiveTimeZoneInfo = **class**(EIdException)

**Description**

This exception class is raised by OffsetFromUTC *↗OffsetFromUTC* when Time Zone information could not be retrieved from the user's computer.

**3.1.13. EldFTPFileAlreadyExists**

Exception raised a file cannot be overwritten in an FTP Get request.

EIdFTPFileAlreadyExists = **class**(EIdException)

**Description**

This exception is raised if the TIdFTP.Get method is not permitted to overwrite a file and the file you specified in the destination already exists on the user's computer. Usually, you can trap this exception in your code or if you intended to overwrite a file on the user's computer, you can use code such as:

FTP1.Get('ASampleFile.TXT','ASampleFile.TXT',True);

**3.1.14. EldHTTPCannotSwitchSessionStateWhenActive**

HTTP server exception for changing state on an active server.

EIdHTTPCannotSwitchSessionStateWhenActive = **class**(EIdHTTPServerError)

**Description**

EIdHTTPCannotSwitchSessionStateWhenActive is an EldHTTPServerError *↗EldHTTPServerError* descendant raised when a HTTP server attempts to change the value of SessionState on an Active server.

**3.1.15. EldHTTPErrorParsingCommand**

HTTP server exception for an unknown HTTP command.

EIdHTTPErrorParsingCommand = **class**(EIdHTTPServerError)

**Description**

EIdHTTPErrorParsingCommand is an EldHTTPServerError *↗EldHTTPServerError* exception descendant raised when a HTTP server is unable to determine the HTTP command from a client connection.

### 3.1.16. EldHTTPHeaderAlreadyWritten

HTTP response exception raised during writing headers.

```
EIdHTTPHeaderAlreadyWritten = class(EIdHTTPServerError)
```

**Description**

EIdHTTPHeaderAlreadyWritten is an EldHTTPServerError *↗EldHTTPServerError* exception descendant that represents the exception raised when TIdHTTPResponseInfo *↗TIdHTTPResponseInfo* attempts to write headers for a response where the headers have already been written.

### 3.1.17. EldHTTPServerError

Ancestor for Indy HTTP exceptions.

```
EIdHTTPServerError = class(EIdException)
```

**Description**

EIdHTTPServerError is an EldException *↗EldException* descendant that is the ancestor for all HTTP-related exceptions in Indy.

### 3.1.18. EldHTTPUnsupportedAuthorisationScheme

HTTP server exception for an invalid authentication scheme.

```
EIdHTTPUnsupportedAuthorisationScheme = class(EIdHTTPServerError)
```

**Description**

EIdHTTPUnsupportedAuthorisationScheme is an EldHTTPServerError *↗EldHTTPServerError* exception descendant raised when a HTTP server has received a request with an invalid Authorization header. When "Authorization" is used in the request, it must contain the "Basic"

authentication type with Base64-encoded user and password information.

### 3.1.19. EldIcmpException

Exception for ICMP components.

```
EIdIcmpException = class(EIdException)
```

**Description**

EldIcmpException is an exception type raised by the Indy ICMP components.

### 3.1.20. EldInterceptPropInvalid

This exception class is raised whenever an there is an attempt to use a capability that is not supported by the current TIdConnectionIntercept *↗TIdConnectionIntercept* that was assigned to the Intercept property.

```
EIdInterceptPropInvalid = class(EIdTCPConnectionError)
```

**Description**

This exception class is raised whenever an there is an attempt to use a capability that is not supported by the current TIdConnectionIntercept *↗TIdConnectionIntercept* that was assigned to the Intercept property. This can occur if you attempt to use a TIdHTTP *↗TIdHTTP* component access a secure HTTP site (HTTPS) while the TIdHTTP's Intercept property is not set to a TIdSSLConnectionIntercept *↗TIdSSLConnectionIntercept* descendant.

### 3.1.21. EldInterceptProplsNil

Exception raised when attempting to enable a Nil Intercept.

```
EIdInterceptPropIsNil = class(EIdTCPConnectionError)
```

**Description**

This exception class is raised whenever an attempt to set the InterceptEnabled to true while

Intercept is nil. To enable intercepts, you first must set the Intercept property to a TldConnectionIntercept *↗TldConnectionIntercept* descendant before setting the InterceptEnabled property to true.

3.1.22. EldInvalidServiceName

Exception raised for an invalid port service.

```
EIdInvalidServiceName = class(EIdException)
```

**Description**  
EldInvalidServiceName is an EldException *↗EldException* descendant that is raised for an invalid port service.

3.1.23. EldInvalidSocket

Exception raised for a connection closed unexpectedly.

```
EIdInvalidSocket = class(EIdException)
```

**Description**  
EldInvalidSocket is an Exception type raised when a connection was closed unexpectedly.

3.1.24. EldLoginException

```
EIdLoginException = class(EIdTelnetServerException)
```

**Description**  
This exception class is the ancestor class for TldTelnetServer *↗TldTelnetServer* exceptions related to logging into the server. This exception is handled internally in the TldTelnetServer *↗TldTelnetServer*.

3.1.25. EldMaxLoginAttempt

```
EIdMaxLoginAttempt = class(EIdLoginException)
```

**Description**  
This exception class is raised if a user has exceeded the limit of login attempts permitted by the TldTelnetServer.LoginAttempts property.  
This exception is handled internally in the TldTelnetServer *↗TldTelnetServer*.

3.1.26. EldMessageException

Ancestor class for TldMessage *↗TldMessage* and TldMessagePart *↗TldMessagePart* exceptions.

```
EIdMessageException = class(EIdException)
```

**Description**  
EldMessageException is an EldException *↗EldException* exception descendant that is the ancestor class for TldMessage *↗TldMessage* and TldMessagePart *↗TldMessagePart* exceptions.

3.1.27. EldMoreThanOneTldAntiFreeze

This exception is raised whenever there is an attempt to create more than one TldAntiFreeze component in your application

```
EIdMoreThanOneTldAntiFreeze = class(EIdException)
```

**Description**  
EldMoreThanOneTldAntiFreeze

EldMoreThanOneTldAntiFreeze is raised when ever there is an attempt to create more than one TldAntiFreeze *↗TldAntiFreeze* instance in an application. Your program is permitted to have **only** one TldAntiFreeze *↗TldAntiFreeze* component. You do not need more than one TldAntiFreeze *↗TldAntiFreeze* component in your applications.

### 3.1.28. EldNNTPConnectionRefused

Indicates a connection to the NNTP server has been refused.

```
EIdNNTPConnectionRefused = class(EIdProtocolReplyError)
```

**Description**

EIdNNTPConnectionRefused is an EldProtocolReplyError *↗EldProtocolReplyError* descendant raised when an the TIdNNTP.Connect method receives the 502 NNTP response code.

### 3.1.29. EldNNTPException

Ancestor class for NNTP exceptions.

```
EIdNNTPException = class(EIdException)
```

**Description**

EIdNNTPException is an EIdException *↗EIdException* descendant that is the ancestor of Indy NNTP exceptions.

### 3.1.30. EldNNTPNoOnNewGroupsList

NNTP exception for retrieval of the new Newsgroups list.

```
EIdNNTPNoOnNewGroupsList = class(EIdNNTPException)
```

**Description**

EIdNNTPNoOnNewGroupsList is an EldNNTPException *↗EldNNTPException* exception descendant raised when TIdNNTP.GetNewGroupsList is called and the TIdNNTP.OnNewGroupsList event handler has not been assigned.

### 3.1.31. EldNNTPNoOnNewNewsList

Exception for retrieval of the new NNTP new message list.

```
EIdNNTPNoOnNewNewsList = class(EIdNNTPException)
```

**Description**

EIdNNTPNoOnNewNewsList is an EldNNTPException *↗EldNNTPException* descendant raised when TIdNNTP.GetNewNewsList is called and the TIdNNTP.OnNewNewsList event handler has not been assigned.

### 3.1.32. EldNNTPNoOnNewsgroupList

Exception for NNTP newsgroup list retrieval.

```
EIdNNTPNoOnNewsgroupList = class(EIdNNTPException)
```

**Description**

EIdNNTPNoOnNewsgroupList is an EldNNTPException *↗EldNNTPException* exception descendant raised when TIdNNTP.GetNewsgroupList is called and a TIdNNTP.OnNewsgroupList event handler has not been assigned.

### 3.1.33. EldNNTPStringListNotInitialized

Exception for unsuccessful message retrieval.

```
EIdNNTPStringListNotInitialized = class(EIdNNTPException)
```

**Description**

EIdNNTPStringListNotInitialized is an EldNNTPException *↗EldNNTPException* exception descendant raised when TIdNNTP.SendCheck is called and the TStringList container for messages has not been initialized.

### 3.1.34. EldNoBindingsSpecified

Exception raised creating the listener thread socket handle.

```
EIdNoBindingsSpecified = class(EIdUDPServerException)
```

**Description**

EldNoBindingsSpecified is an EldUDPServerException *↗EldUDPServerException* exception raised when a UDP server attempts to initialize the socket handle allocated for the server listener thread and the default port number has not been assigned.

### 3.1.35. EldNoDataToRead

Exception raised when writing an empty TStream with WriteStream.

```
EIdNoDataToRead = class(EIdTCPConnectionError)
```

**Description**

This exception is raised when you attempt to send a Stream through WriteStream that is empty (contains no data).

### 3.1.36. EldNoExecuteSpecified

Exception raised after unsuccessful peer thread execution.

```
EIdNoExecuteSpecified = class(EIdTCPServerError)
```

**Description**

EldNoExecuteSpecified is an EldTCPServerError *↗EldTCPServerError* descendant raised when TIdPeerThread *↗TIdPeerThread* executes in the server context for the connection, and returns False as a result of the method execution.

### 3.1.37. EldNoOnAuthentication

```
EIdNoOnAuthentication = class(EIdTelnetServerException)
```

**Description**

This exception class is raised if there is no OnAuthentication event in the TIdTelnetServer *↗TIdTelnetServer*. To avoid this problem, please use the OnAuthentication event. This exception is handled internally in the TIdTelnetServer *↗TIdTelnetServer*.

### 3.1.38. EldNotAllBytesSent

Exception for an incomplete transmission.

```
EIdNotAllBytesSent = class(EIdSocketHandleError)
```

**Description**

EldNotAllBytesSent is an EldSocketHandleError *↗EldSocketHandleError* exception descendant raised when TIdSocketHandle.SendTo is unable to send all bytes in a transmission to the peer connection.

### 3.1.39. EldNotEnoughDataInBuffer

Exception raised when there are not enough bytes in TIdTCPConnection's internal buffer.

```
EIdNotEnoughDataInBuffer = class(EIdTCPConnectionError)
```

**Description**

This exception is raised when there are not enough bytes in the internal buffer for TIdTCPConnection *↗TIdTCPConnection* for an operation such as ExtractXBytesFromBuffer and RemoveXBytes.

### 3.1.40. EldObjectTypeNotSupported

Exception raised when Capture can not support an object as a Destination.

```
EIdObjectTypeNotSupported = class(EIdTCPConnectionError)
```

**Description**

This exception is raised when Capture is passed an object in the ADest parameter which Capture does not support. Capture only supports the following destination objects: TStream and TStrings.

### 3.1.41. EldOpenSSLError

Ancestor for Indy Open SSL error classes.

```
EIdOpenSSLError = class(EIdException)
```

**Description**

EIdOpenSSLError is an EIdException *⚡EIdException* descendant that acts the ancestor for Indy Open SSL error classes.

### 3.1.42. EldOpenSSLLoadError

Ancestor class for error conditions related to loading the OpenSSL library.

```
EIdOpenSSLLoadError = class(EIdOpenSSLError)
```

**Description**

EIdOpenSSLLoadError is an EIdOpenSSLError *⚡EIdOpenSSLError* descendant that acts as the ancestor class for error conditions related to loading the OpenSSL library.

### 3.1.43. EldOSSLAcceptError

Exception raised when a new SSL connection cannot be accepted.

```
EIdOSSLAcceptError = class(EIdOpenSSLError)
```

**Description**

EIdOSSLAcceptError is an EIdOpenSSLError *⚡EIdOpenSSLError* descendant raised when an error occurs while attempting to accept an new Secure Socket *⚡Socket* Layer connection in TIdSSLSocket.Accept.

### 3.1.44. EldOSSLConnectError

Exception raised for SSL connection errors.

```
EIdOSSLConnectError = class(EIdOpenSSLError)
```

**Description**

EIdOSSLConnectError is an EIdOpenSSLError *⚡EIdOpenSSLError* descendant raised when an error occurs during connection for a Secure Socket *⚡Socket* Layer handle in TIdSSLSocket.Connect.

### 3.1.45. EldOSSLCouldNotLoadSSLLibrary

Error raised when the OpenSSL library cannot be loaded.

```
EIdOSSLCouldNotLoadSSLLibrary = class(EIdOpenSSLLoadError)
```

**Description**

EIdOSSLCouldNotLoadSSLLibrary is an EIdOpenSSLLoadError *⚡EIdOpenSSLLoadError* descendant that indicates when a error has occurred while loading the OpenSSL library. EIdOSSLCouldNotLoadSSLLibrary can be raised from the TIdSSLContext.Create method.



### 3.1.46. EldOSSLCreatingContextError

Exception raised when recreating the SSL content after a change in mode.

```
EIdOSSLCreatingContextError = class(EIdOpenSSLSError)
```

**Description**

EIdOSSLCreatingContextError is an EldOpenSSLSError *↗EldOpenSSLSError* descendant raised when an error occurs while recreating the SSL context after a change to TIdSSLContext.Mode.

### 3.1.47. EldOSSLDatabindingError

Exception raised when an application cannot bind to the SSL socket.

```
EIdOSSLDatabindingError = class(EIdOpenSSLSError)
```

**Description**

EIdOSSLDatabindingError is an EldOpenSSLSError *↗EldOpenSSLSError* descendant raised when an error occurs while attempting to bind to a SSL socket for the purpose of providing data to the application in TIdSSLSocket.Accept.

### 3.1.48. EldOSSLGetMethodError

Exception raised when a SSL context contains an invalid Mode.

```
EIdOSSLGetMethodError = class(EIdOpenSSLSError)
```

**Description**

EIdOSSLGetMethodError is an EldOpenSSLSError *↗EldOpenSSLSError* raised when TIdSSLContext *↗TIdSSLContext* contains an invalid Mode value.

### 3.1.49. EldOSSLLoadingCertError

Exception raised when certificate file(s) cannot be loaded for an SSL context.

```
EIdOSSLLoadingCertError = class(EIdOpenSSLLoadError)
```

**Description**

EIdOSSLLoadingCertError is an EldOpenSSLLoadError *↗EldOpenSSLLoadError* raised when an error occurs while loading a RootCertFile (Root certificate file) or CertFile (certificate file) after a change to the Mode property for TIdSSLContext *↗TIdSSLContext*.

### 3.1.50. EldOSSLLoadingKeyError

Exception raised when the key file cannot be opened for an SSL context.

```
EIdOSSLLoadingKeyError = class(EIdOpenSSLLoadError)
```

**Description**

EIdOSSLLoadingKeyError is an EldOpenSSLLoadError *↗EldOpenSSLLoadError* descendant raised when an error occurs while attempting to open KeyFile after a change to the Mode property in TIdSSLContext *↗TIdSSLContext*.

### 3.1.51. EldOSSLLoadingRootCertError

Exception raised when loading a root certificate file for an SSL context.

```
EIdOSSLLoadingRootCertError = class(EIdOpenSSLLoadError)
```

**Description**

EIdOSSLLoadingRootCertError is an EldOpenSSLLoadError *↗EldOpenSSLLoadError* descendant raised when an error occurs while loading the RootCertFile (Root certificate file) for an SSL context after a change to the Mode property.

### 3.1.52. EldOSSLMoDeNotSet

Exception raised for an unassigned SSL context mode.

EldOSSLMoDeNotSet = **class**(EldOpenSSLError)

**Description**

EldOSSLMoDeNotSet is an EldOpenSSLError *↗EldOpenSSLError* descendant raised when the Mode for the SSL context is unassigned.

### 3.1.53. EldOSSLSettiNgCiPhErError

Exception raised after an error while setting the SSL context cipher schemes.

EldOSSLSettiNgCiPhErError = **class**(EldOpenSSLError)

**Description**

EldOSSLSettiNgCiPhErError is an EldOpenSSLError *↗EldOpenSSLError* descendant raised when an error occurs while setting the list of Ciphers for SSL context after a change in Mode.

### 3.1.54. EldPackaGeSiZeTooBiG

Exception raised for an invalid datagram size.

EldPackaGeSiZeTooBiG = **class**(EldSocketHandleError)

**Description**

EldPackaGeSiZeTooBiG is an EldSocketHandleError *↗EldSocketHandleError* exception descendant raised when TIdSocketHandle.SendTo receives the Id\_WSAEMSGSIZE *↗Id\_WSAEMSGSIZE* protocol stack error after transmission of a datagram.

### 3.1.55. EldProtocolReplyError

Exception type for protocol errors.

EldProtocolReplyError = **class**(EIdException)

**Description**

EldProtocolReplyError is an exception type raised for protocol errors, such as the HTTP 404 error.

### 3.1.56. EldResponseError

Exception for errors in an expected response.

EldResponseError = **class**(EIdException)

**Description**

EldResponseError is a EIdException *↗EIdException* descendant that represents the exception raised when an expected response has not been received during a protocol operation.

### 3.1.57. EldSetSiZeExceeded

Represents an FD SET error.

EldSetSiZeExceeded = **class**(EIdException)

**Description**

EldSetSiZeExceeded is an EIdException *↗EIdException* descendant that is raised when the number of FD SET calls has exceeded the maximum allowed for the operating platform.

### 3.1.58. EldSilentException

Exceptions that behave like VCL EAbort.

```
EIdSilentException = class(EIdException)
```

**Description**

EIdSilentException is a exception class is for exceptions which should behave like the VCL EAbort exception. We recommend that you set your IDE to ignore those exceptions by clicking Tools|Debugger Options..|Language Exceptions|Add and entering "EIdSilentException".

3.1.59. EIdSocketError

Exception for Socket *↗*Socket errors.

```
EIdSocketError = class(EIdException)
```

**Description**

EIdSocketError is an exception class raised when a socket error occurs. Last Error is the number code for that error.

3.1.60. EIdSocketHandleError

Ancestor exception for Indy socket handle errors.

```
EIdSocketHandleError = class(EIdException)
```

**Description**

EIdSocketHandleError is an EIdException *↗EIdException* descendant that is the ancestor class for Indy TIdSocketHandle *↗TIdSocketHandle* errors.

3.1.61. EIdSocksAuthError

This exception class is raised if the SOCKS proxy reports an authentication error.

```
EIdSocksAuthError = class(EIdSocksError)
```

**Description**

This exception class is raised if the SOCKS proxy reports an authentication error.

3.1.62. EIdSocksAuthMethodError

This exception class is raised if the SOCKS proxy reports an invalid or unsupported authentication method.

```
EIdSocksAuthMethodError = class(EIdSocksError)
```

**Description**

This exception class is raised if the SOCKS proxy reports an invalid or unsupported authentication method.

3.1.63. EIdSocksError

This exception class is the ancestor for exception classes that are raised due to SOCKS errors.

```
EIdSocksError = class(EIdException)
```

**Description**

This exception class is the ancestor for exception classes that are raised due to SOCKS errors that occur when using TIdTCPClient.Connect. Descendants include EIdSocksRequestFailed *↗EIdSocksRequestFailed*, EIdSocksRequestServerFailed *↗EIdSocksRequestServerFailed*, EIdSocksRequestIdntFailed *↗EIdSocksRequestIdntFailed*, EIdSocksUnknownError *↗EIdSocksUnknownError*, EIdSocksServerRespondError *↗EIdSocksServerRespondError*, EIdSocksAuthMethodError *↗EIdSocksAuthMethodError*, EIdSocksAuthError *↗EIdSocksAuthError*, EIdSocksServerGeneralError *↗EIdSocksServerGeneralError*, EIdSocksServerPermissionError *↗EIdSocksServerPermissionError*, EIdSocksServerNetUnreachableError *↗EIdSocksServerNetUnreachableError*, EIdSocksServerHostUnreachableError *↗EIdSocksServerHostUnreachableError*, EIdSocksServerConnectionRefusedError *↗EIdSocksServerConnectionRefusedError*, EIdSocksServerTTLExpiredError *↗EIdSocksServerTTLExpiredError*, EIdSocksServerCommandError *↗EIdSocksServerCommandError*, and EIdSocksServerAddressError *↗EIdSocksServerAddressError*.

### 3.1.64. EldSocksRequestFailed

This exception class is raised if the SOCKS proxy rejected or could not fulfill a request from TIdTCPClient *↗TIdTCPClient*.

```
EIdSocksRequestFailed = class(EIdSocksError)
```

**Description**

This exception class is raised if the SOCKS proxy rejected or could not fulfill a request from TIdTCPClient *↗TIdTCPClient*.

### 3.1.65. EldSocksRequestIdentFailed

This exception class is raised if the SOCKS proxy indicates that the User ID that the TIdTCPClient *↗TIdTCPClient* sent does not match the User ID provided by an Ident Daemon running on the user's computer.

```
EIdSocksRequestIdentFailed = class(EIdSocksError)
```

**Description**

This exception class is raised if the SOCKS proxy indicates that the User ID that the TIdTCPClient *↗TIdTCPClient* sent does not match the User ID provided by an Ident Daemon running on the user's computer. Ident (RFC 1413) is like a "caller ID" for the internet where a server will connect to a client computer and request the user ID for the current connection.

### 3.1.66. EldSocksRequestServerFailed

This exception class is raised if the SOCKS proxy could not make connection requested by the TIdTCPClient *↗TIdTCPClient*.

```
EIdSocksRequestServerFailed = class(EIdSocksError)
```

**Description**

>This exception class is raised if the SOCKS proxy could not make connection requested by the TIdTCPClient *↗TIdTCPClient*.

### 3.1.67. EldSocksServerAddressError

This exception class is raised if the SOCKS proxy reports that it does not support the address type sent by TIdTCPClient *↗TIdTCPClient*.

```
EIdSocksServerAddressError = class(EIdSocksError)
```

**Description**

This exception class is raised if the SOCKS proxy reports that it does not support the address type sent by TIdTCPClient *↗TIdTCPClient*.

### 3.1.68. EldSocksServerCommandError

This exception class is raised if the SOCKS proxy reports that it does not support the command sent by TIdTCPClient *↗TIdTCPClient*.

```
EIdSocksServerCommandError = class(EIdSocksError)
```

**Description**

This exception class is raised if the SOCKS proxy reports that it does not support the command sent by TIdTCPClient *↗TIdTCPClient*.

### 3.1.69. EldSocksServerConnectionRefusedError

This exception class is raised if the SOCKS proxy reports that the host specified by TIdTCPClient *↗TIdTCPClient* refused the connection request sent by the SOCKS proxy.

```
EIdSocksServerConnectionRefusedError = class(EIdSocksError)
```

**Description**

This exception class is raised if the SOCKS proxy reports that the host specified by TIdTCPClient *↗TIdTCPClient* refused the connection request sent by the SOCKS proxy. This

usually means that the host computer is not running a server that is listening on the port specified in TldTCPClient.Port.

3.1.70. EldSocksServerGeneralError

This exception class is raised if the SOCKS proxy reports a general failure.

```
EIdSocksServerGeneralError = class(EIdSocksError)
```

Description

This exception class is raised if the SOCKS proxy reports a general failure.

3.1.71. EldSocksServerHostUnreachableError

This exception class is raised if the SOCKS proxy reports that the host specified by TldTCPClient *⌘TldTCPClient* is unreachable.

```
EIdSocksServerHostUnreachableError = class(EIdSocksError)
```

Description

This exception class is raised if the SOCKS proxy reports that the host specified by TldTCPClient *⌘TldTCPClient* is unreachable from the network. This can be due to network conditions between the SOCKS proxy and server being requested by TldTCPClient *⌘TldTCPClient*.

3.1.72. EldSocksServerNetUnreachableError

This exception class is raised if the SOCKS proxy reports that the network is unreachable.

```
EIdSocksServerNetUnreachableError = class(EIdSocksError)
```

Description

This exception class is raised if the SOCKS proxy reports that the network is unreachable.

3.1.73. EldSocksServerPermissionError

This exception class is raised if the SOCKS proxy reports that a connection is not allowed by the ruleset.

```
EIdSocksServerPermissionError = class(EIdSocksError)
```

Description

This exception class is raised if the SOCKS proxy reports that a connection is not allowed by the ruleset.

3.1.74. EldSocksServerRespondError

This exception class is raised if the SOCKS proxy did not send a reply to the request sent by TldTCPClient *⌘TldTCPClient*.

```
EIdSocksServerRespondError = class(EIdSocksError)
```

Description

This exception class is raised if the SOCKS proxy did not send a reply to the request sent by TldTCPClient *⌘TldTCPClient*.

3.1.75. EldSocksServerTTLExpiredError

This exception class is raised if the SOCKS proxy reports that the host specified by TldTCPClient *⌘TldTCPClient* could not be reached because the TTL was exceeded.

```
EIdSocksServerTTLExpiredError = class(EIdSocksError)
```

Description

This exception class is raised if the SOCKS proxy reports that the host specified by TldTCPClient *⌘TldTCPClient* could not be reached because the TTL (Time To Live) on the

packet the SOCKS proxy sent to the Host had expired.

3.1.76. EldSocksUnknownError

This exception class is raised if the SOCKS proxy replies to a request with an value that is not known to Internet Direct.

```
EIdSocksUnknownError = class(EIdSocksError)
```

Description

This exception class is raised if the current code does not know how to process a reply code from the SOCKS proxy. The code in Internet Direct does assume that the value means a SOCKS error.

3.1.77. EldStackCanNotLoadWinsock

```
EIdStackCanNotLoadWinsock = class(EIdException)
```

Description

The text for this class has been generated automatically. This means that it is not documented.

3.1.78. EldStackError

Ancestor class for Indy protocol stack errors.

```
EIdStackError = class(EIdException)
```

Description

EIdStackError is an EIdException *⚡EIdException* exception descendant that is the ancestor for Indy protocol stack errors.

3.1.79. EldStackInitializationFailed

Exception for protocol stack initialization failure.

```
EIdStackInitializationFailed = class(EIdStackError)
```

Description

EIdStackInitializationFailed is an EIdStackError *⚡EIdStackError* exception descendant raised when Indy is unable to initialize the protocol stack.

3.1.80. EldStackSizeExceeded

Exception raised when the maximum number of socket descriptors is exceeded.

```
EIdStackSizeExceeded = class(EIdStackError)
```

Description

EIdStackSizeExceeded is an EIdStackError *⚡EIdStackError* exception descendant raised when the protocol stack attempts to allocate a socket descriptor and the maximum number of socket descriptors has been exceeded.

3.1.81. EldTableNotFound

Error for an invalid character conversion using the coder table.

```
EIdTableNotFound = class(EIdException)
```

Description

EIdTableNotFound is an EIdException *⚡EIdException* descendant that represents the exception raised when a coder requests lookup for a character that does not appear in the CodingTable for the coder.

3.1.82. EldTCPConnectionError

Ancestor for TIdTCPConnection *⚡TIdTCPConnection* exception classes.

```
EIdTCPConnectionError = class(EIdException)
```

**Description**

This exception class is the ancestor for exceptions that occur in TIdTCPConnection *↗TIdTCPConnection* such as EldObjectTypeNotSupported *↗EldObjectTypeNotSupported*, EldNotEnoughDataInBuffer *↗EldNotEnoughDataInBuffer*, EldInterceptPropsNil *↗EldInterceptPropsNil*, EldInterceptPropInvalid *↗EldInterceptPropInvalid*, and EldNoDataToRead *↗EldNoDataToRead*.

3.1.83. EldTCPServerError

Ancestor for TCP server errors.

```
EIdTCPServerError = class(EIdException)
```

**Description**

EIdTCPServerError is an EldException *↗EldException* exception descendant that is the ancestor for Indy TCP server errors.

3.1.84. EldTelnetClientConnectError

Exception for Telnet connection errors.

```
EIdTelnetClientConnectError = class(EIdTelnetError)
```

**Description**

EldTelnetClientConnectError is an EldTelnetError *↗EldTelnetError* exception raised when an error occurs in TIdTelnet.Connect.

3.1.85. EldTelnetError

Ancestor for Indy Telnet exceptions.

```
EIdTelnetError = class(EIdException)
```

**Description**

EldTelnetError is an EldException *↗EldException* descendant that is the ancestor class for Indy Telnet exceptions.

3.1.86. EldTelnetServerException

```
EIdTelnetServerException = class(EIdException)
```

**Description**

This exception class is the ancestor for exception classes that are raised in the TIdTelnetServer *↗TIdTelnetServer*.

3.1.87. EldTelnetServerOnDataAvailableIsNil

Exception for Telnet OnDataAvailable event notification.

```
EIdTelnetServerOnDataAvailableIsNil = class(EIdTelnetError)
```

**Description**

EldTelnetServerOnDataAvailableIsNil is an EldTelnetError *↗EldTelnetError* exception descendant raised when TIdTelnet.DoOnDataAvailable is executed and no event handler has been assigned for the notification.

3.1.88. EldTextInvalidCount

Indicates an error creating text message parts.

EIdTextInvalidCount = **class**(EIdMessageException)

**Description**

EIdTextInvalidCount is a EIdMessageException *↗EIdMessageException* exception descendant that identifies when a error has occurred while creating TIdText *↗TIdText* portions of TIdMessage *↗TIdMessage* during message transmission.

3.1.89. EIdTFTPAccessViolation

FTP access violation exception.

EIdTFTPAccessViolation = **class**(EIdTFTPException)

**Description**

EIdTFTPAccessViolation is an exception type raised for FTP access violations.

3.1.90. EIdTFTPAllocationExceeded

FTP Allocation exceeded exception.

EIdTFTPAllocationExceeded = **class**(EIdTFTPException)

**Description**

EIdTFTPAllocationExceeded is an exception type raised when FTP allocation is exceeded.

3.1.91. EIdTFTPException

Trivial FTP Exception type.

EIdTFTPException = **class**(EIdException)

**Description**

EIdTFTPException is an exception type raised by the Trivial FTP Components when an error occurs.

3.1.92. EIdTFTPFileAlreadyExists

FTP exception raised when creating a file that already exists.

EIdTFTPFileAlreadyExists = **class**(EIdTFTPException)

**Description**

EIdTFTPFileAlreadyExists is an exception type raised when creating a file that already exists.

3.1.93. EIdTFTPFileNotFound

EIdTFTPFileNotFound = **class**(EIdTFTPException)

**Description**

The text for this class has been generated automatically. This means that it is not documented.

3.1.94. EIdTFTPIllegalOperation

Exception for illegal FTP operations.

EIdTFTPIllegalOperation = **class**(EIdTFTPException)

**Description**

EIdTFTPIllegalOperation is an exception type raised when an illegal FTP operation is performed.

3.1.95. EIdTFTPNoSuchUser

Represents an FTP User Authentication error.

EIdTFTPNoSuchUser = **class**(EIdTFTPException)

**Description**

EIdTFTPNoSuchUser is an EIdException *↗EIdException* descendant that represents the exception raised when an FTP USeR Authentication error has occurred.



### 3.1.96. EldTFTPOptionNegotiationFailed

Represent s the error for an invalid FTP Option Negotiation response.

```
EIdTFTPOptionNegotiationFailed = class(EIdTFTPException)
```

**Description**

EIdTFTPOptionNegotiationFailed is an EldException *⚡EldException* raised when an invalid response is detected during FTP Option Negotiation. This generally occurs when TIdFTP.BlockSize has been changed to a non-default value.

### 3.1.97. EldTFTPUnknownTransferID

FTP Exception raised for an Unknown Transfer ID.

```
EIdTFTPUnknownTransferID = class(EIdTFTPException)
```

**Description**

EIdTFTPUnknownTransferID is an EldException *⚡EldException* descendant that represents the exception raised when an unknown FTP Transfer ID has been detected.

### 3.1.98. EldThreadClassNotSpecified

Exception raised when creating a thread without a class type.

```
EIdThreadClassNotSpecified = class(EIdThreadMgrError)
```

**Description**

EIdThreadClassNotSpecified is an EldThreadMgrError *⚡EldThreadMgrError* descendant raised when TIdThreadMgr.CreateNewThread is called an no ThreadClass is assigned for the thread manager.

### 3.1.99. EldThreadMgrError

Ancestor for Indy Thread Manager exceptions.

```
EIdThreadMgrError = class(EIdException)
```

**Description**

EldThreadMgrError is an EldException *⚡EldException* exception descendant that acts as the ancestor class for Indy Thread Manager exceptions.

### 3.1.100. EldTunnelConnectToMasterFailed

Error for a failed connection from a Tunnel Slave to the Tunnel Master.

```
EIdTunnelConnectToMasterFailed = class(EIdTunnelException)
```

**Description**

EldTunnelConnectToMasterFailed is an EldTunnelException *⚡EldTunnelException* raised when a TIdTunnelSlave *⚡TIdTunnelSlave* is unable to establish a connection to the host and port number for the TIdTunnelMaster *⚡TIdTunnelMaster* server.

### 3.1.101. EldTunnelCRCFailed

Error raised when a CRC calculation fails.

```
EIdTunnelCRCFailed = class(EIdTunnelException)
```

**Description**

EldTunnelCRCFailed is an EldTunnelException *⚡EldTunnelException* raised when the CRC calculation for a packet received by TSlaveThread *⚡TSlaveThread* does not match the CRC value in the data packet.

### 3.1.102. EldTunnelCustomMessageInterpretationFailure

Error raised when a Slave thread receives an unknown message type.

```
EldTunnelCustomMessageInterpretationFailure =
class(EldTunnelException)
```

**Description**

EldTunnelCustomMessageInterpretationFailure is an EldTunnelException ⚡*EldTunnelException* descendant raised when an unknown message type is received for a TSlaveThread ⚡*TSlaveThread*.

### 3.1.103. EldTunnelDontAllowConnections

Tunnel Slave cannot accept connections for Service threads.

```
EldTunnelDontAllowConnections = class(EldTunnelException)
```

**Description**

EldTunnelDontAllowConnections is an EldTunnelException ⚡*EldTunnelException* descendant raised when a TldTunnelSlave ⚡*TldTunnelSlave* is unable to accept connections and service client threads.

### 3.1.104. EldTunnelException

Ancestor of all Indy Tunnel exceptions.

```
EldTunnelException = class(EIdException)
```

**Description**

EldTunnelException is an EIdException ⚡*EldException* descendant that is the ancestor of all Indy Tunnel exceptions.

### 3.1.105. EldTunnelInterpretationOfMessageFailed

Error for an invalid session TSlaveThread ⚡*TSlaveThread* session message.

```
EldTunnelInterpretationOfMessageFailed = class(EldTunnelException)
```

**Description**

EldTunnelInterpretationOfMessageFailed is an EldTunnelException ⚡*EldTunnelException* descendant raised when an error occurs while interpreting a session message for TSlaveThread ⚡*TSlaveThread*.

### 3.1.106. EldTunnelMessageHandlingFailed

Error raised when an invalid data message is received in TSlaveThread ⚡*TSlaveThread*.

```
EldTunnelMessageHandlingFailed = class(EldTunnelException)
```

**Description**

EldTunnelMessageHandlingFailed is an EldTunnelException ⚡*EldTunnelException* descendant raised when TSlaveThread ⚡*TSlaveThread* receives a data message type that cannot be handled in a ClientOperation for the thread.

### 3.1.107. EldTunnelMessageTypeRecognitionError

Error for an unknown TSlaveThread ⚡*TSlaveThread* message type.

```
EldTunnelMessageTypeRecognitionError = class(EldTunnelException)
```

**Description**

EldTunnelMessageTypeRecognitionError is an EldTunnelException ⚡*EldTunnelException* descendant raised when an unknown message type is received in a TSlaveThread ⚡*TSlaveThread* message.

### 3.1.108. EldTunnelTransformError

Represents Transform errors on a Tunnel connection.

EldTunnelTransformError = **class**(EldTunnelException)

**Description**

EldTunnelTransformError is an EldTunnelException *↗EldTunnelException* descendant that represents errors occurring when a Tunnel performs data transformation and transmission, or the return message from a tunnel connection includes the tmError *↗tmError* message type.

### 3.1.109. EldTunnelTransformErrorBeforeSend

Represents Transform errors prior to transmission on a Tunnel connection.

EldTunnelTransformErrorBeforeSend = **class**(EldTunnelException)

**Description**

EldTunnelTransformErrorBeforeSend is an EldTunnelException *↗EldTunnelException* descendant that represents errors occurring prior to Tunnel data transformation and transmission.

### 3.1.110. EldUDPException

Ancestor class for Indy UDP exceptions.

EldUDPException = **class**(EldException)

**Description**

EldUDPException is an EldException *↗EldException* descendant that is the ancestor class for Indy UDP exceptions.

### 3.1.111. EldUDPReceiveErrorZeroBytes

Exception raised when no data was read from the connection.

EldUDPReceiveErrorZeroBytes = **class**(EldUDPException)

**Description**

EldUDPReceiveErrorZeroBytes is an EldUDPException *↗EldUDPException* exception descendant raised when TldUDPBase.ReceiveBuffer is called and no data is read from the connection.

### 3.1.112. EldUDPServerErrorException

Ancestor class for Indy UDP server exceptions.

EldUDPServerErrorException = **class**(EldUDPException)

**Description**

EldUDPServerErrorException is an EldUDPException *↗EldUDPException* descendant that is the ancestor class for Indy UDP server exceptions.

### 3.1.113. MClientThread

Performs communication with service threads.

MClientThread = **class**(TThread)

**Description**

MClientThread is a TThread descendant that encapsulates the TldTunnelMaster *↗TldTunnelMaster* connection from the client to the service threads for the tunnel master. MClientThread is used to store references to the TCP client connection to the tunnel server, and the master thread used by the tunnel server to handle client requests. MClientThread communicates with the tunnel server by reading messages from the outbound TCP client, and constructs headers that encapsulate the message for the tunnel server. When the TCP client is no longer readable, the MClientThread will terminate.

MClientThread notifies the tunnel server when a service thread is disconnected, and allows the tunnel server to maintain the number of connected services.

3.1.114. TAREcord

Represents an Address Resource Record.

```
TAREcord = class(TIdDNSResourceItem)
Description
TAREcord is a TIdDNSResourceItem ↗TIdDNSResourceItem descendant that represents an
Address Resource Record.
```

3.1.115. TClientData

Represents data about a client connection to TIdTunnelSlave ↗TIdTunnelSlave.

```
TClientData = class
Description
TClientData is a class used to represent data about a client connection tot he TIdTunnelSlave
↗TIdTunnelSlave. TClientData instances are stored in the TIdPeerThread.Data property when
the peer thread connects to the TIdTunnelSlave ↗TIdTunnelSlave. TClientData provides the
client context for operations in Connect ↗Connect and Execute on the TIdTunnelSlave
↗TIdTunnelSlave server.
```

3.1.116. THInfoRecord

Represents a Host resource record.

```
THInfoRecord = class(TIdDNSResourceItem)
Description
THInfo is a TIdDNSResourceItem ↗TIdDNSResourceItem descendant that represents a Host
resource record from a DNS query response.
```

3.1.117. TId3To4Coder

Ancestor for coders that perform conversion to/from binary and ASCII formats.

```
TId3To4Coder = class(TIdASCIICoder)
Description
TId3To4Coder is a TIdASCIICoder ↗TIdASCIICoder descendant that is also the ancestor for
coders which perform encoding or decoding that requires conversion from binary format to
ASCII format, or ASCII format to binary format. Descendants of IdCoder3To4 include
TIdBase64Decoder ↗TIdBase64Decoder, TIdBase64Encoder ↗TIdBase64Encoder,
TIdUUDecoder ↗TIdUUDecoder, TIdUUEncoder ↗TIdUUEncoder, TIdXXDecoder
↗TIdXXDecoder, and TIdXXEncoder ↗TIdXXEncoder, and represent implementations that
use specific coding tables and algorithms.
TId3To4Coder provides private method implementations that perform the encoding or
decoding operations, and include methods like Code3To4, Code4To3, CodeLine3To4,
CompleteLine3To4, and CodeLine4To3.
```

3.1.118. TIdAntiFreeze

Prevents an application UI from freezing.

```
TIdAntiFreeze = class(TIdAntiFreezeBase)
Description
Indy works on the blocking model. That is when calls are made to Indy the do not return until
they are complete. If calls are made in the main thread this will cause the Application User
Interface to "freeze" during Indy calls. TIdAntiFreeze counter acts this effect. TIdAntiFreeze
allows Indy subsystem to make Application.ProcessMessage calls so that Windows
messages continue to be executed while Indy calls are in process.
Only one TIdAntiFreeze can be active in an application. If another instance already exists, an
exception is raised. TIdAntiFreezeBase ↗TIdAntiFreezeBase uses the global variable
GAntiFreeze ↗GAntiFreeze, declared in the TIdAntiFreezeBase ↗TIdAntiFreezeBase unit, to
determine if another instance has already been created.
Note: GAntiFreeze ↗GAntiFreeze is not assigned if an instance is added in the form designer
(during design time).
Note: The TIdAntiFreeze.pas unit must NOT appear in the uses clause of any Indy or
```

descendant unit. This unit is linked in an application when the component is placed on a Form. This is done to preserve isolation from the FORMS.PAS and QForms.pas units.

3.1.119. TIdAntiFreezeBase

```
TIdAntiFreezeBase = class(TIdBaseComponent)
```

Description

Do not create instances of TIdAntiFreezeBase. Applications should use the TIdAntiFreeze *↗*TIdAntiFreeze component.

3.1.120. TIdASCIICoder

Ancestor for coders that use a table for encoding and decoding.

```
TIdASCIICoder = class(TIdCoder)
```

Description

TIdASCIICoder is a TIdCoder *↗*TIdCoder descendant that is the ancestor class for coders that use a table for encoding and decoding. **Note:** Do not create or use instance of TIdASCIICoder, use one the descendant classes like TId3To4Coder *↗*TId3To4Coder.

3.1.121. TIdAttachment

Encapsulates a MIME-encoded attachment or inline graphic.

```
TIdAttachment = class(TIdMessagePart)
```

Description

TIdAttachment is a TIdMessagePart *↗*TIdMessagePart descendant that encapsulates a MIME-encoded attachment or inline graphic. TIdAttachment and TIdText *↗*TIdText are used as collection items in a TIdMessageParts *↗*TIdMessageParts collection. TIdAttachment provides methods and properties relevant to Internet Message attachments including ContentDisposition, FileName, and SaveToFile. TIdAttachment also reintroduces

the Create constructor to specify the owner Collection and the file used to store the attachment.

3.1.122. TIdBase64Decoder

Implements a decoder for the Base64 MIME encoding scheme.

```
TIdBase64Decoder = class(TId3To4Coder)
```

Description

TIdBase64Decoder is a TIdCoder3To4 descendant that decodes Base64-encoded ASCII data into a it's binary format. TIdBase64Decoder utilizes the Base64 encoding scheme as described in the Internet Standards document:

- RFC 2045, Multipurpose Internet Mail Extensions (MIME), Part One: Format of Internet Message Bodies, Section 6.8 Base64 Content-Transfer-Encoding

The Base64 encoding mechanism is designed to represent arbitrary sequences of byte data in a form that need not be humanly readable. Base64 encoding is almost identical to the mechanism used in Privacy Enhanced Mail (PEM) applications, as defined in RFC 1421. Base64 encoding uses a 64-character subset of US-ASCII, enabling 6 bits to be represented per printable character. An extra character ("=") has special significance in Base64 encoding. Base64 encoding can represent all version of ISO 646, including US-ASCII, and all characters in the subset are also represented identically in EBCDIC. Base64 encoding represents 24-bit groups of input as output strings of 4 encoded characters. TIdBase64Decoder is designed to reverse the Base64 encoding operation to derive the original binary data. Any characters outside of the Base64 alphabet are to be ignored in Base64-encoded data. Base64 encoding is commonly used in E-Mail MIME messages and Usenet messages.

3.1.123. TIdBase64Encoder

Implements an encoder for the Base64 MIME encoding scheme.

```
TIdBase64Encoder = class(TId3To4Coder)
```

Description

TldBase64Encoder is a TldCoder3To4 descendant that encodes binary data into a 7-bit textual representation, called Base64, as described in the Internet Standards document:

- RFC 2045, Multipurpose Internet Mail Extensions (MIME), Part One: Format of Internet Message Bodies, Section 6.8 Base64 Content-Transfer-Encoding

The Base64 encoding mechanism is designed to represent arbitrary sequences of byte data in a form that need not be humanly readable. Base64 encoding is almost identical to the mechanism used in Privacy Enhanced Mail (PEM) applications, as defined in RFC 1421. Base64 encoding uses a 64-character subset of US-ASCII, enabling 6 bits to be represented per printable character. An extra character ("=") has special significance in Base64 encoding. Base64 encoding can represent all version of ISO 646, including US-ASCII, and all characters in the subset are also represented identically in EBCDIC.

Base64 encoding represents 24-bit groups of input as output strings of 4 encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating 3 8bit input groups. These 24 bits are then treated as 4 concatenated 6-bit groups, each of which is translated into a single digit in the base64 alphabet. When encoding a bit stream using Base64 encoding, the stream must be presumed to be ordered with the most-significant-bit first. That is, the first bit in the stream will be the high-order bit in the first 8bit byte, and the eighth bit will be the low-order bit in the first 8bit byte, etc.

Each 6-bit group is used as an index into an array of 64 printable characters. The character referenced by the index is placed in the output string. These characters are selected to be universally representable, and the set excludes characters with particular significance to SMTP (".", CR ~~CR~~, LF) and to the multipart boundary delimiters defined in RFC 2046 ("-"). The following table identifies characters and encodings for the Base64 Alphabet:

Value	Encoding	Value	Encoding
0	A	34	i
1	B	35	j
2	C	36	k
3	D	37	l
4	E	38	m
5	F	39	n
6	G	40	o
7	H	41	p
8	I	42	q
9	J	43	r
10	K	44	s
11	L	45	t

12	M	46	u
13	N	47	v
14	O	48	w
15	P	49	x
16	Q	50	y
17	R	51	z
18	S	52	0
19	T	53	1
20	U	54	2
21	V	55	3
22	W	56	4
23	X	57	5
24	Y	58	6
25	Z	59	7
26	a	60	8
27	b	61	9
28	c	62	+
29	d	63	/
30	e	(pad)	=
31	f		
32	g		
33	h		

Base64-encoded output must be represented in lines of no more than 76 characters each. All line breaks or other characters not found in the Base64 alphabet must be ignored by decoding software.

Special processing is performed if fewer than 24 bits are available at the end of the data being encoded. A full encoding quantum is always completed at the end of a body. When fewer than 24 input bits are available in an input group, zero bits are added (on the right) to form an integral number of 6-bit groups. Padding at the end of the data is performed using the "=" character. Since all Base64 input is an integral number of octets, only the following cases can arise:

- The final quantum of encoding input is an integral multiple of 24 bits; here, the final unit of encoded output will be an integral multiple of 4 characters with no "=" padding.
- The final quantum of encoding input is exactly 8 bits; here, the final unit of encoded output will be two characters followed by two "=" padding characters.
- The final quantum of encoding input is exactly 16 bits; here, the final unit of encoded output will be three characters followed by one "=" padding character.

Because it is used only for padding at the end of the data, the occurrence of any "=" characters may be taken as evidence that the end of the data has been reached (without truncation in transit). No such assurance is possible, however, when the number of octets transmitted was a multiple of three and no "=" characters are present.

Any characters outside of the Base64 alphabet are to be ignored in Base64-encoded data. Base64 encoding is commonly used in E-Mail MIME messages and Usenet messages.

### 3.1.124. TIdBaseComponent

Ancestor class for all Indy components.

```
TIdBaseComponent = class(TComponent)
```

**Description**

TIdBaseComponent is the base class that is the ancestor for all Indy components.

### 3.1.125. TIdBuffer

Buffer used in Indy read and write operations.

```
TIdBuffer = class(TMemoryStream)
```

**Description**

TIdBuffer is a **TMemoryStream** descendant used in read and write buffering operations for Indy components. TIdBuffer includes all the inherited stream functionality of **TMemoryStream**, and extends the class with the RemoveXBytes method to manage dynamic memory buffer usage in Indy.

TIdBuffer is used as a generic buffer in operations that read data from the TCP/IP protocol stack, and in communications classes that perform write buffering like TIdTCPConnection *↗TIdTCPConnection*.

### 3.1.126. TIdCardAddressItem

Encapsulates a complete VCard address.

```
TIdCardAddressItem = class(TCollectionItem)
```

**Description**

The TIdCardAddressItem object encapsulates a VCard address in the TIdVCard *↗TIdVCard* component.

### 3.1.127. TIdCardPhoneNumber

Encapsulates a VCard telephone number.

```
TIdCardPhoneNumber = class(TCollectionItem)
```

**Description**

TIdCardPhoneNumber encapsulates a VCard telephone number in the TIdVCard *↗TIdVCard* component.

### 3.1.128. TIdChargenServer

Defines a Character Generator server.

```
TIdChargenServer = class(TIdTCPServer)
```

**Description**

TIdChargenServer defines a Chargen server as defined in the Internet Standards document: RFC 864. Chargen or Character Generator Protocol is used for stress-testing a connection.

### 3.1.129. TIdCoder

```
TIdCoder = class(TIdBaseComponent)
```

**Description**

TIdCoder is a base class that identifies common properties and method needed to implement encryption, decryption, encoding, and decoding of string variables or file content. TIdCoder is the ancestor of the various coders and decoders included with Indy.

**Note:** Do not create instances of TIdCoder. Use one of the descendant classes that implements specific Coder functionality, including:

- TIdBase64Encoder *↗TIdBase64Encoder*
- TIdBase64Decoder *↗TIdBase64Decoder*
- TIdUUEncoder *↗TIdUUEncoder*
- TIdUUDecoder *↗TIdUUDecoder*
- TIdIMFDecoder *↗TIdIMFDecoder*
- TIdCoderMD2 *↗TIdCoderMD2*
- TIdCoderMD4 *↗TIdCoderMD4*
- TIdCoderMD5 *↗TIdCoderMD5*
- TIdQuotedPrintableEncoder *↗TIdQuotedPrintableEncoder*

3.1.130. TIdCoderCollection

Collection of registered coder classes.

TIdCoderCollection = **class**(TCollection)

**Description**

TIdCoderCollection is a collection that stores registration information about a coder which was registered in your program using the RegisterCoderClass *↗RegisterCoderClass* procedure. This collection stores items as TIdCoderItem *↗TIdCoderItem* objects. You do not create this class or free it at all but use the CoderCollective *↗CoderCollective* global variable.

3.1.131. TIdCoderCRC16

Implements a CRC calculation encoder for Tunnel communication classes.

TIdCoderCRC16 = **class**(TIdCoder)

**Description**

TIdCoderCRC16 is a TIdCoder *↗TIdCoder* descendant that provides CRC calculation for encoded message values. TIdCoderCRC16 is used by Tunnel classes like TReceiver *↗TReceiver* and TSender *↗TSender* to construct CRC values for encapsulated tunnel message headers.

3.1.132. TIdCoderItem

Represent a registered coder.

TIdCoderItem = **class**(TCollectionItem)

**Description**

TIdCoderItem is a collection item that stores registration information about a coder which was registered in your program using the RegisterCoderClass *↗RegisterCoderClass* procedure.

3.1.133. TIdCoderMD2

TIdCoderMD2 = **class**(TIdCoder)

**Description**

TIdCoderMD2 is a TIdCoder *↗TIdCoder* descendant that implements the RSA-MD2 encryption algorithm as described in the Internet Standards documents:

- RFC 1115, Privacy Enhancement for Internet Electronic Mail: Part III - Algorithms, Modes, and Identifiers
- RFC 1319, The MD2 Message-Digest Algorithm

TIdCoderMD2 can be used in various cryptography implementations that require the RSA-MD2 algorithm, such as:

- RFC 1423, Privacy Enhancement for Internet Electronic Mail, Part III: Algorithms, Modes, and Identifiers
- RFC 1507, Distributed Authentication Security Service (DASS)
- RFC 1750, Randomness Recommendations for Security
- RFC 2313, PKCS #1: RSA Encryption Version 1.5
- RFC 2315, PKCS #7: Cryptographic Message Syntax Version 1.5



- RFC 2440, OpenPGP Message Format
- RFC 2459, Internet X.509 Public Key Infrastructure Certificate and CRL Profile
- RFC 2660, The Secure HyperText Transfer Protocol

### 3.1.134. TIdCoderMD4

Implements the MD4 encryption algorithm.

```
TIdCoderMD4 = class(TIdCoder)
```

**Description**

TIdCoderMD4 is a TIdCoder *≠* TIdCoder descendant that implements the RSA-MD4 encryption algorithm as described in the Internet Standards document:

- RFC 1186, MD4 Message Digest Algorithm

TIdCoderMD4 can be used in various cryptography implementations that require the RSA-MD4 algorithm. The RSA-MD4 algorithm takes an input message of an arbitrary length and produces a 64-byte "fingerprint" or "message digest" of the input.

The MD4 algorithm is used for digital signature applications, where a large file must be "compressed" in a secure manner before being signed with the RSA public-key cryptosystem. The MD4 algorithm has been placed in the public domain.

### 3.1.135. TIdCoderMD5

Implements the MD5 encryption algorithm.

```
TIdCoderMD5 = class(TIdCoderMD4)
```

**Description**

TIdCoderMD5 is a TIdCoderMD4 *≠* TIdCoderMD4 descendant that implements the RSA-MD5 encryption algorithm as described in the Internet Standards document:

- RFC 1321, The MD5 Message-Digest Algorithm

TIdCoderMD5 implements the MD5 algorithm used for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA. The RSA-MD5 algorithm takes an input message of an arbitrary length and produces a 16-byte "fingerprint" or "message digest" of the input.

The MD5 algorithm is an extension of the MD4 message-digest algorithm. MD5 is slightly slower than MD4, but is more "conservative" in design. MD5 exists to compensate for potential security holes that exist in MD4.

TIdCoderMD5 differs from TIdCoderMD4 *≠* TIdCoderMD4 in the following ways:

- A fourth round operation is added to the algorithm.
- Each step now has a unique additive constant.
- Rounding in second step of the algorithm has been made less symmetrical than it's MD4 predecessor.
- Each step accumulates the result of the previous step promoting an "avalanche effect".
- The input word order in rounds 2 and 3 is changed to make these patterns less predictable and repetitive.
- The shift amounts in each round have been approximately optimized, to yield a faster "avalanche effect." The shifts in each of the rounds is distinct.

TIdCoderMD5 can be used in various cryptography implementations that require the RSA-MD5 algorithm. Some MD5 implementations are described in the following documents:

- RFC 1828, IP Authentication using Keyed MD5
- RFC 1864, The Content-MD5 Header Field
- RFC 2082, RIP-2 MD5 Authentication
- RFC 2085, HMAC-MD5 IP Authentication with Replay Prevention
- RFC 2385, Protection of BGP Sessions via the TCP MD5 Signature Option
- RFC 2537, RSA/MD5 KEYS and SIGs in the Domain Name System (DNS)

The MD5 algorithm has been placed in the public domain.

### 3.1.136. TIdComponent

Ancestor of all Indy client and server components.

`TIdComponent = class(TIdBaseComponent)`

**Description**

TIdComponent is the ancestor of all Indy components which implement client or server functionality.

### 3.1.137. TIdConnectionIntercept

Ancestor for Indy Secure Socket *↗*Socket Layer connection intercept classes.

`TIdConnectionIntercept = class(TIdBaseComponent)`

**Description**

The TIdSSLConnectionIntercept *↗TIdSSLConnectionIntercept* class is ancestor for Secure Sockets Layer connection intercept classes such as TIdConnectionInterceptOpenSSL *↗TIdConnectionInterceptOpenSSL*. You do not create instances of this object.

Indy does not give you any legal rights to use SSL. Some nations including the U.S. regulate or prohibit the export of strong-encryption such as SSL. In addition, using SSL in some nations may be illegal. It is up to you to determine the legal situation in your nation.

### 3.1.138. TIdConnectionInterceptOpenSSL

Implements Secure Socket *↗*Socket Layer support for Indy Connection Intercept components.

`TIdConnectionInterceptOpenSSL = class(TIdSSLConnectionIntercept)`

**Description**

The TIdConnectionInterceptOpenSSL intercept class implements the Open SSL implementation of Secure Sockets Layer through support .DLL's available at <http://www.intelicom.si/>. You have to deploy these support .DLL's in order to use this

component's functionality in your application. To do this, we recommend that you install the .DLL's in the user's system directory.

Indy does not give you any legal rights to use SSL. Some nations including the U.S. regulate or prohibit the export of strong-encryption such as SSL. In addition, using SSL in some nations may be illegal. It is up to you to determine the legal situation in your nation.

### 3.1.139. TIdCookie

Implements the HTTP session state mechanism, or "Cookie".

`TIdCookie = class(TCollectionItem)`

**Description**

TIdCookie implements the session state management mechanism for HTTP requests and responses, or "Cookie", as described in the Internet Standards document:

- HTTP State Management Mechanism, RFC 2109

"Cookies" are used to carry state information between participating HTTP servers and user agents (clients). The information stored on the user agent is returned back to the web server in subsequent HTTP requests.

TIdCookie is a TCollectionItem descendant and can be stored in a TIdCookieCollection *↗TIdCookieCollection*. TIdHTTPRequestInfo *↗TIdHTTPRequestInfo* and TIdHTTPResponseInfo *↗TIdHTTPResponseInfo* are cookie collections used to perform the exchange of session state "Cookies" for a corresponding HTTP request or response.

### 3.1.140. TIdCookieCollection

TIdCookieCollection is a container for TIdCookie *↗TIdCookie* objects.

`TIdCookieCollection = class(TCollection)`

**Description**

TIdCookieCollection is a container for TIdCookie *↗TIdCookie* objects. TIdCookieCollection provides the indexed Items property to access a TIdCookie *↗TIdCookie* object by it's numeric

position in the collection. TldCookieCollection provides the indexed Cookie property to access a TldCookie *↗* *TldCookie* object in the collection by Name. Use GetCookieIndex to locate the index position of a cookie with a specified Name. Use AddSrcCookie to create a new TldCookie *↗* *TldCookie* in the collection using the source code form.

### 3.1.141. TldDateTimeStamp

Provides Date and Time handling for Internet Protocols. Description:

TldDateTimeStamp is a TldBaseComponent *↗* *TldBaseComponent* descendant that processes date and time values using the various formats required by some Internet Protocols. TldDateTimeStamp methods performs operations based on units of Milliseconds, Seconds, Days, and Years with smaller time units affecting larger time units where necessary. All other common time units are converted to these base units before processing.

```
TldDateTimeStamp = class(TldBaseComponent)
```

### 3.1.142. TldDayTime

Implements a DayTime protocol client.

```
TldDayTime = class(TIdTCPClient)
```

**Description**

TldDayTime implements the DayTime protocol (RFC 867) as a client. This protocol is simply where a DayTime server sends the current day and time in a human readable format and is sometimes used for debugging. If you need a time-synchronization solution, we recommend that you use the Time protocol encapsulated by TldTime *↗* *TldTime* and TldTimeServer *↗* *TldTimeServer*.

### 3.1.143. TldDayTimeServer

Implements a DayTime protocol server.

```
TldDayTimeServer = class(TIdTCPServer)
```

**Description**

TldDayTimeServer implements the DayTime protocol (RFC 867) as a server. This protocol is simply where a DayTime server sends the current day and time in a human readable format and is sometimes used for debugging. If you need a time-synchronization solution, we recommend that you use the Time protocol encapsulated by TldTime *↗* *TldTime* and TldTimeServer *↗* *TldTimeServer*.

### 3.1.144. TldDICTServer

Implements a Dictionary Protocol server for dictionary databases.

```
TldDICTServer = class(TIdTCPServer)
```

**Description**

TldDICTServer is a server implementation of the Dictionary Server Protocol as described in the Internet Standards document A Dictionary Server Protocol, RFC 2229, by R. Faith, October 1997. The Dictionary Server Protocol (DICT) is a TCP transaction based query/response protocol that allows a client to access dictionary definitions from a set of natural language dictionary databases. TldDICTServer uses an event-based architecture that provides event handlers for the following DICT protocol commands:

- AUTH - OnCommandAuth
- CLIENT - OnCommandClient
- DEFINE - OnCommandDefine
- HELP - OnCommandHelp
- MATCH - OnCommandMatch
- OPTION - OnCommandOption
- QUIT - OnCommandQuit
- SASLAUTH - OnCommandSASLAuth
- SHOW - OnCommandShow

- STATUS - OnCommandStatus

TIdDICTServer also provides the OnCommandOther event handler to respond to unknown DICT commands, or DICT protocol extensions.  
TIdDICTServer does not provide an implementation of the event handlers. The application should assign procedures to the event handlers to respond to event notifications, or the result code 500 with the error message RSCMDNotRecognized *↗RSCMDNotRecognized* will be generated.

### 3.1.145. TIdDISCARDServer

Implements a Discard Protocol server.

```
TIdDISCARDServer = class(TIdTCPServer)
```

**Description**

TIdDISCARDServer implements the Discard Protocol (RFC 863) as a server. This is a useful debugging and measurement protocol where the data sent to the server is simply discarded.

### 3.1.146. TIdDNSHeader

Represents the DNS Header.

```
TIdDNSHeader = class(TObject)
```

**Description**

TIdDNSHeader is a class that provides access to the DNS header for a DNS query or response. TIdDNSResolver *↗TIdDNSResolver* uses a TIdDNSHeader instance to prepare a DNS query for transmission to the DNS server, and capture return values from the DNS response. TIdDNSHeader exposes all fields in the DNS header to allow construction of a DNS server.

### 3.1.147. TIdDNSQuestionList

Contains header, question, and response objects for a DNS Message.

```
TIdDNSQuestionList = class(TCollection)
```

**Description**

TIdDNSQuestionList is a TCollection descendant used to hold Header, question, and response objects that make up the DNS Message object. TIdDNSQuestionList is used to hold both the sent and received queries in easy to handle pascal data structure.

A Question Consists of a Name, aType and aClass. Name can be a domain name or an IP address depending on the type of question.

### 3.1.148. TIdDNSResolver

Implements a UDP-based resolver for DNS protocol queries.

```
TIdDNSResolver = class(TIdUDPClient)
```

**Description**

TIdDNSResolver is a TIdUDPClient *↗TIdUDPClient* descendant that provides a stateless UDP-based implementation of a resolver for DNS (Domain Name Server) queries using the DNS protocol. DNS is described in the Internet Standards documents:

- RFC 1034 Domain Names - Concepts and Facilities
- RFC 1035 Domain Names - Implementation and Specification
- RFC 1591 Domain Name System Structure and Delegation
- RFC 1183 New DNS RR Definitions.
- RFC 2181 Clarifications to the DNS Specification.

TIdDNSResolver provides facilities for accessing the Domain Name Space and Resource Records retrieved from a DNS Server as a result of queries for specified domain names, resource types, and resource classes.

### 3.1.149. TIdDNSResourceItem

Contains data returned in a DNS server response.

```
TIdDNSResourceItem = class(TCollectionItem)
```

**Description**

TIdDNSResourceItem is a TCollectionItem descendant that represents The Data returned by a DNS server in a response to a DNS query. TIdDNSResourceItem the ancestor class for class instances that represent Resource Records from the DNS response, such as:

- TMXRecord ↗ *TMXRecord*
- TAREcord ↗ *TAREcord*
- TNameRecord ↗ *TNameRecord*
- TPTRRecord ↗ *TPTRRecord*
- THInfoRecord ↗ *THInfoRecord*
- TMInfoRecord ↗ *TMInfoRecord*
- TMRecord ↗ *TMRecord*
- TSOARRecord ↗ *TSOARRecord*
- TWKSRecord ↗ *TWKSRecord*

### 3.1.150. TIdDNSResourceList

Container collection for DNS responses from a DNS query.

```
TIdDNSResourceList = class(TCollection)
```

**Description**

TIdDNSResourceList is a TCollection descendant that acts as a container for TIdDNSResourceItem ↗ *TIdDNSResourceItem* instances returned from a DNS Query.

### 3.1.151. TIdEcho

Implements an Echo client.

```
TIdEcho = class(TIdTCPClient)
```

**Description**

TIdEcho implements the TCP-based variant of the Echo Protocol as described in the Internet Standards document EchoProtocol, RFC 862, by Jon Postel.

TIdEcho is a TIdTCPClient ↗ *TIdTCPClient* descendant that is a useful debugging and measurement protocol where the data sent to the server is simply sent back to the originating client. TIdEcho is provided as a measure of the time it takes to send and receive data from an Echo Server (TIdECHOSever).

The TCP-based implementation of the Echo Protocol is a connection based application. An Echo Server listens for TCP connections on TCP Port 7. When a connection is established, any received data is returned to the originating client. The connection continues until the client application closes the connection.

To use TIdEcho, perform the following actions in your application:

- Call Connect ↗ *Connect* to connect to the Echo Server.
- Call Echo to send data to the server and receive the server response.
- Use EchoTime to retrieve the number of milliseconds needed to send and receive data from the server.
- Call Disconnect to close the connection to the server.

### 3.1.152. TIdECHOSever

Implements an Echo Protocol server.

```
TIdECHOSever = class(TIdTCPServer)
```

**Description**

TIdECHOSever implements the Echo Protocol (RFC 862) as a server. This is a useful debugging and measurement protocol where the data sent to the server is simply sent back to the client.

### 3.1.153. TIdEmailAddressItem

Represents the name an email address in RFC 822 form.

```
TidEmailAddressItem = class(TCollectionItem)
```

**Description**

The TidEmailAddressItem object structures a person's name and E-Mail address into a RFC 822-compliant form and can parse a RFC 822 compliant E-Mail address into the person's name and E-Mail address. In addition, encoding and decoding required for some languages is also built in.

3.1.154. TidEmailAddressList

Represents a collection of email addresses.

```
TidEmailAddressList = class(TOwnedCollection)
```

**Description**

TidEmailAddressList is a collection that contains a list of E-Mail addresses, TidEmailAddressItem *↗* *TidEmailAddressItem* objects, which can be imported and exported in comma delimited form which is used in many E-Mail messages.

3.1.155. TidFinger

Implements a Finger protocol client.

```
TidFinger = class(TidTCPClient)
```

**Description**

The TidFinger implements the Finger User Information Protocol or Finger (RFC 1288) as a client. The finger protocol is an interface for a database of information about users on a remote system and could include information such as if they were logged in to the system, when they last read their E-Mail or sometimes information the user wishes to make available publicly (.plan file).

3.1.156. TidFingerServer

Specifies a Finger protocol server.

```
TidFingerServer = class(TidTCPServer)
```

**Description**

The TidFingerServer helps to implement the Finger User Information Protocol or Finger (RFC 1288) as a server. The finger protocol is an interface for a database of information about users on a remote system and could include information such as if they were logged in to the system, when they last read their E-Mail or sometimes information the user wishes to make available publicly (.plan file).

3.1.157. TidFTP

Implements a File Transfer Protocol (FTP) client.

```
TidFTP = class(TidTCPClient)
```

**Description**

TidFtp implements a File Transfer Protocol (FTP) client as described in the Internet Standards document:

- File Transfer Protocol (FTP), RFC 959, by J. Postel and J. Reynolds, October 1985.

FTP is a basic file-sharing system for uploading and downloading files, as well as managing files and directories. The FTP protocol is commonly used for software distribution, uploading files to a web-server, and providing archives for various purposes.

3.1.158. TidGopher

Implements a Gopher client.

```
TidGopher = class(TidTCPClient)
```

**Description**

The TIdGopher is a client implementation of the Internet Gopher protocol (RFC 1436) as described in the Internet Standards document:

- The Internet Gopher Protocol, RFC 1436, by F. Anklesaria, M. McCahill, P. Lindner, D. Johnson, D. Torrey and B. Alberti, March 1993

TIdGopher also implements partial support for the Gopher+ protocol as described in the document:

- Gopher+, Upward-Compatible Enhancements to the Internet Gopher protocol, by Farhad Anklesaria, Paul Lindner, Mark P. McCahill, Daniel Torrey, David Johnson, and Bob Alberti, July 1993 )

Gopher is a distributed document system developed at the University of Minnesota as an attempt to make locating information resources intuitive for users through a series of structured menus leading to documents or pointers to other protocols such as Telnet and TN3270. While this has been succeeded by the World Wide Web, this is still useful for accessing legacy systems and has a low overhead due to its simplicity and structure.

**3.1.159. TIdGopherMenu**

Collection of Gopher menu items.

TIdGopherMenu = **class**(TCollection)

**Description**

TIdGopherMenu is a collection of TIdGopherMenuitem *↗TIdGopherMenuitem* instances, and encapsulates a Gopher menu or a Gopher + Extended Menu. TIdGopherMenu is returned by the following:

- TIdGopher.GetExtendedMenu
- TIdGopher.GetMenu
- TIdGopher.Search

**3.1.160. TIdGopherMenuitem**

Implements a Gopher menu item.

TIdGopherMenuItem = **class**(TCollectionItem)

**Description**

TIdGopherMenuitem implements an Internet Gopher menu item as described in the Internet Standards document:

- The Internet Gopher Protocol, RFC 1436, by F. Anklesaria, M. McCahill, P. Lindner, D. Johnson, D. Torrey and B. Alberti, March 1993

TIdGopherMenuitem also implements support for the Gopher+ item block as described in the document:

- Gopher+, Upward-Compatible Enhancements to the Internet Gopher protocol, by Farhad Anklesaria, Paul Lindner, Mark P. McCahill, Daniel Torrey, David Johnson, and Bob Alberti, July 1993 )

**3.1.161. TIdGopherServer**

Implements a Gopher server.

TIdGopherServer = **class**(TIdTCPServer)

**Description**

The TIdGopherServer is a server implementation of the Internet Gopher protocol (RFC 1436) as described in the Internet Standards document:

- The Internet Gopher Protocol, RFC 1436, by F. Anklesaria, M. McCahill, P. Lindner, D. Johnson, D. Torrey and B. Alberti, March 1993

TIdGopherServer also implements partial support for the Gopher+ protocol as described in the document:

- Gopher+, Upward-Compatible Enhancements to the Internet Gopher protocol, by

Farhad Anklesaria, Paul Lindner, Mark P. McCahill, Daniel Torrey, David Johnson, and Bob Alberti, July 1993 )

Gopher is a distributed document system developed at the University of Minnesota as an attempt to make locating information resources intuitive for users through a series of structured menus leading to documents or pointers to other protocols such as Telnet and TN3270. While this has been succeeded by the World Wide Web, this is still useful for accessing legacy systems and has a low overhead due to its simplicity and structure.

3.1.162. TIdHeaderInfo

Encapsulates HTTP header information for HTTP requests and responses.

TIdHeaderInfo = **class**(TPersistent)

Description

TIdHeaderInfo is a TPersistent descendant used to encapsulate HTTP header information for HTTP requests and responses.

3.1.163. TIdHeaderList

Implements a collection of header names and values.

TIdHeaderList = **class**(TStringList)

Description

TIdHeaderList is a TStringList descendant that implements a collection of header names and values, and facilitates the processing of headers used in many protocols such as Gopher+, HTTP, NNTP, POP3, and SMTP. Features include the use of a ":" instead of a comma for separating field names from values (this can be changed with the NameValueSeparator property), and optional header folding and unfolding.

3.1.164. TIdHostNameServer

Implements a HostName server.

TIdHostNameServer = **class**(TIdTCPServer)

Description

TIdHostNameServer is a TIdTCPServer *↗TIdTCPServer* descendant that provides a server implementation of the Hostname Server as described in the Internet Standards document:

- HOSTNAME Server, RFC 953, by K. Harrenstien, M. Stahl, and E. Feinler, October 1985

HostName Server provides machine-readable name and/or address information describing networks, gateways, hosts, and eventually domains, within the internet environment.

3.1.165. TIdHTTP

HTTP client implementation.

TIdHTTP = **class**(TIdTCPClient)

Description

TIdHTTP is a TIdTCPClient *↗TIdTCPClient* descendant that is a client implementation of the Hypertext Transfer Protocol (HTTP) as described in the Internet Standards documents:

- Hypertext Transfer Protocol version 1.0 (HTTP/1.0), RFC 1945
- Hypertext Transfer Protocol version 1.1 (HTTP/1.1), RFC 2616

TIdHTTP supports HTTP 1.0 and HTTP 1.1 protocols for use as a web-browser or web-robot. TIdHTTP also supports the Secure Hypertext Transport protocol (HTTPS) as described in the Internet Standards document:

- The Secure HyperText Transfer Protocol, RFC 2660

Assign an instance of TIdConnectionInterceptOpenSSL *↗TIdConnectionInterceptOpenSSL* to



the Intercept property to allow use of the HTTPS protocol. For the Windows platform, you must install the Indy OpenSSL support .DLL's available at <http://www.intelicom.si> to enable Secure Socket ~~SSL~~ *Socket* Layer support.

### 3.1.166. TIdHTTPRequestInfo

Implements a HTTP Request object.

```
TIdHTTPRequestInfo = class(TObject)
```

**Description**

TIdHTTPRequestInfo is a TObject descendant that encapsulates access to information for a HTTP request.

TIdHTTPRequestInfo publishes properties that provide access to various information for a HTTP request. The properties include the HTTP session, authentication parameters, the remote computer addresses, HTTP headers, cookies, the HTTP command and version, and the document URL for the request.

TIdHTTPRequestInfo is used by TIdHTTPServer ~~SSL~~ *TIdHTTPServer* to capture HTTP headers, and to store a reference to the persistent HTTP session in the session list.

### 3.1.167. TIdHTTPResponseInfo

Implements a HTTP response object.

```
TIdHTTPResponseInfo = class(TObject)
```

**Description**

TIdHTTPResponseInfo is a TObject descendant that encapsulates access to information for a HTTP response.

TIdHTTPResponseInfo publishes properties that provide access to various information for a HTTP response.

These properties include the HTTP session, the authentication realm, Cookies, Headers, and the response content, length, and type.

TIdHTTPResponseInfo is used by TIdHTTPServer ~~SSL~~ *TIdHTTPServer* to prepare a HTTP response for a peer thread request.

### 3.1.168. TIdHTTPServer

HTTP Server Implementation.

```
TIdHTTPServer = class(TIdTCPServer)
```

**Description**

TIdHTTPServer is a TIdTCPServer ~~SSL~~ *TIdTCPServer* descendant that is a server implementation of the Hypertext Transfer Protocol (HTTP) as described in the Internet Standards document:

- Hypertext Transfer Protocol version 1.0 (HTTP), RFC 1945

TIdHTTPServer supports the HTTP 1.0 protocol for use as a web server.

TIdHTTPServer also supports the Secure Hypertext Transport protocol (HTTPS) as described in the Internet Standards document:

- The Secure HyperText Transfer Protocol, RFC 2660

Assign an instance of TIdServerInterceptOpenSSL ~~SSL~~ *TIdServerInterceptOpenSSL* to the Intercept property to allow use of the HTTPS protocol. For the Windows platform, you must install the Indy OpenSSL support .DLL's available at <http://www.intelicom.si> to enable Secure Socket ~~SSL~~ *Socket* Layer support.

### 3.1.169. TIdHTTPSession

Implements a persistent HTTP Session.

```
TIdHTTPSession = class(TObject)
```

**Description**

TIdHTTPSession is a TObject descendant that implements a HTTP Session.

TIdHTTPSession also provides support for maintaining a persistent HTTP session and state for TIdHTTPServer ~~SSL~~ *TIdHTTPServer*. TIdHTTPSession is used with TIdHTTPSessionList

↗*TidHTTPSessionList* to provide persistent HTTP session management using the SessionID property and the "IDSESSIONID" Cookie.  
SessionID is the unique identifier for the HTTP session.  
LastTimestamp is the date and time that the HTTP session was last modified.  
RemoteHost is the address of the remote computer using the HTTP session.  
Content is a storage area for string values that can be used to form HTTP request and/or response messages.

3.1.170. TidHTTPSessionList

Implements a container for HTTP sessions.

```
TidHTTPSessionList = class(TObject)
```

Description

TidHTTPSessionList is a TObject instance that implements a container for HTTP session objects.  
TidHTTPSessionList also provides support for maintaining persistent HTTP sessions and state for TidHTTPServer ↗*TidHTTPServer*.  
Use Clear to signal HTTP session completion and remove all sessions from the session list.  
Use CreateSession to initialize a new persistent HTTP session and add it to the session list.  
Use GetSession to retrieve a persistent HTTP session or add a new session to the session list.  
Use SessionTimeout to specify how long persistent sessions will remain in the session list.  
Use PurgeStaleSessions to incrementally allow HTTP sessions where the SessionTimeout period has expired to be removed from the session list.  
Use the OnSessionStart and OnSessionEnd event handlers to signal the application when a persistent HTTP session is added or removed from the session list.

3.1.171. TidIcmpClient

```
TidIcmpClient = class(TIdRawClient)
```

Description

TidIcmpClient is a Internet Protocol client implementation based on the Internet standards

document, RFC 792 - INTERNET CONTROL MESSAGE PROTOCOL , by Jon Postel.  
TidIcmpClient is used to send an Internet Control Message Protocol (ICMP) packet to another computer. ICMP packets are used for ping and TraceRoute capability to help diagnose problems with connectivity.  
TidIcmpClient is a descendant of the abstract class TIdRawClient ↗*TidRawClient*, and uses the Protocol value **Id\_IPPROTO\_ICMP** ↗*Id\_IPPROTO\_ICMP* (Decimal 1). TidIcmpClient uses the message formats and algorithms as defined in RFC 792.  
Use TidIcmpClient to diagnose a communications links by using Ping. Host identifies the IP address or computer name for the echo request. Use ReplyStatus and OnReply to access information received as a result of the echo request datagram. Set ReceiveTimeout to control the time to wait for a response to the echo request datagram.  
Use the TTL property to set the packet's Time To Live value: the maximum number of hops the packet will travel before being bounced.  
For Traceroute application, send ping echo requests with increased TTL values. Each reply will hold the IP address of the hop where the TTL expired, allowing the program to build the packet's route. .

3.1.172. TidIMAP4Server

Implements an IMAP4 server.

```
TidIMAP4Server = class(TIdTCPServer)
```

Description

TidIMAP4Server helps developers implement a Internet Message Access Protocol, Version 4rev1 or IMAP4 server (RFC 2060).  
The IMAP4 protocol permits manipulation of remote message folders, called "mailboxes", in a way that is functionally equivalent to local mailboxes.

3.1.173. TidIMFDecoder

Implements a decoder for Internet Message Format-compliant messages.

```
TidIMFDecoder = class(TIdCoder)
```

**Description**

TidIMFDecoder is a TidCoder *↗TidCoder* descendant that implements a decoder for Internet Message Format-compliant messages, as described in the Internet Standards documents:

- Standard for the Format of ARPA Internet Text Messages, RFC 822, Revised by David H. Crocker, August 1982
- Standard for Interchange of USENET Messages, RFC 1036, by M. Horton, December 1987

TidIMFDecoder extends TidCoder *↗TidCoder* to include the capability to detect RFC-822 headers for Content-Type, Content-Transfer-Encoding, and MIME boundaries. TidIMFDecoder is used to decode raw message content when it is received using TidMessageClient *↗TidMessageClient*.

3.1.174. TidIMFUUDecoder

TidIMFUUDecoder is not currently implemented...

```
TidIMFUUDecoder = class(TidIMFDecoder)
```

**Description**

TidIMFUUDecoder is not currently implemented...

3.1.175. TidIPWatch

Determines the online status and IP addresses for a computer.

```
TidIPWatch = class(TidComponent)
```

**Description**

TidIPWatch determines Online status, returns current IP address, and (optionally) keeps history of IP addresses issued to the computer using the class instance. TidIPWatch uses CurrentIP to determine online status, and will not establish a new connection using RAS or DUN.

3.1.176. TidIPWatchThread

Provides a Timer-like thread for TidIPWatch *↗TidIPWatch*.

```
TidIPWatchThread = class(TidThread)
```

**Description**

TidIPWatchThread is a TidThread *↗TidThread* descendant that provides the ability to monitor the local PC for new IP addresses at a defined interval. By default, TidIPWatchThread awakens every Interval milliseconds and calls the TimerEvent until Terminated.

3.1.177. TidIRCServer

Specifies an Internet Relay Chat Protocol server.

```
TidIRCServer = class(TidTCPServer)
```

**Description**

TidIRCServer is a TidTCPServer *↗TidTCPServer* descendant that provides a server implementation of the Internet Relay Chat (IRC) Protocol as described in the Internet Standards document:

- Internet Relay Chat Protocol (IRC), RFC 1459, by J. Oikarinen and D. Reed, May 1993

The IRC protocol provides users with a way to chat among themselves in "chat rooms". The IRC protocol is a text-based conferencing protocol that uses TCP/IP as the network transport. A typical IRC environment involves a single process (the server) that forms a central point for client (or other server) connections, performing the required message delivery/multiplexing and other functions. IRC allows various forms of message delivery including user-to-user, user-to-list, user-to-group, user-to-server, and server-to-server communication modes. The only network configuration allowed for IRC servers is that of a spanning tree where each server acts as a central node for the rest of the network visible to the server. An IRC client can be any socket-enable application capable of connecting to an IRC server that is not already an IRC server. TidIRCServer uses the standard port number IdPORT\_IRC *↗IdPORT\_IRC* reserved for the IRC protocol to listen for client connections. TidIRCServer, like TidTCPServer

✂*TIdTCP*Server, is a multithreaded application where new connections are respresented by a TIdPeerThread ✂*TIdPeerThread* that is serviced by the TIdIRCServer.

TIdIRCServer recognizes the following IRC commands as defined in the RFC specification:

- ADMIN
- AWAY
- CONNECT
- ERROR
- INFO
- INVITE
- ISON
- JOIN
- KICK
- KILL
- LINKS
- LIST
- MODE
- NAMES
- NICK
- NOTICE
- OPER
- PART
- PASS
- PING
- PONG
- PRIVMSG
- QUIT
- REHASH
- RESTART
- SERVER
- SQUIT
- STATS
- SUMMON
- TIME
- TOPIC
- TRACE
- USER
- USERHOST
- USERS
- VERSION

- WALLOPS
- WHO
- WHOIS
- WHOWAS

TIdIRCServer provides an event handler architecture for all IRC commands that allow the IRC server application to determine the appropriate mechanism used to respond to an IRC command. TIdIRCServer does not provide an implementation for the event handlers.

See Also

Technical Support ✂*Technical Support*

3.1.178. TIdListenerThread

Listener thread for TCP Servers.

TIdListenerThread = **class**(TIdThread)

Description

TIdListenerThread is a thread, based on TIdThread ✂*TIdThread*, that listens for client connections. A TIdListenerThread is used by TIdTCPServer ✂*TIdTCP*Server, and descendants, to detect new client connection requests and to spawn new TIdPeerThreads for each successful connection to the server.

TIdListenerThread maintains a reference to the server that owns the listening thread, and a list of socket handles that have been accepted by the server. TIdListenerThread also provides a general mechanism to encapsulate the time that the listening thread should wait to accept new socket connection requests.

TIdListenerThread differs from a typical TThread by extending the Execute method to allow access to BeforeRun, Run, and AfterRun methods. In addition, Start and Stop methods are provide allow finer control of thread initialization, resumption, and immediate termination.

3.1.179. TIdLogBase

Defines a component logging Framework.

```
TIdLogBase = class(TIdConnectionIntercept)
```

Description

TIdLogBase is an abstract class that defines a framework for logging information about Indy communication components.

TIdLogBase is a descendant of TIdConnectionIntercept *↗TIdConnectionIntercept*. TIdLogBase implements inherited virtual methods to write text strings to a log when TIdSocketHandle *↗TIdSocketHandle* operations are performed. These operations include connect, read, write, and disconnect.

Messages written to the log will transform all **EOL** *↗EOL* characters, (**Carriage Return + Line Feed**) by default, to the token '<EOL>'. A log message can optionally generate the date and time it was written to the log.

**Note:** TIdLogBase does not specify the destination for log messages. TIdLogBase descendants must implement the virtual method Log to resolve where log messages are stored, or written.

TIdLogBase is the ancestor class for the TIdLogDebug *↗TIdLogDebug* component.

TIdLogDebug *↗TIdLogDebug* demonstrates an implementation that can store messages in a file stream or write messages to the standard Debugger Output.

3.1.180. TIdLogDebug

```
TIdLogDebug = class(TIdLogBase)
```

Description

TIdLogDebug is an implementation of the ancestor class TIdLogBase *↗TIdLogBase*, and extends the framework for logging information about Indy communication components.

TIdLogDebug provides a flexible means of selecting the destination of log messages.

Messages can be written to either a file or the WIN32 API Debug Output stream.

TIdLogDebug can also trigger TIdLogItemEvent events as the primary logging mechanism, or as a supplement to the written log messages.

Like the ancestor class TIdLogBase *↗TIdLogBase*, messages written to the log will transform all EOL *↗EOL* characters, (**Carriage Return + Line Feed** ) by default, to the token '<EOL>'. A log message may optionally generate the date and time it was written to the log.

TIdLogDebug is very useful for capturing information about Indy communication components for debugging, trouble-shooting, and general feedback purposes.

3.1.181. TIdMappedPortTCP

Implements a proxy for TCP connections to a remote computer.

```
TIdMappedPortTCP = class(TIdTCPServer)
```

Description

TIdMappedPortTCP is a TIdTCPServer *↗TIdTCPServer* descendant that implements a proxy for connections to a remote computer system. TIdMappedPortTCP listens for connections on a designated port number and makes a connection to another server using a specified port number. This is useful for certain types of proxy programs.

3.1.182. TIdMappedPortTCPData

Encapsulates the outbound connection for a mapped port.

```
TIdMappedPortTCPData = class
```

Description

TIdMappedPortTCPData is an object that represents the outbound connection and all client connections for TIdMappedPortTCP *↗TIdMappedPortTCP*.

3.1.183. TIdMessage

Encapsulates an Internet Message.

```
TIdMessage = class(TIdBaseComponent)
```

Description

TIdMessage encapsulates a complete Internet Message as described in the Internet

Standards documents:

- Standard for the Format of ARPA Internet Text Messages, RFC 822, Revised by David H. Crocker, August 1982
- Standard for Interchange of USENET Messages, RFC 1036, by M. Horton, December 1987

TidMessage is used with message-based protocols, such as POP3, SMTP, and NNTP. TidMessage supports Multipurpose Internet Mail Extensions (MIME) as described in the Internet Standards documents:

- Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, RFC 2045, by N. Freed and N. Borenstein, November 1996
- Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, RFC 2046, by N. Freed and N. Borenstein, November 1996
- Multipurpose Internet Mail Extensions (MIME) Part Three: Message Header Extensions for Non-ASCII Text, RFC 2047, by K. Moore, November 1996
- Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures, RFC 2048, by N. Freed, J. Klensin and J. Postel, November 1996
- Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples, RFC 2049, by N. Freed and N. Borenstein, November 1996

### 3.1.184. TidMessageClient

Implements a message client.

```
TidMessageClient = class(TIdTCPClient)
```

**Description**

TidMessageClient is a TIdTCPClient *↗TIdTCPClient* descendant that implements a generic message client. TidMessageClient can read and write Internet messages compliant with the RFC 822 message specification.

### 3.1.185. TidMessagePart

Implements the base class for MIME message parts.

```
TidMessagePart = class(TCollectionItem)
```

**Description**

TidMessagePart is a TCollectionItem descendant that encapsulates a MIME message part. TidMessagePart is the ancestor class for common MIME message parts like TIdText *↗TIdText* and TIdAttachment *↗TIdAttachment*. TidMessagePart is also the collection item added to the TidMessageParts *↗TidMessageParts* collection.

TidMessagePart provides methods and properties common to all MIME message parts, including properties that represent values from Internet Message headers.

### 3.1.186. TidMessageParts

Collection for individual message parts in the message.

```
TidMessageParts = class(TOwnedCollection)
```

**Description**

TidMessageParts is a collection which contains the individual message parts for a MIME message, and contains ancestors of TidMessagePart *↗TidMessagePart* such as TIdText *↗TIdText* and TIdAttachment *↗TIdAttachment*.

### 3.1.187. TidMimeTable

Provides access to MIME types for the local computer.

```
TidMimeTable = class(TObject)
```

**Description**

TidMimeTable is a class used by Indy to provide access to the MIME types known from settings on the local computer system. TidMimeTable provides both cached and dynamic access to the MIME types registered on the local computer,

Use Create to instantiates the class instance.  
Use Free to release the class instance.  
Use BuildCache to load MIME type from system settings on the local computer.  
Use GetFileMIMEType to retrieve the MIME type for a specific file name.  
Use GetDefaultFileExt to retrieve the file extension associated with the specific MIME type.

3.1.188. TIdNetworkCalculator

Implements an IP address calculator.

```
TIdNetworkCalculator = class(TIdBaseComponent)
```

**Description**  
TIdNetworkCalculator is a component that can be used to calculate the validity of a network address, or calculate a list of valid network addresses.  
The 32-bit IP address used by TIdNetworkCalculator is based on the network address scheme described in the Internet Standards document An IP Address Extension Proposal, RFC 1365, by K. Siyan.  
Assign values to the properties NetworkClass, NetworkAddress, and NetworkMask to allow the component to validate the IP address desired. You can also use IsAddressInNetwork to determine if a specific network address is valid based on the NetworkMask settings. Use ListIP to view a list of IP addresses accessible over a network having the given network address and network mask.

3.1.189. TIdNNTP

Implements an NNTP client.

```
TIdNNTP = class(TIdMessageClient)
```

**Description**  
TIdNNTP implements a client or newsreader based on the Network News Transfer Protocol (NNTP) as described in the Internet Standards documents:

- Network News Transfer Protocol, RFC 977, by Brian Kantor and Phil Lapsley, February 1986 .
- Common NNTP Extensions, RFC 2980, by S. Barber, October 2000 .

NNTP is used for distributing messages publicly to central locations on a network.

3.1.190. TIdNNTPServer

```
TIdNNTPServer = class(TIdTCPServer)
```

**Description**  
TIdNNTPServer provides the developer with a framework for implementing a server based on the Network News Transfer Protocol (NNTP) as described in the Internet Standards documents:

- Network News Transfer Protocol, RFC 977, by Brian Kantor and Phil Lapsley, February 1986.
- Common NNTP Extensions, RFC 2980, by S. Barber, October 2000.

NNTP is used for distributing messages publicly to central locations on a network.  
TIdNNTPServer provides OnCommandXXX event handlers to respond to the following NNTP commands and extensions as described in RFCs 977 and 2980:

- ARTICLE
- BODY
- HEAD
- STAT
- GROUP
- LIST
- HELP
- IHAVE
- LAST
- NEWGROUPS
- NEWNEWS
- NEXT
- POST
- QUIT
- SLAVE
- AUTHINFO
- XOVER

- XHDR
- DATE
- LISTGROUP
- MODE
- TAKETHIS
- CHECK
- XTHREAD
- XGTITLE
- XPAT

TIdNNTPServer does **NOT** provide an implementation of the event handler methods. This allows the developer to determine the storage mechanism used for NNTP groups, articles, and overview databases.

See Also

### 3.1.191. TIdPeerThread

Thread used for client connections.

```
TIdPeerThread = class(TIdThread)
```

**Description**

TIdPeerThread is a thread that is created for every connection made to the TIdTCPServer *↗TIdTCPServer*. Every client connection runs in it's own thread on the TIdTCPServer *↗TIdTCPServer*. Requests for TIdPeerThreads are created by a TIdListenerThread *↗TIdListenerThread*, and are delegated to the TIdThreadMgr *↗TIdThreadMgr* for the TCP server.

### 3.1.192. TIdPOP3

Implements a POP3 client.

```
TIdPOP3 = class(TIdMessageClient)
```

**Description**

TIdPOP3 is a TIdMessageClient *↗TIdMessageClient* descendant that provides a client implementation of the Post Office Protocol version 3 (POP3) as described in the Internet Standards document:

- Post Office Protocol Version 3, RFC 1939, by J. Myers, May 1996

POP3 is used to retrieve E-Mail messages on a mail server and return the messages to the user's computer. POP3 is not intended to provide extensive manipulation operations of mail on the server; normally, mail is downloaded and then deleted. See IMAP for a more advanced (and complex) protocol.

### 3.1.193. TIdQOTD

Implements a Quote of the Day Protocol client.

```
TIdQOTD = class(TIdTCPClient)
```

**Description**

TIdQOTD implements a Quote of the Day Protocol (QUOTD) client as described in the Internet Standards document:

- RFC 865

QUOTD is a simple protocol for retrieving a short quote from a server.

### 3.1.194. TIdQOTDServer

Defines an implementation framework for a Quote of the Day Protocol server.

```
TIdQOTDServer = class(TIdTCPServer)
```

**Description**

TIdQOTDServer defines an implementation framework for a Quote of the Day Protocol or QUOTD server as described in the Internet Standards document:



- RFC 865

QUOTD is a simple protocol for retrieving a short quote from a server.

### 3.1.195. TIdQuotedPrintableDecoder

Implements a MIME decoder for the Quoted-Printable encoding scheme.

TIdQuotedPrintableDecoder = **class**(TIdCoder)

**Description**

TIdQuotedPrintableDecoder is a TIdCoder *↗TIdCoder* descendant that implements a decoder for MIME-encoded content using the Quoted-Printable transfer encoding scheme. Quoted-Printable transfer encoding is described in the Internet Standards document:

- RFC 2045, Multipurpose Internet Mail Extensions (MIME), Part One: Format of Internet Message Bodies, Section 6.7 Quoted-Printable Content Transfer Encoding

TIdQuotedPrintableDecoder is intended to transform data represented in the Quoted-Printable encoding scheme into the original representation of the encoded data.

### 3.1.196. TIdQuotedPrintableEncoder

Implements a MIME encoder for the Quoted-Printable encoding scheme.

TIdQuotedPrintableEncoder = **class**(TIdCoder)

**Description**

TIdQuotedPrintableEncoder is a TIdCoder *↗TIdCoder* descendant that implements an encoder for MIME-encoded content using the Quoted-Printable transfer encoding scheme. Quoted-Printable transfer encoding is described in the Internet Standards document:

- RFC 2045, Multipurpose Internet Mail Extensions (MIME), Part One: Format of Internet Message Bodies, Section 6.7 Quoted-Printable Content Transfer Encoding

TIdQuotedPrintableEncoder is intended to represent data that largely consists of octets that correspond to printable characters in the US-ASCII character set. It encodes the data in a way that minimizes the chances of the message content being modified by a mail transport. The quoted-printable encodings transforms input into characters in the "7bit" range.

Using TIdQuotedPrintableEncoder, data is generally represented using the following rules:

- Any byte except a CR *↗CR* or LF *↗LF* may be represented by an "=" followed by a two digit uppercase hexadecimal representation of the byte's value.
- Byte's with the decimal values of 33 through 60 inclusive, and 62 through 126, inclusive, MAY be represented as the US-ASCII characters which correspond to those bytes. (
- Byte's with values of 9 and 32 MAY be represented as US-ASCII TAB *↗TAB* (HT) and SPACE characters, but MUST NOT appear at the end of an encoded line. Any whitespace characters muse be followed by a printable character. An "=" at the end of an encoded line, indicates a soft line break, and may follow one or more whitespace characters.
- A CRLF sequence in a message must be represented by a CRLF sequence in the Quoted-Printable encoding. Sequences like "=0D", "=0A", "=0A=0D" and "=0D=0A" will routinely appear in non-text data represented in quoted- printable, but are not significant to the formatting of the message.
- The Quoted-Printable encoding REQUIRES that encoded lines be no more than 76 characters long. If longer lines are to be encoded with the Quoted-Printable encoding, "soft" line breaks must be used. An equal sign as the last character on a encoded line indicates such a non-significant ("soft") line break in the encoded text.

To insure reliable message transport through EBCIDIC networks, TIdQuotedPrintableEncoder also encodes the following ASCII characters:

! " # \$ % & [ ] ^ \_ ` | ~

### 3.1.197. TIdRawBase

Ancestor class for Raw socket clients.

TIdRawBase = **class**(TIdComponent)

**Description**

The TIdRawBase is an ancestor of components which use raw sockets. This allows the programmer to code for protocols that are not supported by the standard stack (ICMP, IGMP or any custom protocol).

Depending on the underlying operating system and on the security privilege of the user, it might also allow the programmer to build the IP header (currently, this is only supported by Windows 2000).

**3.1.198. TIdRawClient**

Specifies a client that uses Raw sockets.

`TIdRawClient = class(TIdRawBase)`

**Description**

TIdRawClient is a TIdRawBase *↗TIdRawBase* descendant that specifies a client capable of using Raw socket connections. This allows the programmer to code for protocols that are not supported by the standard stack (ICMP, IGMP or any custom protocol).

TIdRawClient does not provide an implementation; implementation will be provided in descendant classes tailored to a specific protocol, like TIdIcmpClient *↗TIdIcmpClient*.

Depending on the underlying operating system and on the security privilege of the user, it might also allow the programmer to build the IP header (currently, this is only supported by Windows 2000).

**3.1.199. TIdServerIntercept**

Specifies a socket handler for use with peer connections create by the listener thread in a TCP server.

`TIdServerIntercept = class(TIdBaseComponent)`

**Description**

TIdServerIntercept is an abstract class that creates instances of TIdConnectionIntercept *↗TIdConnectionIntercept* socket handlers for peer connections created by the listener thread in a TCP Server.

TIdConnectionIntercept *↗TIdConnectionIntercept* descendants will be used by peer

connection to act as an intermediary between the socket binding for the connection and low-level operations that communicate with the protocol stack.

TIdServerIntercept provides virtual methods to initialize the server socket handler, and to provide new TIdConnectionIntercept *↗TIdConnectionIntercept* handlers for peer connections when the connection is accepted.

Call Init to perform initialization tasks prior to using the TIdServerIntercept instance to accepting connections. Call Accept to create the TIdConnectionIntercept *↗TIdConnectionIntercept* handler used by the peer connection.

**Note:** Do not create instances of TIdServerIntercept. Use descendant classes, like TIdServerInterceptOpenSSL *↗TIdServerInterceptOpenSSL*, or create custom descendants that implement the virtual methods of TIdServerIntercept.

**3.1.200. TIdServerInterceptOpenSSL**

Implements Secure Sockets Layer support for Indy Server Intercept components.

`TIdServerInterceptOpenSSL = class(TIdSSLServerIntercept)`

**Description**

The TIdServerInterceptOpenSSL server intercept class implements the Open SSL implementation of Secure Sockets Layer through support .DLL's available at <http://www.intelicom.si/>. You have to deploy these support .DLL's in order to use this component's functionality in your application. To do this, we recommend that you install the .DLL's in the user's system directory.

Indy does not give you any legal rights to use SSL. Some nations including the U.S. regulate or prohibit the export of strong-encryption such as SSL. In addition, using SSL in some nations may be illegal. It is up to you to determine the legal situation in your nation.

**3.1.201. TIdSimpleServer**

Implements a single-threaded TCP Server.

`TIdSimpleServer = class(TIdTCPConnection)`

**Description**

TIdSimpleServer is a TIdTCPConnection *↗TIdTCPConnection* descendant that implements a single-threaded TCP (Transmission Control Protocol) server. TIdSimpleServer is essentially a TCP connection that gains the ability to listen for serial connection requests. TIdSimpleServer can be used as a base class to create custom single-threaded TCP server descendants.

3.1.202. TIdSMTP

Implements a Simple Mail Transfer Protocol client.

```
TIdSMTP = class(TIdMessageClient)
```

Description

TIdSMTP is a TIdMessageClient *↗TIdMessageClient* descendant that implements a Simple Mail Transfer Protocol or SMTP client, as described in the Internet Standards documents:

- Simple Mail Transfer Protocol (SMTP), RFC 821
- SMTP Service Extensions, RFC 1869
- SMTP Service Extension for Authentication, RFC 2554

Use TIdPOP3 *↗TIdPOP3* to retrieve E-Mail messages from a mail server to the user's computer.

3.1.203. TIdSNTP

Implements a Simple Network Time Protocol client.

```
TIdSNTP = class(TIdUDPClient)
```

Description

The TIdSNTP implements a client for the Simple Network Time Protocol (SNTP), as described in the Internet standards document:

- Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI, RFC 2030

SNTP is an extremely reliable protocol for time-synchronization on the Internet with accuracy from 1 to 50 milliseconds, even over great distances. SNTP Version 4 includes certain optional extensions to the basic Version 3 model. TIdSNTP does not implement the optional extensions for either Key Identifier or Message Digest portions of the NTP/SNTP message format. TIdSNTP, a descendant of TIdUDPClient *↗TIdUDPClient*, utilizes SNTP in the unicast client mode.

3.1.204. TIdSocketHandle

Represents a low-level socket binding.

```
TIdSocketHandle = class(TCollectionItem)
```

Description

TIdSocketHandle encapsulates the low-level socket binding and its associated functions. A socket binding is the handle used for sending and receiving data, making and closing a connection, or waiting for a connection (listening). TIdSocketHandle is used by TIdTCPConnection *↗TIdTCPConnection* descendants in Indy to encapsulate the low-level connection and methods used to access the protocol stack in Binding.

3.1.205. TIdSocketHandles

Collection of socket handles.

```
TIdSocketHandles = class(TOwnedCollection)
```

Description

The TIdSocketHandles collection contains TIdSocketHandle *↗TIdSocketHandle* objects for the TIdTCPServer *↗TIdTCPServer* and TIdUDPServer *↗TIdUDPServer* components in their Bindings property. This permits a server to listen on more than one Port and IP address. The bindings must meet the following criterion:

- The binding must contain a port.
- The bindings for a server must **NEVER** allow duplicates for port or IP/port

assignments. (e.g. :30 - :30, 127.0.0.1:30, 127.0.0.1:30)

TIdSocketHandles is a TOwnedCollection descendant that maintains additional information about the ItemClass (TIdSocketHandle) for Items contained in the collection.  
TIdSocketHandles contains the DefaultPort number to assign to new TIdSocketHandle *↗*TIdSocketHandle instances added to the collection.

3.1.206. TIdSSLConnectionIntercept

TIdSSLConnectionIntercept = **class**(TIdConnectionIntercept)  
**Description**  
The text for this class has been generated automatically. This means that it is not documented.

3.1.207. TIdSSLContext

Implements a generic client or server SSL context.

TIdSSLContext = **class**(TObject)  
**Description**  
TIdSSLContext is a class that provides an implementation of a Secure Socket *↗*Socket Layer Context. TIdSSLContext is used by TIdSSLConnectionIntercept *↗*TIdSSLConnectionIntercept descendants and TIdSSLSocket *↗*TIdSSLSocket to allow a client or server SSL socket to connect using the desired SSL options and certificates.

3.1.208. TIdSSLOptions

Represents SSL certificate and usage options.

TIdSSLOptions = **class**(TPersistent)  
**Description**

TIdSSLOptions is a TPersistent descendant that represents SSL options for Indy connection intercept classes. TIdSSLOptions implements properties that identify certificate files for the Root and User Certificates, User Key file, as well as SSL version number, method, and mode. TIdSSLOptions is used in TIdConnectionInterceptOpenSSL *↗*TIdConnectionInterceptOpenSSL and TIdServerInterceptOpenSSL *↗*TIdServerInterceptOpenSSL implementations.

3.1.209. TIdSSLServerIntercept

Ancestor for Indy Secure Sockets Layer server intercept classes.

TIdSSLServerIntercept = **class**(TIdServerIntercept)  
**Description**  
The TIdSSLServerIntercept class is ancestor for Secure Sockets Layer server intercept classes such as TIdServerInterceptOpenSSL *↗*TIdServerInterceptOpenSSL. You do not create instances of this object.  
Indy does not give you any legal rights to use SSL. Some nations including the U.S. regulate or prohibit the export of strong-encryption such as SSL. In addition, using SSL in some nations may be illegal. It is up to you to determine the legal situation in your nation.

3.1.210. TIdSSLSocket

Implements a socket connection for the SSL transport protocol.

TIdSSLSocket = **class**(TObject)  
**Description**  
TIdSSLSocket is a class that provides a socket implementation that supports the Secure Socket *↗*Socket Layer transport protocol. TIdSSLSocket allows both client and server SSL connections using an SSL context, as well as support for session identifiers and X.509 certificate files.

### 3.1.211. TIdStack

Ancestor for platform-specific protocol stack interfaces.

TIdStack = **class**

**Description**

TIdStack is an ancestor class for platform-specific protocol stack interfaces. This class is created and destroyed as needed. This is mentioned only because there a few useful low-level properties and methods.

### 3.1.212. TIdStackVersion

Represent version information for the protocol stack.

TIdStackVersion = **class**

**Description**

TIdStackVersion is an abstract class used to represent information about the protocol stack implementation on the local computer system. TIdStackVersion is used as an object property in TIdStack.StackVersion. TIdStackVersion can be extremely helpful for diagnosing problems on some computers.

### 3.1.213. TIdStackVersionWinsock

Represents Winsock-specific protocol stack version information.

TIdStackVersionWinsock = **class**(TIdStackVersion)

**Description**

TIdStackVersionWinsock is a TIdStackVersion *↗TIdStackVersion* descendant used to represent Winsock-specific protocol stack version and initialization information for Indy.

### 3.1.214. TIdStackWinsock

Encapsulates the Windows-specific Winsock protocol stack used by Indy.

TIdStackWinsock = **class**(TIdStack)

**Description**

TIdStackWinsock is a TIdStack *↗TIdStack* descendant that encapsulates the Winsock library into an object at a low-level. This class is created and destroyed as needed. This is mentioned only because there a few useful low-level properties and methods.

### 3.1.215. TIdTCPClient

Implement a TCP Client.

TIdTCPClient = **class**(TIdTCPConnection)

**Description**

TIdTCPClient encapsulates a complete TCP (Transmission Control Protocol) client including socks support. TIdTCPClient can be used for as an ancestor class for specific protocol implementations. Many Indy client components, such as TIdDayTime *↗TIdDayTime*, TIdEcho *↗TIdEcho*, TIdFinger *↗TIdFinger*, TIdFTP *↗TIdFTP*, TIdGopher *↗TIdGopher*, TIdHTTP *↗TIdHTTP*, TIdNNTP *↗TIdNNTP*, TIdPOP3 *↗TIdPOP3*, TIdQUOTD, TIdSMTP *↗TIdSMTP*, TIdTelnet *↗TIdTelnet*, and TIdWhois *↗TIdWhois* are TIdTCPClient descendants.

### 3.1.216. TIdTCPConnection

Implements a TCP Connection.

TIdTCPConnection = **class**(TIdComponent)

**Description**

TIdTCPConnection encapsulates a complete Transmission Control Protocol (TCP) connection. TIdTCPConnection is the ancestor class for TIdTCPClient *↗TIdTCPClient* and TIdTCPServer *↗TIdTCPServer*.

### 3.1.217. TIdTCPServer

Implements a multi-threaded TCP Server.

```
TIdTCPServer = class(TIdComponent)
```

**Description**

TIdTCPServer encapsulates a complete, multi-threaded TCP (Transmission Control Protocol) server. TIdTCPServer uses a thread to listen for client connections, and in conjunction with a TIdThreadMgr *↗TIdThreadMgr*, allocates a separate thread to handle each client connection to the server.

TIdTCPServer can be used as a base class to create custom TCP server descendants. Many of the Indy server components, such as TIdChargenServer *↗TIdChargenServer*, TIdDayTimeServer *↗TIdDayTimeServer*, TIdDICTServer *↗TIdDICTServer*, TIdEchoServer, TIdFingerServer *↗TIdFingerServer*, TIdGopherServer *↗TIdGopherServer*, TIdHostNameServer *↗TIdHostNameServer*, TIdHTTPServer *↗TIdHTTPServer*, TIdIRCServer *↗TIdIRCServer*, TIdNNTPServer *↗TIdNNTPServer*, TIdQUOTDServer, TIdTelnetServer *↗TIdTelnetServer*, and TIdWhoisServer, are descendants of the TIdTCPServer component.

### 3.1.218. TIdTCPServerConnection

Encapsulates a server connection to a client.

```
TIdTCPServerConnection = class(TIdTCPConnection)
```

**Description**

TIdTCPServerConnection encapsulates a server connection to a client on a TIdTCPServer *↗TIdTCPServer*. TIdTCPServerConnection is used by TIdPeerThread *↗TIdPeerThread* to provide the server context for accessing any properties, methods, or events of the TCP server.

### 3.1.219. TIdTelnet

Implements a TELNET protocol client.

```
TIdTelnet = class(TIdTCPClient)
```

**Description**

TIdTelnet is a client implementation of the TELNET Protocol as described in the Internet Standards documents:

- TELNET Protocol Specification, RFC 854, by J. Postel and J. Reynolds, May 1983.
- TELNET Option Specifications, RFC 855, by J. Postel and J. Reynolds, May 1983.
- TELNET Binary Transmission, RFC 856, by J. Postel and J. Reynolds, May 1983.
- TELNET Echo Option, RFC 857, J. Postel and J. Reynolds, May 1983.
- TELNET Suppress Go Ahead Option, RFC 858, by J. Postel and J. Reynolds, May 1983.
- TELNET Status Option, RFC 859, by J. Postel and J. Reynolds, May 1983.
- TELNET Timing Mark Option, RFC 860, by J. Postel and J. Reynolds, May 1983.
- TELNET Extended Options: List Option, RFC 861, by J. Postel and J. Reynolds, May 1983.
- Assigned Numbers, RFC 1700, by J. Postel and J. Reynolds, October 1994.

TIdTelnet is a TIdTCPClient *↗TIdTCPClient* descendant used to transmit data with interspersed TELNET control information.

TIdTelnet provides a general, eight-bit byte-oriented communications facility designed to interface terminal devices and terminal-oriented processes. The TELNET protocol may also be used for terminal-to-terminal communication ("linking") and process-to-process communication (distributed computation).

TIdTelnet also provides negotiated options that allows additional services over and above those available within an NVT (Network Virtual Terminal).

### 3.1.220. TIdTelnetReadThread

TELNET client thread that listens for responses to key input and protocol negotiations.

```
TIdTelnetReadThread = class(TIdThread)
```

**Description**

TIdTelnetReadThread is a TIdThread *↗TIdThread* descendant that represents the thread of execution in the TIdTelnet *↗TIdTelnet* client that listens for responses to key input and protocol option negotiations.

TIdTelnetReadThread implements the thread handling framework inherited from TIdThread *↗TIdThread* to control execution of the thread.

TIdTelnetReadThread is used by the OnDataAvailable event handler in TIdTelnet *↗TIdTelnet* to perform read operations for the TELNET client.

Do not create instances of TIdTelnetReadThread. TIdTelnet *↗TIdTelnet* creates an instance of TIdTelnetReadThread when the client is connected to the host.

3.1.221. TIdTelnetServer

Specifies a Telnet Protocol server architecture.

TIdTelnetServer = **class**(TIdTCPServer)

**Description**

TIdTelnetServer is sever implementation of the Telnet Protocol as described in the Internet Standards documents:

- TELNET Protocol Specification, RFC 854, by J. Postel and J. Reynolds, May 1983
- TELNET Option Specifications, RFC 855, by J. Postel and J. Reynolds, May 1983
- TELNET Echo Option, RFC 857, J. Postel and J. Reynolds, May 1983.
- TELNET Suppress Go Ahead Option, RFC 858, by J. Postel and J. Reynolds, May 1983.
- TELNET Status Option, RFC 859, by J. Postel and J. Reynolds, May 1983.
- TELNET Timing Mark Option, RFC 860, by J. Postel and J. Reynolds, May 1983.
- TELNET Extended Options: List Option, RFC 861, by J. Postel and J. Reynolds, May 1983.
- Assigned Numbers, RFC 1700, by J. Postel and J. Reynolds, October 1994.

The TELNET Protocol provides a general communications facility to allow a standard method of interfacing terminal devices and terminal-oriented processes.

Telnet is used to connect to a remote computer system and allow the client to act as a dumb terminal on that system.

3.1.222. TIdText

Encapsulates a MIME-encoded text message part.

TIdText = **class**(TIdMessagePart)

**Description**

TIdText is a TIdMessagePart *↗TIdMessagePart* descendant that encapsulates a MIME textual message part. TIdAttachment *↗TIdAttachment* and TIdText are used as collection items in a TIdMessageParts *↗TIdMessageParts* collection.

TIdText provides the Body property to represent the textual content of the message part.

3.1.223. TIdThread

Ancestor for all Indy threads.

TIdThread = **class**(TThread)

**Description**

TIdThread is a descendant of the Delphi **TThread**, and the ancestor of thread classes used in Indy. Create a descendant of TIdThread to represent a thread of execution in a multi-threaded application.

TIdThread extends the functionality of TThread to include flexible methods for control of thread state and notification of changes to thread state.

TIdThread implements the inherited abstract Execute method to provide a known thread execution mechanism. Execute can detect thread termination, suspend and resume threads, and provide exception handling for an exception raised during thread execution. Execute also provides finer control of thread execution by extending TThread to include the equivalent of thread event handlers.

TIdThread and descendants should override the abstract Run method, and implement the virtual methods BeforeRun and AfterRun.

The Execute method iterates through a loop, making calls to BeforeRun, Run, and AfterRun to provide thread functionality. This loop can be stopped by calling the Stop method. Use the Start method to begin or resume execution of the loop.

TIdThread descendants can be used in conjunction with a TIdThreadMgr *↗TIdThreadMgr*

descendants to address the overhead issues associated with thread creation and resource allocation in a multi-threaded application.

**Note:** Do not use properties and methods of other objects directly in the Run method of a thread. Use the Synchronize method to call a procedure that can access objects and resources which are not thread-safe.

3.1.224. TIdThreadMgr

Specifies the base class for the Indy thread management framework.

```
TIdThreadMgr = class(TIdBaseComponent)
```

Description

TIdThreadMgr is the ancestor class for the thread manager classes in Indy, such as TIdThreadMgrDefault *↗TIdThreadMgrDefault* and TIdThreadMgrPool *↗TIdThreadMgrPool*, and provides a thread management framework. TIdThreadMgr defines properties and methods needed to allocate and release threads, monitor active threads, terminate a list of running threads, safeguard thread operations, and determine the thread class used by the thread manager to create new thread instances. Instances of a TIdThreadMgr descendant can be assigned to the TIdTCPServer.ThreadMgr property to provide the functionality specific to the thread manager. Descendants of TIdThreadMgr must implement the virtual abstract methods GetThread and ReleaseThread, and use Lock or another mechanism to protect resources while maintaining the list of active threads during these operations. Use TerminateThreads to notify the ActiveThreads list that all TIdThread *↗TIdThread* instances should close their socket connections, release the thread with RelaseThread, and remove the thread from ActiveThreads. Assign an instance of TIdThreadClass *↗TIdThreadClass* to ThreadClass before using the thread manager to allocate new threads.

3.1.225. TIdThreadMgrDefault

Default thread manager for Indy servers.

```
TIdThreadMgrDefault = class(TIdThreadMgr)
```

Description

TIdThreadMgrDefault is a descendant of TIDThreadMgr that acts as the default thread manager in Indy for TIdTCPServer *↗TIdTCPServer* and descendants. TIdThreadMgrDefault utilizes the thread management framework defined in TIdThreadMgr *↗TIdThreadMgr*. TIdThreadMgrDefault creates and destroys threads for every connection, and stores active thread instances in the ActiveThreads list. TIdThreadMgrDefault uses Lock to protect the thread list while maintaining ActiveThreads. TIdThreadMgrDefault implements the inherited abstract methods GetThread and ReleaseThread. GetThread creates a new thread instance for the thread manager. ReleaseThread handles freeing and releasing a thread instance. Use TerminateThreads to notify all TIdThread *↗TIdThread* instances to close their socket connection, release the thread, and remove the thread from ActiveThreads. Assign an instance of TIdThreadClass *↗TIdThreadClass* to ThreadClass before using the thread manager to allocate new threads. TIdThreadMgrDefault is the default thread manager created for instances of TIdTCPServer *↗TIdTCPServer* and descendants. To use another thread manager, create an instance of a TIdThreadMgr *↗TIdThreadMgr* descendant, and assign the object reference to TIdTCPServer.ThreadMgr. If you are using protocols such as HTTP, Time, or Gopher, where the connection is closed after the request is fulfilled, you may wish to consider using the TIdThreadMgrPool *↗TIdThreadMgrPool* thread manager.

3.1.226. TIdThreadMgrPool

Provides thread management using a pool of TIdThread *↗TIdThread* instances.

```
TIdThreadMgrPool = class(TIdThreadMgr)
```

Description

TIdThreadMgrPool is a thread manager that utilizes the thread management framework defined in TIdThreadMgr *↗TIdThreadMgr*. TIdThreadMgrPool provides thread management using a pool of TIdThread *↗TIdThread* instances. Threads are created by the thread manager, when necessary, or retrieved from the thread pool. TIdThreadMgrPool also keeps a list of active threads. The thread pool can have a maximum size to conserve system resources. TIdThreadMgrPool uses Lock to protect thread manager resources in GetThread and ReleaseThread. TIdThreadMgrPool implements the inherited abstract methods GetThread and



ReleaseThread. GetThread provides a thread instance for the thread manager. ReleaseThread handles freeing and releasing a thread instance, or returning the instance to the thread pool.

Use TerminateThreads to notify all TldThread *↗TldThread* instances to close their socket connection, release the thread, and remove the thread from ActiveThreads.

Assign an instance of TldThreadClass *↗TldThreadClass* to ThreadClass before using the thread manager to allocate new threads.

To use TldThreadMgrPool with TldTCPSever *↗TldTCPSever*, create an instance of TldThreadMgrPool, and assign the object reference to TldTCPSever.ThreadMgr. If you are using protocols which do not close the socket connection after each request, you may wish to consider using the TldThreadMgrDefault *↗TldThreadMgrDefault* thread manager.

3.1.227. TldTime

Implements a Time client.

TldTime = **class**(TldTCPClient)

**Description**

TldTime is a client implementation of the Time Protocol as described in the Internet Standards document:

- Time Protocol, (RFC 868)

Time is a simple protocol for synchronizing time on a local network. For a time protocol with higher accuracy over several networks, use TldSNTP *↗TldSNTP*. To retrieve the current date and time in human-readable form, use TldDayTime *↗TldDayTime*.

Please note that the Time Protocol in its current form can handle most dates after the year 2035. This limitation is stated in RFC 868.

3.1.228. TldTimeServer

Implements a Time protocol server.

TldTimeServer = **class**(TldTCPSever)

**Description**

TldTimeServer is a TldTCPSever *↗TldTCPSever* descendant that provides a server implementation of the Time Protocol as described in the Internet Standard document:

- Time Protocol, RFC 868, by J. Postel and K. Harrenstien, May 1983

Time is a simple protocol for synchronizing time on a local network. If you need the time and day only in a human readable form, we recommend that you use the TldDayTimeServer *↗TldDayTimeServer* component.

**Note:** The Time protocol in its current form cannot handle most dates after the year 2035. This limitation is stated in RFC 868.

3.1.229. TldTrivialFTP

Implements a Trivial File Transfer Protocol client.

TldTrivialFTP = **class**(TldUDPCClient)

**Description**

TldTrivialFTP is a TldUDPCClient *↗TldUDPCClient* descendant that implements a Trivial File Transfer Protocol (TFTP) client as described in the Internet Standards documents:

- RFC 1350, Trivial File Transfer Protocol
- RFC 1782, TFTP Option Extension
- RFC 1783, TFTP Blocksize Option

Trivial FTP is an extremely lightweight and simple UDP-Based file transfer protocol that is normally used read and write files from/to a remote server. TFTP cannot list directories, and currently has no provisions for user authentication.

TldTrivialFTP supports the TFP Option Negotiation Protocol where the client appends options at the end of the Read Request or Write request packet. TldTrivialFTP also supports the TFTP Blocksize Option which allows the client and server to negotiate a blocksize more applicable to the network medium.

### 3.1.230. TldTrivialFTPServer

Implements a Trivial File Transfer Protocol server.

```
TldTrivialFTPServer = class(TIdUDPServer)
```

**Description**

TldTrivialFTPServer is a TIdUDPServer *↗TIdUDPServer* descendant that implements a Trivial File Transfer Protocol (TFTP) server as described in the Internet Standards documents:

- RFC 1350, Trivial File Transfer Protocol
- RFC 1782, TFTP Option Extension
- RFC 1783, TFTP Blocksize Option

Trivial FTP is an extremely lightweight and simple UDP-Based file transfer protocol that is normally used read and write files from/to a remote server. TFTP cannot list directories, and currently has no provisions for user authentication.

TldTrivialFTPServer supports the TFP Option Negotiation Protocol where the client appends options at the end of the Read Request or Write request packet. TldTrivialFTPServer also supports the TFTP Blocksize Option which allows the client and server to negotiate a blocksize more applicable to the network medium.

### 3.1.231. TldTunnelMaster

Implements a TCP Server for IP encapsulation tunnels.

```
TldTunnelMaster = class(TIdTCPServer)
```

**Description**

TldTunnelMaster is a TIdTCPServer *↗TIdTCPServer* descendant that specifies a multithreaded server that acts as a controller for IP encapsulation "tunnels".

TldTunnelMaster acts a Proxy for user connections. The IP encapsulation tunnels represent a User connection and the Server advocate used to process IP datagrams for the tunnel.

TldTunnelMaster also allows construction of Virtual Private Networks through the IP encapsulation tunnel facilities for SOCKS authentication, tunnel management, data transformation, statistical logging, and IP header CRC calculation with optional IP header message types.

TldTunnelMaster works in conjunction with an instance of TldTunnelSlave *↗TldTunnelSlave* that acts as the multithreaded server for managing the endpoint for the IP tunnel.

### 3.1.232. TldTunnelSlave

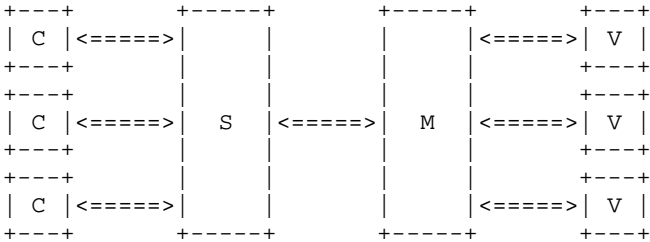
Implements a server that arbitrates client access using tunnel connections.

```
TldTunnelSlave = class(TIdTCPServer)
```

**Description**

TldTunnelSlave is a TIdTCPServer *↗TldTCPServer* descendant that implements a server for communicating with tunnel client connections. TldTunnelSlave utilizes TCP connections to establish an internal link to the tunnel master server that hosts connections to service threads for the client connections.

The following diagram is a general diagram of the relationship between client connection threads, the Tunnel Slave Server, the Tunnel Master Server, and service threads used to support encapsulated tunnel connections:



C: Client connection thread  
S: Tunnel Slave Server  
M: TUnnel Master Server  
V: Service Thread

TldTunnelSlave provides event handlers that allow responses to changes in both server state and client connection state. TldTunnelSlave also provides event handlers for data transformation and session link control of the tunnel for associated TldTunnelMaster *↗TldTunnelMaster* server.

When a TIdTunnelSlave becomes active, it initializes the Host and Port properties for the server's internal TCP connection that will acts as the encapsulated tunnel link tot he TIdTunnelMaster *↗TIdTunnelMaster* server. The connection to the tunnel master server is opened.

TIdTunnelSlave maintains SlaveThread to listen for data from the TIdTunnelMaster *↗TIdTunnelMaster* server service threads. SlaveThread is also used to authenticate the tunnel connection to the master server. If no exception is raised, then the TIdTunnelSlave server will begin to listen for client connections.

When a client requests a connection to TIdTunnelSlave, the server must indicate that AcceptConnections is allowed. When Socks4 indicates that client connection are using a Socks proxy, the server expects to read the IP address and Socks authentication information from the client connection. Socks authentication data cannot exceed 255 characters, and must be zero-terminated. The server will write a Socks response to the client connection and forward the connection request to the tunnel master server.

When a client connection executes a request on a connection to the tunnel slave server, TIdTunnelSlave will prepare encapsulated tunnel headers for the request and write the encapsulated message to the connection for the tunnel master server. The client connection will be closed if an exception is raised during execution of the request.

When a client connection request disconnect from the TIdTunnelSlave, the serve will prepare encapsulated tunnel headers for the request and send the message to the tunnel master server to allow closing of service threads for the client connection.

3.1.233. TIdUDPBase

Specifies a UDP base class.

```
TIdUDPBase = class(TIdComponent)
```

Description

TIdUDPBase specifies an encapsulation of the User Datagram Protocol (UDP), and provides properties and methods common to both UDP clients and UDP servers. TIdUDPServer *↗TIdUDPServer* and TIdUDPClient *↗TIdUDPClient* are descendants of TIdUDPBase.

For more information on the User Datagram Protocol (UDP), refer to the Internet Standards document:

- RFC 768 - User Datagram Protocol (UDP), RFC 768, by Jon Postel, August 1980

TIdUDPBase implements the common UDP facilities required to create a socket binding

using the Socket *↗Socket* Datagram protocol family, perform datagram Broadcasts, and handle Host and Port assignments for the connection-less protocol.

3.1.234. TIdUDPClient

Implements a UDP client.

```
TIdUDPClient = class(TIdUDPBase)
```

Description

TIdUDPClient is a TIdUDPBase *↗TIdUDPBase* descendant that provides a client implementation of the User Datagram Protocol (UDP) as described in the Internet Standards document:

- User Datagram Protocol, RFC 768, by Jon Postel, August 1980

TIdUDPClient implements the Send method to transmit data to the remote computer system specified by Host and Port using the connection-less UDP protocol.

TIdUDPClient is used as an ancestor class for protocol implementations based on UDP. For example, TIdDNSResolver *↗TIdDNSResolver*, TIdSNTP *↗TIdSNTP*, and TIdTrivialFTP *↗TIdTrivialFTP* are descendants of TIdUDPClient.

3.1.235. TIdUDPListenerThread

Specifies the listener thread for TIdUDPServer *↗TIdUDPServer*.

```
TIdUDPListenerThread = class(TIdThread)
```

Description

TIdUDPListenerThread is a TIdThread *↗TIdThread* descendant that is the listening thread for TIdUDPServer *↗TIdUDPServer*. TIdUDPListenerThread is used to detect arrival of data on socket bindings allocated by TIdUDPServer *↗TIdUDPServer*. When data is detected, the server is notified that data is available in Buffer using UDPRead.

### 3.1.236. TIdUDPServer

Implements a UDP server.

```
TIdUDPServer = class(TIdUDPBase)
```

**Description**

TIdUDPServer is a TIdUDPBase *↗TIdUDPBase* descendant that is a server implementation of the User Datagram Protocol(UDP) as described in the Internet standards document:

- User Datagram Protocol, RFC 768, by Jon Postel, August 1980

TIdUDPServer uses an instance of TIdUDPListener to perform read operations for multiple socket bindings created by the server.

TIdUDPServer can be used as an ancestor class for other server components based on the User Datagram Protocol. TIdTrivialFTPServer *↗TIdTrivialFTPServer* is a descendant of the TIdUDPServer component.

### 3.1.237. TIdURI

Represents a Universal Resource Identifier.

```
TIdURI = class
```

**Description**

The TIdURI object parses a URI specified in the Create constructor into its components: Protocol, Host, Port, Path, Document, and Bookmark. You may also modify the URI using object properties.

### 3.1.238. TIdUUDecoder

Implements a decoder for the UUEncode encoding scheme.

```
TIdUUDecoder = class(TId3To4Coder)
```

**Description**

TIdUUDecoder is a TIdCoder3To4 descendant that decodes the 7-bit textual representation of a UUEncoded message into the original binary data.

UUEncode encoding represents 24-bit groups of input as output strings of 4 encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating 3 8bit input groups. These 24 bits are then treated as 4 concatenated 6-bit groups, each of which is translated into a single digit in the UUEncode alphabet.

TIdUUDecoder reverses the UUEncoding process by converting the 4-byte US-ASCII representaiton of the data into it's 3-byte binary equivalent.

### 3.1.239. TIdUUEncoder

Implements an encoder for the UUCP encoding scheme.

```
TIdUUEncoder = class(TId3To4Coder)
```

**Description**

TIdUUEncoder is a TIdCoder3To4 descendant that encodes binary data into a 7-bit textual representation, called UUEncode, as established in UUCP (Unix-to-Unix Copy Program). UUEncode, like Base64, is an encoding mechanism designed to represent arbitrary sequences of byte data in a form that need not be humanly readable. UUEncode of often used to send EMail attachments in an environnment that is not MIME-compliant, and also allows larger files to be divided into multi-part transmissions.

UUEncode encoding represents 24-bit groups of input as output strings of 4 encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating 3 8bit input groups. These 24 bits are then treated as 4 concatenated 6-bit groups, each of which is translated into a single digit in the UUEncode alphabet.

UUEncode differs from Base64 in that it utilizes a different coding table, or alphabet, to represent the encoded output values. The UUEncode Alphabet can be represented by the following encodings and values:

Value	Encoding	Value	Encoding
0	~	33	@
1	!	34	A
2	"	35	B
3	#	36	C
4	\$	37	D
5	%	38	E

6	&	39	f
7	'	40	g
8	'	41	h
9	(	42	i
10	)	43	j
11	*	44	k
12	+	45	l
13	,	46	m
14	-	47	n
15	.	48	o
16	/	49	p
17	0	50	q
18	1	51	r
19	2	52	s
20	3	53	t
21	4	54	u
22	5	55	v
23	6	56	w
24	7	57	x
25	8	58	y
26	9	59	z
27	:	60	[
28	;	61	
29	<	62	]
30	=	63	^
31	>	64	_
32	?		

When encoding a bit stream using UUEncode encoding, the stream must be presumed to be ordered with the most-significant-bit first. That is, the first bit in the stream will be the high-order bit in the first 8bit byte, and the eighth bit will be the low-order bit in the first 8bit byte, etc. UUEncode also adds output headers, Unix privilege levels, and line length encoding for each line in the UUEncode output buffer.

3.1.240. TIdVCard

Implements an electronic business card.

TIdVCard = **class**(TIdBaseComponent)

**Description**  
The TIdVCard component processes Virtual Cards which are electronic business cards through the ReadFromTStrings method. This component is compliant with the VCard specification 2.1 at <http://www.imc.org/pdi/pdiproddev.html> and VCard specification 3.0 (RFC 2425 and 2426).  
Creating VCards will be added at a later time.

3.1.241. TIdVCardAddresses

Encapsulates VCard addresses.

TIdVCardAddresses = **class**(TOwnedCollection)

**Description**  
The TIdVCardAddresses collection contains TIdVCardAddresses objects encapsulating a VCard owner's addresses in a TIdVCard *↗TIdVCard* component.

3.1.242. TIdVCardBusinessInfo

Encapsulates VCard organization information.

TIdVCardBusinessInfo = **class**(TPersistent)

**Description**  
The TIdVCardBusinessInfo object encapsulates a VCard owner's organizational affiliation information in the TIdVCard *↗TIdVCard* component.

3.1.243. TIdVCardEmailAddresses

Encapsulates VCard E-Mail addresses.

TIdVCardEmailAddresses = **class**(TOwnedCollection)

**Description**

The TIdVCardEMailAddresses collection contains TIdVCardEMailItem *↗TIdVCardEMailItem* objects encapsulating a VCard owner's E-Mail addresses in a TIdVCard *↗TIdVCard* component.

3.1.244. TIdVCardEMailItem

encapsulates a VCard E-Mail address.

TIdVCardEMailItem = **class**(TCollectionItem)

Description

The TIdVCardEMailItem object encapsulates a VCard owner's E-Mail address in the TIdVCard *↗TIdVCard* component.

3.1.245. TIdVCardEmbeddedObject

Encapsulates an embedded object in a VCard.

TIdVCardEmbeddedObject = **class**(TPersistent)

Description

The TIdVCardEmbeddedObject object encapsulates an embedded object in a VCard such as a graphic, sound, or public-key (for encryption).

3.1.246. TIdVCardGeog

Encapsulates geographical information about a VCard owner.

TIdVCardGeog = **class**(TPersistent)

Description

The TIdVCardGeog object encapsulates a VCard owner's geographical information such as Latitude, Longitude, and time-zone.

3.1.247. TIdVCardMailingLabelItem

Encapsulates a VCard mailing label.

TIdVCardMailingLabelItem = **class**(TCollectionItem)

Description

The TIdVCardMailingLabelItem object encapsulates a mailing label for the VCard owner.

3.1.248. TIdVCardMailingLabels

Encapsulates a VCard mailing label list.

TIdVCardMailingLabels = **class**(TOwnedCollection)

Description

The TIdVCardMailingLabels collection contains TIdVCardMailingLabelItem *↗TIdVCardMailingLabelItem* objects encapsulating mailing labels for a VCard owner.

3.1.249. TIdVCardName

Encapsulates a VCard name.

TIdVCardName = **class**(TPersistent)

Description

The TIdVCardName object encapsulates a VCard name.

3.1.250. TIdVCardTelephones

Encapsulates a VCard telephone number collection.

TIdVCardTelephones = **class**(TOwnedCollection)

Description

The TIdVCardTelephones collection contains TIdCardPhoneNumber *↗TIdCardPhoneNumber*

objects encapsulating telephone numbers for a VCard owner.

### 3.1.251. TIdWhois

Implements a Whois or Nickname client.

```
TIdWhois = class(TIdTCPClient)
```

**Description**

TIdWhois implements the Nickname or Whois protocol (RFC 954) as a client. Whois is a simple database query system that is commonly used for consulting domain registration records although it can be used for other simple directory services.

### 3.1.252. TIdWholsServer

Implements a Whois or Nickname server.

```
TIdWhoIsServer = class(TIdTCPserver)
```

**Description**

TIdWholsServer helps developers write implementations the Nickname or Whois protocol (RFC 954) as a server. Whois is a simple database query system that is commonly used for consulting domain registration records although it can be used for other simple directory services.

### 3.1.253. TIdX509

Implements X.509 certificates for SSL transports.

```
TIdX509 = class(TObject)
```

**Description**

TIdX509 is a class that implements support for X.509 Certificates as required for use with the Indy implementation of the OpenSSL Secure Socket ~~SSL~~ Socket Layer transport.

### 3.1.254. TIdX509Name

Represents an X.509 Certificate Name.

```
TIdX509Name = class(TObject)
```

**Description**

TIdX509Name is a class used to represent an X.509 Name as required for X.509 Certificate usage.

### 3.1.255. TIdXXDecoder

Implements a decoder for the XXEncode encoding scheme.

```
TIdXXDecoder = class(TIdUUDecoder)
```

**Description**

TIdXXDecoder is a TIdUUDecoder ~~is~~ *TIdUUDecoder* descendant that decodes the 7-bit textual representation of a XXEncoded message into the original binary data.

XXEncode encoding represents 24-bit groups of input as output strings of 4 encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating 3 8bit input groups. These 24 bits are then treated as 4 concatenated 6-bit groups, each of which is translated into a single digit in the XXEncode alphabet.

TIdXXDecoder reverses the XXEncoding process by converting the 4-byte US-ASCII representaiton of the data into it's 3-byte binary equivalent.

### 3.1.256. TIdXXEncoder

Implements an encoder for the XXEncode encoding scheme.

```
TIdXXEncoder = class(TIdUUEncoder)
```

### Description

TidXXEncoder is a TidUUEncoder *descendant* that encodes binary data into a 7-bit textual representation, and is very similar to the UUEncode encoding scheme.

XXEncode, like Base64, is an encoding mechanism designed to represent arbitrary sequences of byte data in a form that need not be humanly readable. XXEncode was designed as a replacement for earlier version of UUEncode, which contained encoding error, but is little since UUEncode has been enhanced.

XXEncode encoding represents 24-bit groups of input as output strings of 4 encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating 3 8bit input groups. These 24 bits are then treated as 4 concatenated 6-bit groups, each of which is translated into a single digit in the XXEncode alphabet.

XXEncode differs from Base64 and UUEncode in that it utilizes a different coding table, or alphabet, to represent the encoded output values. The XXEncode Alphabet can be represented by the following encodings and values:

Value	Encoding	Value	Encoding
0	+	32	U
1	-	33	V
2	0	34	W
3	1	35	X
4	2	36	Y
5	3	37	Z
6	4	38	a
7	5	39	b
8	6	40	c
9	7	41	d
10	8	42	e
11	9	43	f
12	A	44	g
13	B	45	h
14	C	46	i
15	D	47	j
16	E	48	k
17	F	49	l
18	G	50	m
19	H	51	n
20	I	52	o
21	J	53	p
22	K	54	q
23	L	55	r

24	M	56	s
25	N	57	t
26	O	58	u
27	P	59	v
28	Q	60	w
29	R	61	x
30	S	62	y
31	T	63	z

When encoding a bit stream using XXEncode encoding, the stream must be presumed to be ordered with the most-significant-bit first. That is, the first bit in the stream will be the high-order bit in the first 8bit byte, and the eighth bit will be the low-order bit in the first 8bit byte, etc. XXEncode also adds output headers, Unix privilege levels, and line length encoding for each line in the XXEncode output buffer.

### 3.1.257. TlpProperty

Provides persistent storage and manipulation for network address and mask values.

```
TipProperty = class(TPersistent)
```

### Description

TIpProperty is a **TPersistent** descendant that provides storage variables and manipulation methods for the components of a network address or network mask. TIpProperty is used by TIdNetworkCalculator *TIdNetworkCalculator* to represent the NetworkAddress and NetworkMask properties.

Use Byte1, Byte2, Byte3, and Byte4 to access the individual **Byte** values of the 32-bit IP address.

Use `AsDoubleWord` to access the **LongWord** value of the IP address.

Use `AsBinaryString` to access the **String** representation of individual bits of the IP address.

For example: "11111111111111111111111100000000" is the binary string representation of the network mask **255.255.255.0**.

Use `AsString` to access the **String** representation of the IP address. For example: `"255.255.255.0"` is the string representation of a network mask with the "dotted-decimal" value `255.255.255.0`.

Use `SetAll` to simultaneously update the 4 **Byte** values in the IP address.

Use `ByteArray` to access the **Byte** values of the IP address and indicate if the byte has a zero



or non-zero value.  
Use the OnChange event handler to perform application-specific processing when the value of the TIpProperty has been modified.

3.1.258. TLogger

Implements logging for Indy Tunnel components.

```
TLogger = class(TObject)
```

Description

TLogger is a class used to provide support for message and status logging in TIdTunnelMaster *↗TIdTunnelMaster* and TIdTunnelSalve components. TLogger allows messages to be written to a file, and protects resources to prevent simultaneous access by multiple threads in the Indy Tunnel components.

3.1.259. TMInfoRecord

Represents a mail information resource record.

```
TMInfoRecord = class(TIdDNSResourceItem)
```

Description

TMInfoRecord is a TIdDNSResourceItem *↗TIdDNSResourceItem* that represents a mail information resource record.

3.1.260. TMRecord

Represents a mail resource record.

```
TMRecord = class(TIdDNSResourceItem)
```

Description

TMInfoRecord *↗TMInfoRecord* is a TIdDNSResourceItem *↗TIdDNSResourceItem* that represents a mail resource record. TMInfoRecord *↗TMInfoRecord* can be associated with a

simple mailbox, they are usually used with a mailing list.

3.1.261. TMXRecord

Represents a mail exchange resource record.

```
TMXRecord = class(TIdDNSResourceItem)
```

Description

TMXRecord is a TIdDNSResourceItem *↗TIdDNSResourceItem* that represents a mail exchange resource record.

3.1.262. TNameRecord

Represents a domain name resource record.

```
TNameRecord = class(TIdDNSResourceItem)
```

Description

TMInfoRecord *↗TMInfoRecord* is a TIdDNSResourceItem *↗TIdDNSResourceItem* that represents a domain name resource record.

3.1.263. TPTRRecord

Represents a domain name pointer resource record.

```
TPTRRecord = class(TIdDNSResourceItem)
```

Description

TPTRRecord is a TIdDNSResourceItem *↗TIdDNSResourceItem* that represents a domain name pointer resource record.

### 3.1.264. TQuestionItem

represents DNS questions in a DNS query.

```
TQuestionItem = class(TCollectionItem)
```

**Description**

TQuestionItem is a TCollectionItem descendant that represents DNS questions in a DNS query.

### 3.1.265. TReceiver

Implements encapsulated tunnel message receipt and processing.

```
TReceiver = class(TObject)
```

**Description**

TReceiver is a class that implements for receipt and processing of encapsulated tunnel messages for Indy Tunnel components. TIdTunnelMaster *↗* *TIdTunnelMaster* uses TReceiver to handle message types and data from tunnel client connections.

### 3.1.266. TSender

Implements encapsulated tunnel message construction for transmission.

```
TSender = class(TObject)
```

**Description**

TSender is a class that implements support for construction of encapsulated tunnel messages for Indy Tunnel components. TIdTunnelMaster *↗* *TIdTunnelMaster* uses TSender to build message types, headers, and data for tunnel client connections.

### 3.1.267. TSlaveData

Represents a service thread for a tunnel server.

```
TSlaveData = class(TObject)
```

**Description**

TSlaveData is a TObject descendant used to represent sender, receiver, and user-defined information for a tunnel server service thread. TSlaveData instances are created when a tunnel server accepts a new connection in TIdTunnelMaster.DoConnect, and are stored in the TIdPeerThread.Data property for the new connection.

UserData can be used to stored any TObject descendant that has significance to the service thread for the tunnel server connection.

### 3.1.268. TSlaveThread

Implements a listening thread for communication with the tunnel master server.

```
TSlaveThread = class(TIdThread)
```

**Description**

TSlaveThread is a TIdThread *↗* *TIdThread* descendant that implements a listening thread for the TIdTunnelSlave *↗* *TIdTunnelSlave* server.

TSlaveThread overrides the BeforeRun, AfterRun, Run, and Execute virtual methods, inherited from TIdPeerThread *↗* *TIdPeerThread*, that provide the Indy thread execution framework.

TSlaveThread keeps a reference, in SlaveParent, to the TIdTunnelSlave *↗* *TIdTunnelSlave* that owns the listening thread. SlaveParent provides the server context for accessing properties and method of the TIdTunnelSlave *↗* *TIdTunnelSlave* from the listening thread.

TSlaveThread keeps a reference to the TIdTCPClient *↗* *TIdTCPClient* connection that represents the tunnelled link to the TIdTunnelServer. The tunnel connection is established in the constructor for TSlaveThread, closed in Run, and Disconnected in the destructor.

When the TSlaveThread is executed in Run, it insures that the tunnel connection to the TIdTunnelMaster *↗* *TIdTunnelMaster* is active and ready to accept read and write operations. If an exception occurs, the connection is closed and the thread will terminate. If the exception is not EldSocketError *↗* *EldSocketError* or EldSocketClosed, the exception will be raised for the application.

TSlaveThread arbitrates data transformation for messages from the tunnel master server, as

well as session connection and disconnection for client messages.

3.1.269. TSOARecord

Represents a Start of Authority resource record from a DNS Query.

```
TSOARecord = class(TIdDNSResourceItem)
```

Description

TSOA is a TIdDNSResourceItem *↗TIdDNSResourceItem* descendant that represents a Start of Authority resource record from a DNS query response.

3.1.270. TSocksInfo

Represents Socks support information.

```
TSocksInfo = class(TPersistent)
```

Description

TSocksInfo provides an intuitive interface for Socks support in an Indy TIdTCPClient *↗TIdTCPClient*.

3.1.271. TTelnetData

Encapsulates storage for information used by a Telnet connection.

```
TTelnetData = class(TObject)
```

Description

TTelnetData is TObject descendant that provides storage for information used by a Telnet connection. TTelnetData is used by the OnAuthentication event handler to access user name and password information for the Telnet connection.

3.1.272. TWKSRecord

Represents a Well Known Service Resource Record.

```
TWKSRecord = class(TIdDNSResourceItem)
```

Description

TWKSRecord is a TIdDNSResourceItem *↗TIdDNSResourceItem* descendant used to provide access to the IP Address, Protocol, and Bits for WKS Resource Record from a DNS query response.

3.2. Records / Structs

3.2.1. \_TRANSMIT\_FILE\_BUFFERS

```
_TRANSMIT_FILE_BUFFERS = record
  Head: Pointer;
  HeadLength: DWORD;
  Tail: Pointer;
  TailLength: DWORD;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.2. CardinalRec

Represents a Cardinal value.

```
CardinalRec = record
  case Byte of
    1: (aCardinal: Cardinal);
    2: (Words: HI LoWords);
  end;
```

**Description**

CardinalRec is a record type used to provide differing representations of a Cardinal value. Use Cardinal to access the native cardinal value, or HiLoWords to access the Cardinal as Word values.

3.2.3. HiLoBytes

Represents data with High and Low byte values.

```
HiLoBytes = record
  HiByte: Byte;
  LoByte: Byte;
end;
```

**Description**

HiLoBytes is a record type used to represent a value accessed by the HiByte and LoByte members.

3.2.4. HiLoWords

Represents data with High and Low Word values.

```
HiLoWords = record
  HiWord: Word;
  LowWord: Word;
end;
```

**Description**

HiLoWords is a Record type used to access data using the HiWord and LoWord Word values.

3.2.5. hostent

```
hostent = record
  h_name: PChar;
  h_aliases: ^PChar;
```

```
h_addrtype: Smallint;
h_length: Smallint;
case Byte of
  0: (h_addr_list: ^PChar);
  1: (h_addr: ^PChar);
end;
```

**Description**

The text for this record has been generated automatically. This means that it is not documented.

3.2.6. in\_addr

```
in_addr = record
  case integer of
    0: (S_un_b: SunB);
    1: (S_un_w: SunW);
    2: (S_addr: u_long u_long);
  end;
```

**Description**

The text for this record has been generated automatically. This means that it is not documented.

3.2.7. linger

```
linger = record
  l_onoff: u_short u_short;
  l_linger: u_short u_short;
end;
```

**Description**

The text for this record has been generated automatically. This means that it is not documented.

3.2.8. netent

```
netent = record
  n_name: PChar;
  n_aliases: ^PChar;
  n_addrtype: Smallint;
  n_net: u_long u_long;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.9. protoent

```
protoent = record
  p_name: PChar;
  p_aliases: ^Pchar;
  p_proto: Smallint;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.10. servent

```
servent = record
  s_name: PChar;
  s_aliases: ^PChar;
  s_port: Word;
  s_proto: PChar;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.11. sockaddr\_in

```
sockaddr_in = record
  case Integer of
    0: (sin_family: u_short u_short;
        sin_port: u_short u_short;
        sin_addr: TInAddr TInAddr;
        sin_zero: array[0..7] of Char;
        sa_family: u_short u_short;
        sa_data: array[0..13] of Char) end;
    WSAData=recordwVersion: Word;
    wHighVersion: Word;
    szDescription: array[0..WSADESCRIPTION_LEN] of Char;
    szSystemStatus: array[0..WSASYS_STATUS_LEN] of Char;
    iMaxSockets: Word;
    iMaxUdpDg: Word;
    lpVendorInfo: PChar;);
  end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.12. sockproto

```
sockproto = record
  sp_family: u_short u_short;
  sp_protocol: u_short u_short;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.13. SunB

```
SunB = packed record
```

```
s_b1: u_char u_char;  
s_b2: u_char u_char;  
s_b3: u_char u_char;  
s_b4: u_char u_char;  
end;
```

**Description**

The text for this record has been generated automatically. This means that it is not documented.

3.2.14. SunW

```
SunW = packed record  
  s_w1: u_short u_short;  
  s_w2: u_short u_short;  
end;
```

**Description**

The text for this record has been generated automatically. This means that it is not documented.

3.2.15. TByteArray

Represents byte array data used in SSL communications.

```
TByteArray = record  
  Length: Integer;  
  Data: PChar;  
end;
```

**Description**

TByteArray is a record type that represents byte array data used in SSL communications. Length is an Integer member that indicates the number of bytes in Data. Data is a PChar member that contains the byte values for the TByteArray.

3.2.16. TEVP\_MD

Represents an X.509 certificate fingerprint.

```
TEVP_MD = record  
  Length: Integer;  
  MD: Array[0..OPENSSL_EVP_MAX_MD_SIZE-1] of Char;  
end;
```

**Description**

TEVP\_MD is a record type used to represent X.509 certificate fingerprint information. Length is an Integer member that contains the length of the fingerprint data. MD is an Array of Char that contains the byte values for the fingerprint.

3.2.17. TFDSet

```
TFDSet = record  
  fd_count: u_int u_int;  
  fd_array: array[0..FD_SETSIZE FD_SETSIZE - 1] of TSocket TSocket;  
end;
```

**Description**

The text for this record has been generated automatically. This means that it is not documented.

3.2.18. THInfo

Represents a Host resource record.

```
THInfo = record  
  CPUStr: ShortString;  
  OsStr: ShortString;  
end;
```

**Description**

THInfo is a Record type that represents a Host resource record from a DNS response packet. CPUStr specifies the CPU identifier for the host.

OsStr specifies the host's Operating System.

3.2.19. TIdArpHdr

```
TIdArpHdr = packed record
  arp_hrd: word;
  arp_pro: word;
  arp_hln: byte;
  arp_pln: byte;
  arp_op: word;
  arp_sha: TIdEtherAddr;
  arp_spa: TIdInAddr;
  arp_tha: TIdEtherAddr;
  arp_tpa: TIdInAddr;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.20. TIdCardinalBytes

Represents values for TId3To4Coder *↗TId3To4Coder* 4-byte to 3-byte character conversions.

```
TIdCardinalBytes = record
  case integer of
    0: (Byte1: Byte;
        Byte2: Byte;
        Byte3: Byte;
        Byte4: Byte;);
    1: (Whole: Cardinal;);
end;
```

Description

TIdCardinalBytes is a record type that represents the bytes values for TId3To4Coder *↗TId3To4Coder* 4-byte to 3-byte character conversions. Byte1, Byte1, Byte3, and Byte4 are Byte members that represent individual 8-bit values for the encoding.

Whole is a Cardinal member that represents the 32-bit value for the encoding value.

3.2.21. TIdDnsHdr

```
TIdDnsHdr = packed record
  dns_id: word;
  dns_flags: word;
  dns_num_q: word;
  dns_num_answ_rr: word;
  dns_num_auth_rr: word;
  dns_num_addi_rr: word;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.22. TIdEtherAddr

```
TIdEtherAddr = packed record
  ether_addr_octet: array [0..Id_ETHER_ADDR_LEN-1] of byte;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.23. TIdEthernetHdr

```
TIdEthernetHdr = packed record
  ether_dhost: TIdEtherAddr;
  ether_shost: TIdEtherAddr;
  ether_type: word;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.24. TIdHeader

Represents encapsulated tunnel headers and data.

```
TIdHeader = record
  CRC16: Word;
  MsgType: Word;
  MsgLen: Word;
  UserId: Word;
  Port: Word;
  IpAddr: TIdInAddr;
end;
```

Description

TIdHeader is a record type that represents encapsulated tunnel headers and data for Indy Tunnel components.  
CRC16 is a Word member that contains the calculated CRC16 checksum value for the contents of the encapsulated tunnel message.  
MsgType is a Word member that contains a message type constant value representing a handling instruction to the Tunnel component.  
MsgLen is a Word member that contains the length of the encapsulated tunnel message.  
UserId is a Word member that represents the user identifier for the client connection to the Tunnel server.  
Port is a Word member that presents the for the client connection to the Tunnel component.  
IpAddr is a TIdInAddr member that represent the structured IP address for the client connection to the Tunnel component.

3.2.25. TIdIcmpEcho

```
TIdIcmpEcho = packed record
  id: word;
  seq: word;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.26. TIdIcmpFrag

```
TIdIcmpFrag = packed record
  pad: word;
  mtu: word;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.27. TIdIcmpHdr

```
TIdIcmpHdr = packed record
  icmp_type: byte;
  icmp_code: byte;
  icmp_sum: word;
  icmp_hun: packed record case integer of 0: (echo: TIdIcmpEcho); 1:
(gateway: TIdInAddr); 2: (frag: TIdIcmpFrag);;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.28. TIdIcmpTs

```
TIdIcmpTs = packed record
  otime: TIdNetTime ⚡TIdNetTime;
  rtime: TIdNetTime ⚡TIdNetTime;
  ttime: TIdNetTime ⚡TIdNetTime;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.



3.2.29. TIdIgmPHdr

```
TIdIgmPHdr = packed record
  igmp_type: byte;
  igmp_code: byte;
  igmp_sum: word;
  igmp_group: TIdInAddr;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.30. TIdInAddr

Represents an IP address for protocol stack functions.

```
TIdInAddr = record
  case integer of
    0: (S_un_b: TIdSunB);
    1: (S_un_w: TIdSunW);
    2: (S_addr: longword);
  end;
```

Description

TIdInAddr is a variant record type used to represent an IP address in a form required by low-level socket functions. TIdInAddr provides a flexible method for accessing the differing representations of the 32-bit IP address.

S\_un\_b is a TIdSunB member that represents the IP address as 4 Byte values.

S\_un\_w is a TIdSunW member that represents the IP address as 2 Word values.

S\_addr is a LongWord member that represents the 32-bit IP address.

TIdInAddr is used internally in Indy.

3.2.31. TIdIPHdr

```
TIdIPHdr = packed record
  ip_verlen: byte;
```

```
ip_tos: byte;
ip_len: word;
ip_id: word;
ip_off: word;
ip_ttl: byte;
ip_p: byte;
ip_sum: word;
ip_src: TIdInAddr;
ip_dst: TIdInAddr;
ip_options: longword;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.32. TIdIPOptions

```
TIdIPOptions = packed record
  ipopt_list: array [0..Id_MAX_IPOPTLEN-1] of char;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.33. TIdRIPHdr

```
TIdRIPHdr = packed record
  rip_cmd: byte;
  rip_ver: byte;
  rip_rd: word;
  rip_af: word;
  rip_rt: word;
  rip_addr: longword;
  rip_mask: longword;
  rip_next_hop: longword;
  rip_metric: longword;
end;
```

**Description**

The text for this record has been generated automatically. This means that it is not documented.

3.2.34. TIdSocksRequest

Contains data sent to the Socks proxy during a connection request.

```
TIdSocksRequest = record
  Version: Byte;
  OpCode: Byte;
  Port: Word;
  IpAddr: TIdInAddr;
  UserId: String[255];
end;
```

**Description**

TIdSocksRequest is a record type that represents the data sent to the Socks proxy during a TIdTCPClient *↗TIdTCPClient* connection to a Socks proxy server.  
Version is a Byte member that represents the Socks protocol version number required for the connection.  
OpCode is a Byte member that identifies the action to be performed on connection.  
Port is a Word member that identifies the Port number for the client connection.  
IpAddr is a TIdInAddr member that represents the structured IP address used for the client connection.  
UserId is Short String member that identifies the user name used for authentication of the client connection.  
TIdSocksRequest is used internally by TIdTCPClient *↗TIdTCPClient*.

3.2.35. TIdSocksResponse

Contains response data received from the Socks proxy connection.

```
TIdSocksResponse = record
  Version: Byte;
```

```
OpCode: Byte;
Port: Word;
IpAddr: TIdInAddr;
end;
```

**Description**

TIdSocksResponse is a record type that represents a reply from a Socks proxy server.  
Version is a Byte member that represents the Socks protocol version number used for the connection.  
OpCode is a Byte member that identifies the action to performed using the connection.  
Port is a Word member that identifies the Port number used for the client connection.  
IpAddr is a TIdInAddr member that represents the structured IP address used for the client connection.  
TIdSocksResponse is used internally in TIdTCPClient *↗TIdTCPClient*.

3.2.36. TIdSunB

```
TIdSunB = packed record
  s_b1: byte;
  s_b2: byte;
  s_b3: byte;
  s_b4: byte;
end;
```

**Description**

The text for this record has been generated automatically. This means that it is not documented.

3.2.37. TIdSunW

```
TIdSunW = packed record
  s_w1: word;
  s_w2: word;
end;
```

**Description**

The text for this record has been generated automatically. This means that it is not documented.

3.2.38. TIdTcpHdr

```
TIdTcpHdr = packed record
  tcp_sport: word;
  tcp_dport: word;
  tcp_seq: longword;
  tcp_ack: longword;
  tcp_x2off: byte;
  tcp_flags: byte;
  tcp_win: word;
  tcp_sum: word;
  tcp_urp: word;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.39. TIdTcpOptions

```
TIdTcpOptions = packed record
  tcptopt_list: array [0..Id_MAX_IPOPTLEN-1] of byte;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.40. TIdUdpHdr

```
TIdUdpHdr = packed record
  udp_sport: word;
  udp_dport: word;
  udp_ulen: word;
  udp_sum: word;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.41. timeval

```
timeval = record
  tv_sec: Longint;
  tv_usec: Longint;
end;
```

Description

The text for this record has been generated automatically. This means that it is not documented.

3.2.42. TIMFCoderUsage

Represents the current code and message part.

```
TIMFCoderUsage = record
  InUse: Boolean;
  BodyCoder: TIdCoder ↗TIdCoder;
  MP: TIdMessagePart ↗TIdMessagePart;
end;
```

Description

TIMFCoderUsage is a Record structure used to represent the decoder and the message part for the current TIdMessage *↗TIdMessage*. TIMFCoderUsage is used from the TIdMessage.ReceiveBody method.

3.2.43. TIpStruct

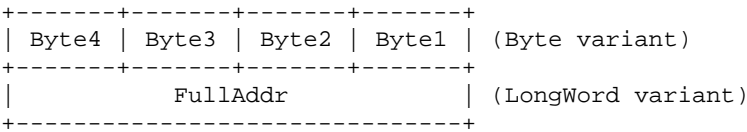
Provides storage for the components of an IP address or network mask.

```
TIpStruct = record
  case integer of
```

```
0: (Byte1: byte;  
    Byte2: byte;  
    Byte3: byte;  
    Byte4: byte;);  
1: (FullAddr: Longword;);  
end;
```

**Description**

TIpStruct is a variant **record** used to store IP address values. TIpStruct has the following format:



TIpStruct provides access to the value of the IP address as a sequence of **Byte** values using Byte1, Byte2, Byte3, and Byte4. TIpStruct also provides access to the value of the IP address as a **LongWord** value using FullAddr.

TIpStruct is used by TIpProperty *≠* *TIpProperty* and the functions IP and StrToIP to store IP addresses.

3.2.44. TLinger

Represents socket Linger options.

```
TLinger = record  
    l_onoff: Word;  
    l_linger: Word;  
end;
```

**Description**

TLinger is a record type used to represent the TCP Linger options used internally by Indy.

3.2.45. TLr

NTPGram LongInt storage structure.

```
TLr = packed record  
    L1: byte;  
    L2: byte;  
    L3: byte;  
    L4: byte;  
end;
```

**Description**

The TLr structure is an internal **packed record** structure used to handle the byte order of LongInt values received in a TNTPGram datagram. TLr is used to flip the order of bytes when TTIdSNTP.DateTime updates the value of Originate, Destination, Transmit, and Receive timestamps.

L1 is used to store the value of byte 1 of 4 bytes in the TLr structure. L2 is used to store the value of byte 2 of 4 bytes in the TLr structure. L3 is used to store the value of byte 3 of 4 bytes in the TLr structure. L4 is used to store the value of byte 4 of 4 bytes in the TLr structure.

3.2.46. TMInfo

Represents a mail information resource record.

```
TMInfo = record  
    EMailBox: ShortString;  
    RMailBox: ShortString;  
end;
```

**Description**

TMInfo is a Record type that represents a mail information resource record.

EMailBox is the E-Mail address which should receive error messages for a mailing list. If this specifies the root address for a domain, the message should be sent to the original sender of the message.

RMailBox specifies the E-Mail address of a person responsible for a mailing list.

3.2.47. TMX

Represents a mail exchange resource record.

```
TMX = record
  Exchange: ShortString;
  Preference: Word;
end;
```

Description

TMX is a Record type that represents a mail exchange resource record. Exchange is an address for a computer which will accept E-Mail for a particular domain. Preference is the priority given for the host computer. A lower priority is the computer which is preferred.

3.2.48. TNPtGram

NTP/SNTP Datagram message structure.

```
TNPtGram = packed record
  Head1: byte;
  Head2: byte;
  Head3: byte;
  Head4: byte;
  RootDelay: longint;
  RootDispersion: longint;
  RefID: longint;
  Ref1: longint;
  Ref2: longint;
  Org1: longint;
  Org2: longint;
  Rcv1: longint;
  Rcv2: longint;
  Xmit1: longint;
  Xmit2: longint;
end;
```

Description

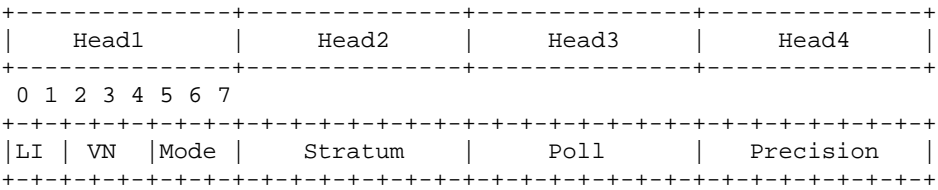
TNPtGram is the packed record structure used by TldSNTP ~~to~~ *TldSNTP* for processing SNTP requests and responses. TNPtGram is the data transmitted in UDP packets, and represents the NTP/SNTP Datagram message format as described in the Internet standards document:

- RFC 2030 - Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI

SNTP Version 4 includes certain optional extensions to the basic Version 3 model. TNPtGram does not implement the optional extensions for either Key Identifier or Message Digest portions of the NTP/SNTP message format.

Head1 - Leap Indicator, Version Number, and Mode

Head1 represents byte 1 of 4 bytes in the TNPtGram header. Head1 is utilized to store the values of Leap Indicator, Version Number, and Mode fields according to the following diagram:



Bits	Field
====	=====
0-1	Leap Indicator
2-4	Version Number
5-7	Mode

**Leap Indicator (LI)** - Two-bit code warning of an impending leap second to be inserted/deleted in the last minute of the current day, with bit 0 and bit 1, respectively, coded as follows:

- 0 - no warning
- 1 - last minute has 61 seconds
- 2 - last minute has 59 seconds)
- 3 - alarm condition (clock not synchronized)

**Version Number (VN)** - Three-bit integer indicating the NTP/SNTP version number. The

version number is 3 for Version 3 (IPv4 only) and 4 for Version 4 (IPv4, IPv6 and OSI). If necessary to distinguish between IPv4, IPv6 and OSI, the encapsulating context must be inspected.

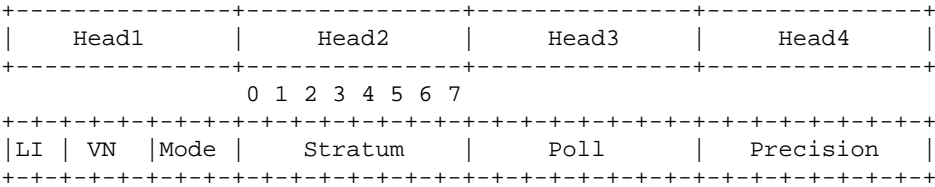
**Mode** - Three-bit integer indicating the mode, with values defined as follows:

- 0 - reserved
- 1 - symmetric active
- 2 - symmetric passive
- 3 - client
- 4 - server
- 5 - broadcast
- 6 - reserved for NTP control message
- 7 - reserved for private use

**Note:** In unicast and anycast modes, the client sets this field to 3 (client) in the request and the server sets it to 4 (server) in the reply. In multicast mode, the server sets this field to 5 (broadcast).

**Head2 - Statum level of the local clock**

Head2 represents byte 2 of 4 bytes in the TNTPGram header. Head2 is utilized to store the values of Stratum, as indicated in the following diagram:



Bits    Field  
====   =====  
0-7     Stratum

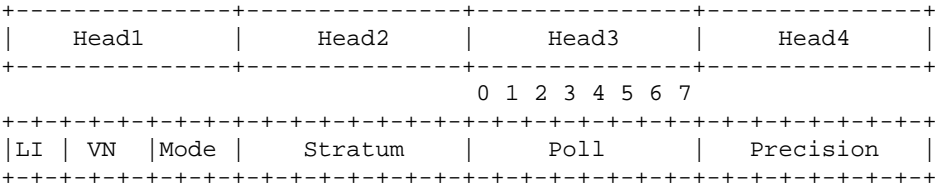
**Stratum** - Eight-bit unsigned integer indicating the stratum level of the local clock, with values defined as follows:

- 0 unspecified or unavailable
- 1 primary reference (e.g., radio clock)

- 2-15 secondary reference (via NTP or SNTP)
- 16-255 reserved

**Head3 - Polling Interval**

Head3 represents byte 3 of 4 bytes in the TNTPGram header. Head3 is utilized to store the values of Poll, as indicated in the following diagram:

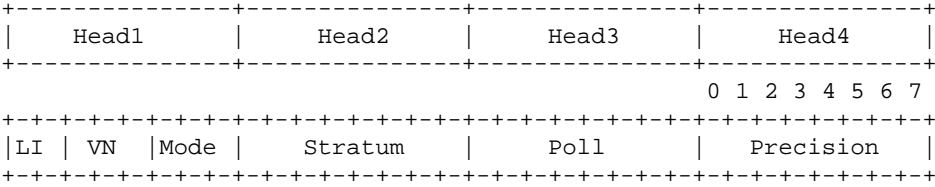


Bits    Field  
====   =====  
0-7     Poll

**Poll Interval** - Eight-bit signed integer indicating the maximum interval between successive messages, in seconds to the nearest power of two. The values that can appear in this field presently range from 4 (16 s) to 14 (16284 s); however, most applications use only the sub-range 6 (64 s) to 10 (1024 s).

**Head4 - Precision of the local clock**

Head4 represents byte 4 of 4 bytes in the TNTPGram header. Head4 is utilized to store the values of Precision, as indicated in the following diagram:



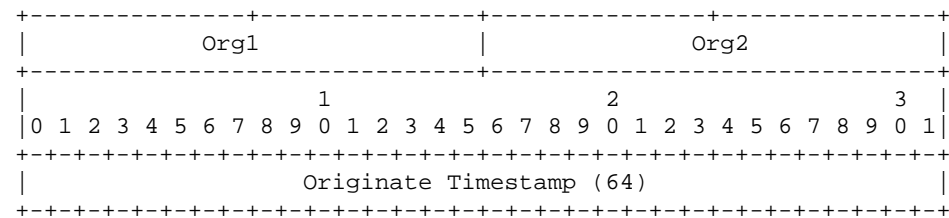
Bits    Field

```
==== =====
0-7 Precision
```

**Precision** - Eight-bit signed integer indicating the precision of the local clock, in seconds to the nearest power of two. The values that normally appear in this field range from -6 for mains-frequency clocks to -20 for microsecond clocks found in some workstations.

**Org1 - Date portion of the Originate Timestamp.**

Org1 is used to store the **LongInt** value representing the date portion of the Originate Timestamp, as indicated in the following diagram:



Bits	Field
====	=====
0-15	Org1
16-31	Org2

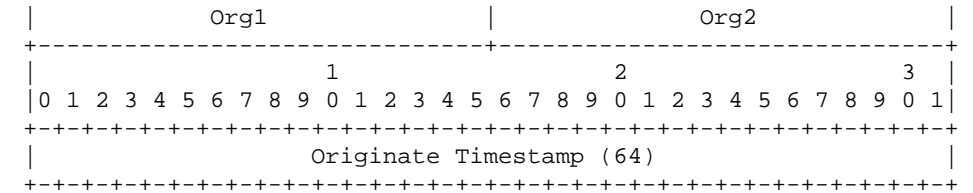
If the server is unsynchronized, all timestamp fields are set to zero.

There is some latitude on the part of most clients to forgive invalid timestamps, such as might occur when first coming up or during periods when the primary reference source is inoperative. The most important indicator of an unhealthy server is the LI field in Head1, in which a value of 3 indicates an unsynchronized condition. When this value is displayed, clients should discard the server message, regardless of the contents of other fields.

### Org2 - Time portion of the Originate Timestamp

Org2 is used to store the **LongInt** value representing the time portion of the Originate Timestamp, as indicated in the following diagram:

-----+



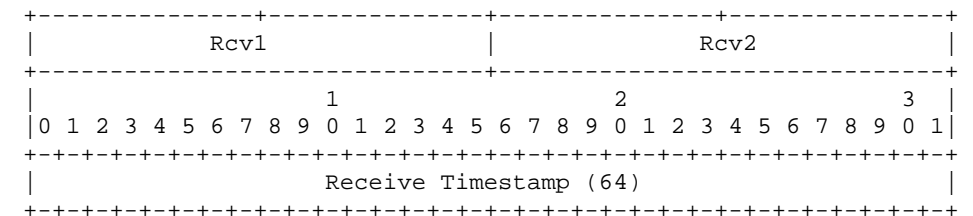
Bits	Field
0-15	Org1
16-31	Org2

If the server is unsynchronized, all timestamp fields are set to zero.

There is some latitude on the part of most clients to forgive invalid timestamps, such as might occur when first coming up or during periods when the primary reference source is inoperative. The most important indicator of an unhealthy server is the LI field in Head1, in which a value of 3 indicates an unsynchronized condition. When this value is displayed, clients should discard the server message, regardless of the contents of other fields.

### Rcv1 - Date portion of the Receive Timestamp

Rcv1 is used to store the **LongInt** value representing the date portion of the Receive Timestamp, as indicated in the following diagram:



Bits	Field
0-15	Rcv1
16-31	Rcv2

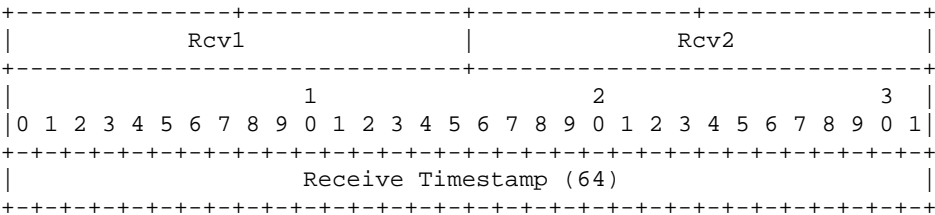
If the server is unsynchronized, all timestamp fields are set to zero. If synchronized, the Reference Timestamp is set to the time the last update was received from the radio clock or modem.

In unicast and anycast modes, the Receive Timestamp and Transmit Timestamp fields are set to the time of day when the message is sent and the Originate Timestamp field is copied unchanged from the Transmit Timestamp field of the request. It is important that this field be copied intact, as a NTP client uses it to avoid replays. In multicast mode, the Originate Timestamp and Receive Timestamp fields are set to 0 and the Transmit Timestamp field is set to the time of day when the message is sent.

There is some latitude on the part of most clients to forgive invalid timestamps, such as might occur when first coming up or during periods when the primary reference source is inoperative. The most important indicator of an unhealthy server is the LI field in Head1, in which a value of 3 indicates an unsynchronized condition. When this value is displayed, clients should discard the server message, regardless of the contents of other fields.

Rcv2 - Time portion of the Receive Timestamp

Rcv2 is used to store the **LongInt** value representing the time portion of the Receive Timestamp, as indicated in the following diagram:



Bits    Field  
====    =====  
0-15    Rcv1  
16-31   Rcv2

If the server is unsynchronized, all timestamp fields are set to zero. If synchronized, the Reference Timestamp is set to the time the last update was received from the radio clock or modem.

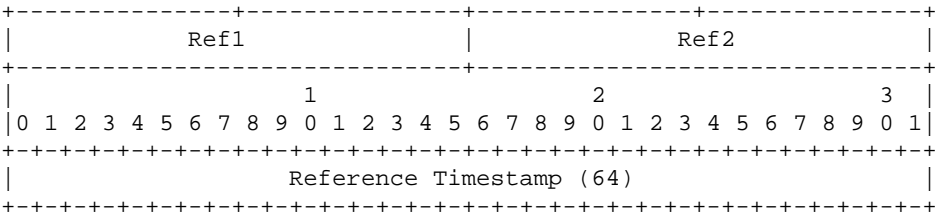
In unicast and anycast modes, the Receive Timestamp and Transmit Timestamp fields are set to the time of day when the message is sent and the Originate Timestamp field is copied unchanged from the Transmit Timestamp field of the request. It is important that this field be

copied intact, as a NTP client uses it to avoid replays. In multicast mode, the Originate Timestamp and Receive Timestamp fields are set to 0 and the Transmit Timestamp field is set to the time of day when the message is sent.

There is some latitude on the part of most clients to forgive invalid timestamps, such as might occur when first coming up or during periods when the primary reference source is inoperative. The most important indicator of an unhealthy server is the LI field in Head1, in which a value of 3 indicates an unsynchronized condition. When this value is displayed, clients should discard the server message, regardless of the contents of other fields.

Ref1 - Date portion of the Reference Timestamp

Ref1 is used to store the **LongInt** value representing the date portion of the Reference Timestamp, as indicated in the following diagram:



Bits    Field  
====    =====  
0-15    Ref1  
16-31   Ref2

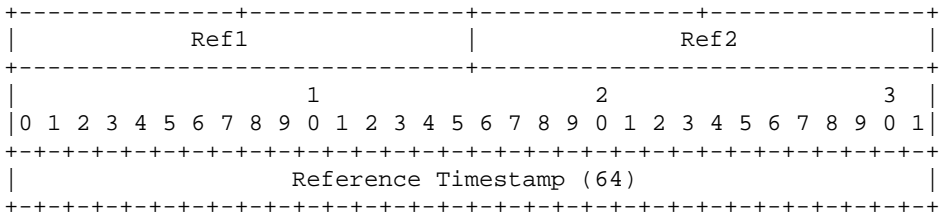
If the server is unsynchronized, all timestamp fields are set to zero. If synchronized, the Reference Timestamp is set to the time the last update was received from the radio clock or modem.

There is some latitude on the part of most clients to forgive invalid timestamps, such as might occur when first coming up or during periods when the primary reference source is inoperative. The most important indicator of an unhealthy server is the LI field in Head1, in which a value of 3 indicates an unsynchronized condition. When this value is displayed, clients should discard the server message, regardless of the contents of other fields.

Ref2 - Time portion of the Reference Timestamp

Ref2 is used to store the **LongInt** value representing the time portion of the Reference Timestamp, as indicated in the following diagram:





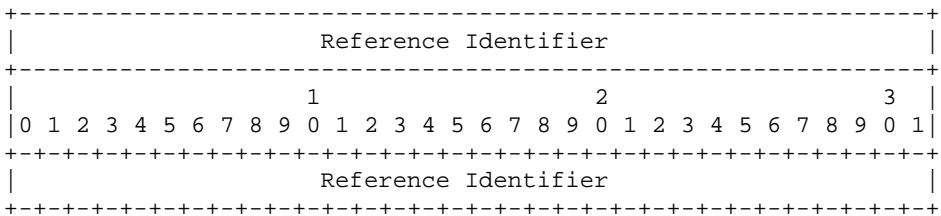
Bits    Field  
==== =====  
0-15   Ref1  
16-31 Ref2

If the server is unsynchronized, all timestamp fields are set to zero. If synchronized, the Reference Timestamp is set to the time the last update was received from the radio clock or modem.

There is some latitude on the part of most clients to forgive invalid timestamps, such as might occur when first coming up or during periods when the primary reference source is inoperative. The most important indicator of an unhealthy server is the LI field in Head1, in which a value of 3 indicates an unsynchronized condition. When this value is displayed, clients should discard the server message, regardless of the contents of other fields.

**RefID - Reference Identifier**

RefID is the **LongInt** value in TNTPGram used to identify the particular reference source, as indicated in the following diagram:



In the case of NTP Version 3 or Version 4 stratum-0 (unspecified) or stratum-1 (primary)

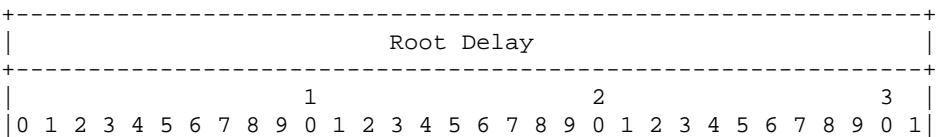
servers, this is a four-character ASCII string, left justified and zero padded to 32 bits. In NTP Version 3 secondary servers, this is the 32-bit IPv4 address of the reference source. In NTP Version 4 secondary servers, this is the low order 32 bits of the latest transmit timestamp of the reference source. In the case of NTP primary (stratum 1) servers, this field is a code identifying the external reference source according to the following list:

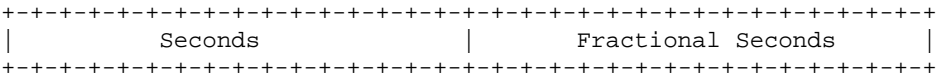
- LOCL - uncalibrated local clock used as a primary reference for a subnet without external means of synchronization
- PPS - atomic clock or other pulse-per-second source individually calibrated to national standards
- ACTS - NIST dialup modem service
- USNO - USNO modem service
- PTB - PTB (Germany) modem service
- TDF - Allouis (France) Radio 164 kHz
- DCF - Mainflingen (Germany) Radio 77.5 kHz
- MSF - Rugby (UK) Radio 60 kHz
- WWV - Ft. Collins (US) Radio 2.5, 5, 10, 15, 20 MHz
- WWVB - Boulder (US) Radio 60 kHz
- WWVH - Kauai Hawaii (US) Radio 2.5, 5, 10, 15 MHz
- CHU - Ottawa (Canada) Radio 3330, 7335, 14670 kHz
- LORC - LORAN-C radionavigation system
- OMEG - OMEGA radionavigation system
- GPS - Global Positioning Service
- GOES - Geostationary Orbit Environment Satellite

If the external reference is one of those listed, the associated code should be used. Codes for sources not listed can be contrived as appropriate.

**RootDelay - Total roundtrip delay to the time server.**

RootDelay is the **LongInt** value in TNTPGram indicating the total roundtrip delay to the primary reference source, in seconds. RootDelay stores fractional seconds after bit 15, as seen in the following diagram:

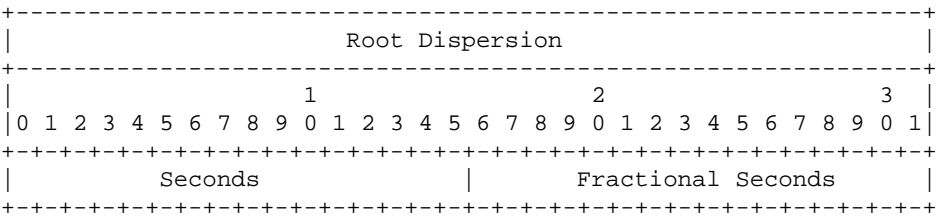




**Note:** RootDelay can have both positive and negative values, depending on the relative time and frequency offsets. The values that normally appear in this field range from negative values of a few milliseconds to positive values of several hundred milliseconds.

**RootDispersion - Nominal error for the time server**

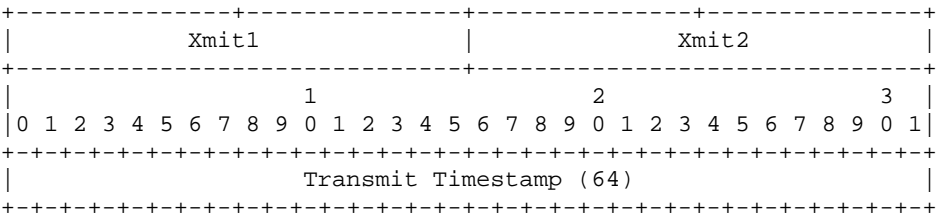
RootDispersion is the **LongInt** value in TNTPGram that indicates the nominal error relative to the primary reference source. RootDispersion stores fractional seconds after bit 15, as seen in the following diagram:



RootDispersion is measured in seconds, and values normally range from 0 to several hundred milliseconds.

**Xmit1 - Date portion of the Transmit Timestamp**

Xmit1 is used to store the **LongInt** value representing the date portion of the Transmit Timestamp, as indicated in the following diagram:



Bits	Field
====	=====
0-15	Xmit1
16-31	Xmit2

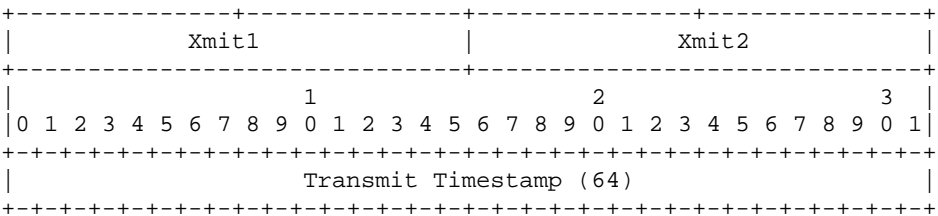
If the server is unsynchronized, all timestamp fields are set to zero. If synchronized, the Reference Timestamp is set to the time the last update was received from the radio clock or modem.

In unicast and anycast modes, the Receive Timestamp and Transmit Timestamp fields are set to the time of day when the message is sent and the Originate Timestamp field is copied unchanged from the Transmit Timestamp field of the request. It is important that this field be copied intact, as a NTP client uses it to avoid replays. In multicast mode, the Originate Timestamp and Receive Timestamp fields are set to 0 and the Transmit Timestamp field is set to the time of day when the message is sent.

There is some latitude on the part of most clients to forgive invalid timestamps, such as might occur when first coming up or during periods when the primary reference source is inoperative. The most important indicator of an unhealthy server is the LI field in Head1, in which a value of 3 indicates an unsynchronized condition. When this value is displayed, clients should discard the server message, regardless of the contents of other fields.

**Xmit2- Time portion of the Transmit Timestamp.**

Xmit2 is used to store the **LongInt** value representing the time portion of the Transmit Timestamp, as indicated in the following diagram:



Bits	Field
====	=====
0-15	Xmit1
16-31	Xmit2

If the server is unsynchronized, all timestamp fields are set to zero. If synchronized, the Reference Timestamp is set to the time the last update was received from the radio clock or modem.

In unicast and anycast modes, the Receive Timestamp and Transmit Timestamp fields are set to the time of day when the message is sent and the Originate Timestamp field is copied unchanged from the Transmit Timestamp field of the request. It is important that this field be copied intact, as a NTP client uses it to avoid replays. In multicast mode, the Originate Timestamp and Receive Timestamp fields are set to 0 and the Transmit Timestamp field is set to the time of day when the message is sent.

There is some latitude on the part of most clients to forgive invalid timestamps, such as might occur when first coming up or during periods when the primary reference source is inoperative. The most important indicator of an unhealthy server is the LI field in Head1, in which a value of 3 indicates an unsynchronized condition. When this value is displayed, clients should discard the server message, regardless of the contents of other fields.

3.2.49. TPeerInfo

Represents connection information for a TrivialFTP client connection.

```
TPeerInfo = record
  PeerIP: string;
  PeerPort: Integer;
end;
```

Description

TPeerInfo is record type represents information about a client connection to a TIdTrivialFTPServer *↗TIdTrivialFTPServer*.  
PeerIP is a String member variable that identifies the client IP address for a connection.  
PeerPort is an Integer member variable that identifies the client port number for a connection.

3.2.50. TQWord

Provides access to the component values for an Int64 integer.

```
TQWord = packed record
```

```
L: LongWord;
H: LongWord;
end;
```

Description

TQWord provides a simple interface to the two component LongWord parts of a Int64 integer. The term QWord comes from the assembler Int64 equivalent.

3.2.51. TRdata

Represents the raw data for a DNS response packet.

```
TRdata = record
  DomainName: string;
  HInfo: THInfo;
  MInfo: TMInfo;
  MX: TMx;
  SOA: TSOA;
  A: Cardinal;
  WKS: TWks;
  Data: string;
  HostAddrStr: string;
end;
```

Description

TRdata is a Record type that represents an intermediate data structure used to process and store DNS Query Response Packet data in TIdDNSResourceItem *↗TIdDNSResourceItem*.  
DomainName is a String member that represents the Domain Name returned in the response packet.  
HInfo is a THInfo member that contains Host information for the resource record.  
MInfo is a TMInfo member that contains Mailbox information for the resource record.  
MX is a TMx member that contains mail exchange information for the resource record.  
SOA is a TSOA member that contains the Start of Authority information for the resource record.  
A is a Cardinal member that contains the Address information for the resource record.  
WKS is a TWks member that contains Well Known Service information for the resource record.  
Data is a String member that represents user-defines data for the resource record.  
HostAddrStr is a String member that contains the Host address for the resource record.

3.2.52. TReplyStatus

Response to an echo request.

```
TReplyStatus = record
  BytesReceived: integer;
  FromIpAddress: string;
  MsgType: byte;
  SequenceId: word;
  MsRoundTripTime: longword;
  TimeToLive: byte;
  ReplyStatusType: TReplyStatusTypes ⚡TReplyStatusTypes;
end;
```

Description

TReplyStatus is a record structure used to store the response to an echo request in the Ping method of TIdlcmpClient *⚡TIdlcmpClient*.  
TReplyStatus contains diagnostic information received as a result of the Ping request.  
TReplyStatus is also provided as a parameter to OnReply event handler in TIdlcmpClient *⚡TIdlcmpClient*.

3.2.53. TSOA

Represents a Start of Authority resource record.

```
TSOA = record
  Expire: Cardinal;
  Minimum: Cardinal;
  MName: ShortString;
  Refresh: Cardinal;
  Retry: Cardinal;
  RName: ShortString;
  Serial: Cardinal;
end;
```

Description

TSOA is a Record type that represents a Start of Authority resources record from a DNS

response packet.  
Expire is a 32 bit time value that specifies the upper limit on the time interval that can elapse before the zone is no longer authoritative.  
Minimum is an unsigned 32 bit minimum TTL field that should be exported with any RR from this zone.  
MName is a string that represents the domain-name of the name server that was the original or primary source of data for this zone.  
Refresh is a 32 bit time interval before the zone should be refreshed.  
Retry is a 32 bit time interval that should elapse before a failed refresh should be retried.  
RName is the E-Mail address of the person responsible for this zone.  
Serial is an unsigned 32 bit version number of the original copy of the zone. Zone transfers preserve this value. This value wraps and should be compared using sequence space arithmetic.

3.2.54. TULong

Represents an unsigned long integer value.

```
TULong = packed record
  case Byte of
    0: (B1: Byte;
        B2: Byte;
        B3: Byte;
        B4: Byte;);
    1: (W1: Word;
        W2: Word;);
    2: (L1: Longint;);
    3: (C1: Cardinal;);
  end;
```

Description

TULong is a variant record type that provides access to differing representations of an unsigned long Integer value.  
B1,B2,B3,B4 are the individual Byte values for the unsigned long integer value.  
W1,W2 are the individual Word values for the unsigned long integer value.  
L1 is the LongInt value for the unsigned long integer value.  
C1 is the Cardinal value for the unsigned long integer value.

3.2.55. TWKS

Represents a Well Known Service Resource Record from a DNS Response packet.

```
TWKS = record
  Address: Cardinal;
  Bits: TWKSBits ↗TWKSBits;
  Protocol: byte;
end;
```

Description

TWKS is a Record type that represents a Well Known Service Resource Record from a DNS Response packet.  
Address is the IP address for a host as a 32 bit value.  
Bits is a variable length series of bits corresponding to IP Port numbers on a computer with bit 0 corresponding to port 0. Thus, if the computer supports Echo (Port 7), the 7th bit is set to 1 or if that bit is 0, Echo is not supported. If there are no bits for a protocol, then that protocol is not supported.

Protocol specifies the IP protocol the host supports. This can be one of these values:

- 0 Reserved
- 1 ICMP Internet Control Message
- 2 IGMP Internet Group Management
- 3 GGP Gateway-to-Gateway
- 4 Unassigned
- 5 ST Stream
- 6 TCP Transmission Control
- 7 UCL
- 8 EGP Exterior Gateway Protocol
- 9 IGP any private interior gateway
- 10 BBN-RCC-MON BBN RCC Monitoring
- 11 NVP-II Network Voice Protocol
- 12 PUP
- 13 ARGUS
- 14 EMCON
- 15 XNET Cross Net Debugger
- 16 CHAOS
- 17 UDP User Datagram
- 18 MUX Multiplexing
- 19 DCN-MEAS DCN Measurement Subsystems

- 20 HMP Host Monitoring
- 21 PRM Packet Radio Measurement
- 22 XNS-IDP XEROX NS IDP
- 23 TRUNK-1 Trunk-1
- 24 TRUNK-2 Trunk-2
- 25 LEAF-1 Leaf-1
- 26 LEAF-2 Leaf-2
- 27 RDP Reliable Data Protocol
- 28 IRTP Internet Reliable Transaction
- 29 ISO-TP4 ISO Transport Protocol Class 4
- 30 NETBLT Bulk Data Transfer Protocol
- 31 MFE-NSP MFE Network Services Protocol
- 32 MERIT-INP MERIT Internodal Protocol
- 33 SEP Sequential Exchange Protocol
- 34-60 Unassigned
- 61 any host internal protocol
- 62 CFTP
- 63 any local network
- 64 SAT-EXPAK SATNET and Backroom EXPAK
- 65 MIT-SUBNET MIT Subnet Support
- 66 RVD MIT Remote Virtual Disk Protocol
- 67 IPPC Internet Pluribus Packet Core
- 68 any distributed file system
- 69 SAT-MON SATNET Monitoring
- 70 Unassigned
- 71 IPCV Internet Packet Core Utility
- 72-75 Unassigned
- 76 BR-SAT-MON Backroom SATNET Monitoring
- 77 Unassigned
- 78 WB-MON WIDEBAND Monitoring
- 79 WB-EXPAK WIDEBAND EXPAK
- 80-254 Unassigned
- 255 Reserved

TWorkInfo

See Also Unit: IdComponent *↗IdComponent.pas*  
[[[ENDKEEPN]]]

3.2.56. TWorkInfo

Represents work information for Indy read and write operations.

```
TWorkInfo = record
  Current: Integer;
  Max: Integer;
  Level: Integer;
end;
```

Description

TWorkInfo is a Record structure used to represent information about Indy read and write operations. TIdComponent *≧*TIdComponent allocates two TWorkInfo records, one to represent read operations and one to represent write operations. TWorkInfo structures are maintain in the BeginWork and EndWork method of TIdComponent *≧*TIdComponent. Current is an Integer value that identifies the number of bytes available to the operation. Max is an Integer value that indicates the maximum number of bytes that are available to the operation. Current should never exceed the value of Max. Level is an Integer value that identifies the number of operations of that type that are pending. Level can range from 0 to the number of nested calls to BeginWork for the operation type.

3.2.57. WordRec

Represents data as HiLoBytes or Word values.

```
WordRec = record
  case byte of
    1: (TheBytes: HiLoBytes);
    2: (AWord: Word);
  end;
```

Description

WordRec is a Record type used to access data that can be represented as High and Low Byte values using HiLoBytes, or as a Word value in AWord.

3.3. Functions

3.3.1. AnsiSameText

Performs a case-insensitive comparison of two Ansi strings.

```
function AnsiSameText(const S1: string; const S2: string): Boolean;
```

Parameters

const S1: string  
The first string used in the comparison.

const S2: string  
The second string used in the comparison.

Returns

**Boolean - True** when the strings are the same.

Description

AnsiSameText is a Boolean function used to perform a case-insensitive comparison of two strings using the locale for the current user on the local computer system. AnsiSameText returns **True** if the strings are equal in lexical value, or **False** when the strings do not share the same collation order value. AnsiSameText encapsulates the platform-specific procedures and functions used to perform string comparisons. On the Windows platform, the WIN32 API function used is CompareString.

3.3.2. Base64Encode

Performs a Base64 encoding operation.

```
function Base64Encode(const s: String): String;
```

Description

Base64Encode is a String function that performs a Base64 encoding operation for the value specified in s. The return value for Base64Encode is the 7-bit ASCII representation of the

input value.

### 3.3.3. CommaSeperatedToStringList

Fills a string list with values from a comma-delimited string.

```
procedure CommaSeperatedToStringList(AList: TStrings; const Value: string);
```

**Parameters**

AList: TStrings  
List to store the values.

```
const Value: string  
String containing comma-delimited values.
```

**Description**

CommaSeperatedToStringList is a procedure used to fill the TStringList AList parameter with the values from the Value parameter.  
AList is cleared prior to adding values in the procedure.  
Value is a comma-delimited list of values to loaded in the TStringList. Value can contain items that include the comma separator by using double quotation marks to enclosed the item. For example:  
'Item One, "Item Two, with comma", Item Three'

### 3.3.4. CopyFileTo

Copies the source file to the destination file.

```
function CopyFileTo(const Source: string; const Destination: string): Boolean;
```

**Parameters**

const Source: string  
Source file name.

```
const Destination: string  
Destination file name.
```

**Returns**

Boolean - **True** if the file is copied, **False** on error.

**Description**

CopyFileTo is a function used to copy the file specified in *Source* to the file specified in *Destination*.  
CopyFileTo will return False if the file in *Destination* already exists.  
CopyFileTo encapsulates the platform-specific calls needed to perform the file copy operation. On the Windows platform, this is the Win32 API function CopyFile. On the Linux platform, CopyFileTo uses a TFileStream instance to create the destination file.

### 3.3.5. CurrentProcessId

Determines the process ID for the calling program.

```
function CurrentProcessId: TIdPID ⚡TIdPID;
```

**Returns**

TIdPID *⚡TIdPID* - Indy Process ID for the current program.

**Description**

CurrentProcessId is a TIdPID *⚡TIdPID* function used to determine the Process Identifier for the calling program.  
CurrentProcessID encapsulates the platform-specific procedures or functions needed to retrieve the current process identifier. For the Windows platform, this is the WIN32 API function GetCurrentProcessId. On the Linux platform, CurrentProcessID returns the valued from the library function getpid.

### 3.3.6. DateTimeToGmtOffSetStr

Retrieves the GMT time offset for a date/time value.

**function** DateTimeToGmtOffSetStr(ADateTime: TDateTime; SubGMT: Boolean): **string**;

**Parameters**

ADateTime: TDateTime  
Value to be converted to a GMT offset string.

SubGMT: Boolean  
Add the prefix "GMT" to the return value.

**Returns**

**String** - The offset from GMT time as a string.

**Description**

DateTimeToGmtOffSetStr is a **String** function used to return the time difference for the *ADateTime* parameter as an Internet Time difference string.  
*ADateTime* is the native Date/Time value to be converted.  
*SubGMT* indicates that the return value should contain the prefix "**GMT**" prior to the time difference string. When *SubGMT* is **True**, the prefix is added to the time difference string, otherwise the prefix is omitted.  
To convert an Internet GMT different to a TDateTime, use the GmtOffsetStrToDateTime *↗* *GmtOffsetStrToDateTime* function.

3.3.7. DateTimeToInternetStr

Converts a native date time value to an Internet timestamp.

**function** DateTimeToInternetStr(**const** Value: TDateTime): **String**;

**Parameters**

const Value: TDateTime  
The date/time value to be converted.

**Returns**

String - Representation of Value as an Internet timestamp.

**Description**

DateTimeToInternetStr is a String function used to convert the native date/time value in Value to a string representing an Internet timestamp.  
To convert a value from an Internet Time stamp to a TDateTime, use the StrInternetToDateTime *↗* *StrInternetToDateTime* function.

3.3.8. DebugOutput

Sends a string to the system debugger.

**procedure** DebugOutput(**const** AText: **string**);

**Parameters**

const AText: string  
Value to be sent to the debugger.

**Description**

DebugOutput is a procedure used to send the string in *AText* to either the active debugger for a process, or the system debugger. If there is no active debugger for the current application, DebugOutput has no effect.  
DebugOutput encapsulates the platform-specific calls needed to send the string to the debugger for the current application. For the Windows platform, the WIN32 API procedure is OutputDebugString. On the Linux platform, DebugOutput writes the message followed by CRLF to the standard error handle (StdErr).

3.3.9. Decode2022JP

Converts Shift\_JIS to ISO-2022-JP character set encoding.

**function** Decode2022JP(**const** S: **string**): **string**;

**Parameters**

const S: string  
Values to be converted.

**Returns**

String - Values in ISO-2022-JP character set.



**Description**

Decode2022JP is a String function that converts Shift\_JIS character set encodings to the ISO-2022-JP character set, in accordance with the operation described in the Internet Standards document:

- RFC 1468

3.3.10. DecodeAddress

Decodes an email address.

**procedure** DecodeAddress(EMailAddr: TIdEmailAddressItem);

**Parameters**

EMailAddr: TIdEmailAddressItem  
Email address to be decoded.

**Description**

DecodeAddress is a procedure that decodes the header representation of an email name specified in EMailAddr.

3.3.11. DecodeAddresses

Decodes an email address list.

**procedure** DecodeAddresses(AEMails: **String**; EMailAddr: TIdEmailAddressList);

**Description**

DecodeAddresses is a procedure that decodes the RFC-822 list of email address specified in AEMails into the TIdEmailAddressList specified in EMailAddr. DecodeAddresses calls DecodeAddress ↗ *DecodeAddress* to perform decoding for an individual email address.

3.3.12. DecodeHeader

Decodes header name and values using the charset for the encoding.

**function** DecodeHeader(Header: **string**): **string**;

**Parameters**

Header: string  
Raw header values to be decoded.

**Returns**

String - Decoded language specific value.

**Description**

DecodeHeader is a String function used to extract names and values from quoted-printable or base-64 encoded headers. DecodeHeader will examine the encoded header for the charset used for the content, and perform the corresponding language decoding.

3.3.13. Encode2022JP

Converts Shift\_JIS to ISO-2022-JP character set encoding.

**function** Encode2022JP(const S: **string**): **string**;

**Parameters**

const S: string  
Values to be converted.

**Returns**

String - Values in ISO-2022-JP character set.

**Description**

Encode2022JP ↗ *Decode2022JP* is a String function that converts Shift\_JIS character set encodings to the ISO-2022-JP character set, in accordance with the operation described in the Internet Standards document:

- RFC 1468

### 3.3.14. EncodeAddress

Converts a list email address to an RFC 822 email list.

```
function EncodeAddress(EmailAddr: TIdEmailAddressList
  ⚡TIdEmailAddressList; const HeaderEncoding: Char; TransferHeader:
  TTransfer ⚡TTransfer; MimeCharSet: string): string;

Parameters
EmailAddr: TIdEmailAddressList
  List of email addresses to convert.

const HeaderEncoding: Char
  Header encoding scheme.

TransferHeader: TTransfer
  Transfer encoding scheme.

MimeCharSet: string
  Charset for the encodings.

Returns
String - RFC 822 email list.

Description
EncodeAddress is a String function used to encode the TIdEmailAddressList
⚡TIdEmailAddressList email addresses specified in EmailAddr to an RFC 822-compliant list
of email addresses.
EncodeAddress calls EncodeAddressItem ⚡EncodeAddressItem to perform encoding for an
individual email address in the list.
```

### 3.3.15. EncodeAddressItem

Encoded an email address using content, transfer, and character set encodings.

```
function EncodeAddressItem(EmailAddr: TIdEmailAddrItem; const
HeaderEncoding: Char; TransferHeader: TTransfer ⚡TTransfer;
```

```
MimeCharSet: string): string;

Parameters
EmailAddr: TIdEmailAddrItem
  Email Address to be encoded.

const HeaderEncoding: Char
  Content type encoding scheme.

TransferHeader: TTransfer
  Transfer encoding scheme.

MimeCharSet: string
  Character set for the encoding.

Returns
String - Encoded email address.
```

**Description**  
EncodeAddressItem is a String function that performs encoding of the Email address specific  
in EmailAddr using the specified content type and transfer encodings for a given character  
set.  
EncodeAddressItem will perform character quoting for the special characters " and '".

### 3.3.16. EncodeHeader

Encode a header containing non-ASCII characters.

```
function EncodeHeader(const Header: string; specials: CSET ⚡CSET;
const HeaderEncoding: Char; TransferHeader: TTransfer ⚡TTransfer;
MimeCharSet: string): string;

Parameters
const Header: string
  Header to be encoded.

specials: CSET
  Set of special characters that require encoding.
```

const HeaderEncoding: Char  
Header encoding identification character.

TransferHeader: TTransfer  
transfer encoding scheme.

MimeCharSet: string  
Character set for the encoding.

**Returns**  
String - Encoded header value.

**Description**  
EncodeHeader is a String function that performs encoding for a header field when the header contains non-ASCII characters.

3.3.17. FD\_CLR

procedure FD\_CLR(Socket *Socket*: TSocket *TSocket*; var FDSet: TFDSet);

**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.3.18. FD\_ISSET

function FD\_ISSET(Socket *Socket*: TSocket *TSocket*; var FDSet: TFDSet): Boolean;

**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.3.19. FD\_SET

procedure FD\_SET(Socket *Socket*: TSocket *TSocket*; var FDSet: TFDSet);

**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.3.20. FD\_ZERO

procedure FD\_ZERO(var FDSet: TFDSet);

**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.3.21. Fetch

Parses a value from the delimited input string.

function Fetch(var AInput: string; const ADelim: string; const ADelete: Boolean): string;

**Parameters**  
var AInput: string  
Value to be parsed.

const ADelim: string = ' '  
Delimiter character(s). Default value is CHAR32 *CHAR32* (' ').

const ADelete: Boolean = true  
Remove the parsed value from the input string. Default value is True.

**Returns**  
String - Value parsed from the input string.

**Description**

Fetch is a String function that is used to parse the string in AInput up to the delimiter character(s) in ADelim.  
ADelim may contain a single character, a sequence of characters, or the value CHAR0 *↵*CHAR0 (Null string).  
Fetch can optionally delete the parsed value from AInput when ADelete is True. When ADelete is True, both the parsed value and the delimiter character(s) are removed from the input string AInput.

3.3.22. **FileSizeByName**

Retrieves the size a file.

```
function FileSizeByName(sFilename: string): cardinal;
```

**Parameters**

sFilename: string  
The file name to be examined.

**Returns**

Cardinal - Size of the file.

**Description**

FileSizeByName is a Cardinal function used to determine the size of the file specified in sFilename.  
FileSizeByName is an OS-independent mechanism used to determine the size of an Operating System file, and uses TFileStream.Size to determine the number of bytes returned by the function.

3.3.23. **FreeAndNil**

Frees an object reference and replaces the reference with Nil.

```
procedure FreeAndNil(var Obj);
```

**Parameters**

var Obj

Object reference to be freed and Nil'd.

**Description**

FreeAndNil is a procedure used to free the object reference specified by **Obj**, and sets the object reference to the **Nil** value.  
Use FreeAndNil to ensure that a variable is **Nil** after you free the object it references. Pass any variable that represents an object as the *Obj* parameter.

**Note:** Do not pass a value for *Obj* if it is not an instance of TObject or a TObject descendant. FreeAndNil is a convenience procedure provided for Delphi or C++ Builder products using a VCL version prior to VCL 5.

3.3.24. **GetMIMETypeFromFile**

Retrieves the MIME type for a given file.

```
function GetMIMETypeFromFile(AFile: TFileName): string;
```

**Parameters**

AFile: TFileName  
Name of the file to examine for the MIME type.

**Returns**

String - The MIME type for the specified file.

**Description**

GetMIMETypeFromFile is a String function used to determine the MIME type for the file specified in AFile.  
GetMIMETypeFromFile uses a TIdMIMETable to determine the registered MIME type for the specified file.

3.3.25. **GetQClassStr**

Retrieves the name for a resource class.

```
function GetQClassStr(QClass: Integer): String;
```

**Parameters**

QClass: Integer  
Resource class used to construct the resource class name.

**Returns**

String - Name for the resource class type.

**Description**

GetQClassStr is a String function that retrieves the name for the resource class specified in QClass. Get GetQClassStr uses the constant array cQClassStr *↗cQClassStr* to retrieve the name for the resource class type. When QClass contains an unknown resource class, the string representation of the Integer value is returned.

3.3.26. **GetTypeStr**

Retrieves the name for a resource type.

```
function GetTypeStr(aQType: Integer): String;
```

**Parameters**

aQType: Integer  
Resource type value.

**Returns**

String - Name for the resource type.

**Description**

GetTypeStr is a String function that retrieves the name for the resource type specified in aQType. When aQType contains an unknown resource type value, the string representation of the Integer value is returned.

3.3.27. **GetSystemLocale**

Retrieves the character set for the local computer system.

```
function GetSystemLocale: TCharSet ↗TCharSet;
```

**Returns**

TCharSet - Character set for the local computer system.

**Description**

GetSystemLocale is a TCharSet function used to retrieve the character set for the local computer system.  
For the Windows platform, GetSystemLocale uses the SysLocale values for PriLangID and SubLangID to determine the character set in use. On Windows, GetSystemLocale will return the following:

- Chinese - csBig5 or csGB2312 when SubLangID is  
SUBLANG\_CHINESE\_SIMPLIFIED
- Japanese - csIso2022jp
- Korean - csEuckKR
- Default - csIso88591

For the Linux platform, GetSystemLocale uses the GSystemLocal variable to determine the character set for the local computer system.

3.3.28. **GetTickCount**

Retrieves the number of milliseconds since the computer was started.

```
function GetTickCount: Cardinal;
```

**Returns**

Cardinal - Number of milliseconds (ticks).

**Description**

GetTickCount is a **Cardinal** function used to retrieve the number of milliseconds since the computer was started.  
GetTickCount is often used when calculating the response times for Indy components such as TIdEcho *↗TIdEcho* and TIdTime *↗TIdTime*.  
GetTickCount encapsulates the platform-specific calls necessary to provide a high resolution timer for fixed duration events. On the Windows platform, the function used is Windows.GetTickCount. On the Linux platform, GetTickCount uses the library function clock

adjusted to thousandths of CLOCKS\_PER\_SECs.

3.3.29. GmtOffsetStrToDateTime

Converts an Internet time difference string to a native date/time value.

```
function GmtOffsetStrToDateTime(S: string): TDateTime;
```

Parameters

S: string  
The Internet time difference string to be converted.

Returns

TDateTime - Native date/time representation of the time difference string.

Description

GmtOffsetStrToDateTime is a TDateTime function used to convert the Internet time difference string in S to a native TDateTime value.  
The return value for GmtOffsetStrToDateTime will contain the time difference from S encoded as a TDateTime value, and can contain either a positive or a negative number of offset hours.  
The return value for GmtOffsetStrToDateTime can be 0.0 if an error occurs during calculation of the offset value.  
GmtOffsetStrToDateTime can be used to create a value that can be added to a TDateTime value expressed in another time zone, to adjust to the local time zone.  
To convert a TDateTime to an Internet Time difference, use the DateTimeToGmtOffSetStr *↗*DateTimeToGmtOffSetStr function.

3.3.30. GMTToLocalDateTime

Converts a GMT time string to the local time.

```
function GMTToLocalDateTime(S: string): TDateTime;
```

Parameters

S: string  
GMT time string.

Returns

TDateTime - Local date time as a Delphi data type.

Description

GMTToLocalDateTime is a TDateTime function that converts an Internet date/time string expressed in GMT or UTC format to a native Delphi TDateTime data type.  
The string S contains the GMT representation of the date/time, in the following format:

```
Sun 15 Oct 2000 12:42:15 -0500
```

3.3.31. IdPorts

Provides a list of port numbers defined for services.

```
function IdPorts: TList;
```

Returns

TList - List of integer port numbers.

Description

IdPorts is a TList function used to provide a list of the integer port numbers defined for services on the protocol stack. IdPorts will initialize the application variable FIdPorts, if it has not been loaded, with the port numbers defined for the local computer system.  
IdPorts can raise an EldCorruptServicesFile *↗*EldCorruptServicesFile exception when the protocol family cannot be determined for a services entry.  
**Note:** Do not free the TList returned by IdPorts. This list is freed by the finalization code in the IdGlobal.pas *↗*IdGlobal.pas unit.

3.3.32. IdRawBuildArp

Builds a datagram using ARP headers and message formats.

```
function IdRawBuildArp(AHwAddressFormat: word; AProtocolFormat: word;  
AHwAddressLen: byte; AProtocolLen: byte; AnOpType: word; ASenderHw:  
TIdEtherAddr; ASenderPr: TIdInAddr; ATargetHw: TIdEtherAddr;
```

```
ATargetPr: TIdInAddr; const APayload; APayloadSize: integer; var
ABuffer): boolean;
```

3.3.33. IdRawBuildDns

Builds a datagram with DNS headers and message formats.

```
function IdRawBuildDns(AnId: word; AFlags: word; ANumQuestions: word;
ANumAnswerRecs: word; ANumAuthRecs: word; ANumAddRecs: word; const
APayload; APayloadSize: integer; var ABuffer): boolean;
```

3.3.34. IdRawBuildEthernet

Builds a datagram using Ethernet headers and message formats.

```
function IdRawBuildEthernet(ADest: TIdEtherAddr; ASource:
TIdEtherAddr; AType: word; const APayload; APayloadSize: integer; var
ABuffer): boolean;
```

3.3.35. IdRawBuildIcmpEcho

Builds a datagram with ICMP Echo headers and message formats.

```
function IdRawBuildIcmpEcho(AType: byte; ACode: byte; AnId: word;
ASeq: word; const APayload; APayloadSize: integer; var ABuffer):
boolean;
```

3.3.36. IdRawBuildIcmpMask

Builds a datagram using ICMP headers and ICMP Mask message formats.

```
function IdRawBuildIcmpMask(AType: byte; ACode: byte; AnId: word;
ASeq: word; AMask: longword; const APayload; APayloadSize: integer;
var ABuffer): boolean;
```

3.3.37. IdRawBuildIcmpRedirect

Builds a datagram using ICMP headers and ICMP Gateway message formats.

```
function IdRawBuildIcmpRedirect(AType: byte; ACode: byte; AGateway:
TIdInAddr; AnOrigLen: word; AnOrigTos: byte; AnOrigId: word;
AnOrigFrag: word; AnOrigTtl: byte; AnOrigProtocol: byte; AnOrigSource:
TIdInAddr; AnOrigDest: TIdInAddr; const AnOrigPayload; APayloadSize:
integer; var ABuffer): boolean;
```

3.3.38. IdRawBuildIcmpTimeExceed

Builds a datagram using ICMP headers and ICMP Tlme Exceeded message formats.

```
function IdRawBuildIcmpTimeExceed(AType: byte; ACode: byte; AnOrigLen:
word; AnOrigTos: byte; AnOrigId: word; AnOrigFrag: word; AnOrigTtl:
byte; AnOrigProtocol: byte; AnOrigSource: TIdInAddr; AnOrigDest:
TIdInAddr; const AnOrigPayload; APayloadSize: integer; var ABuffer):
boolean;
```

3.3.39. IdRawBuildIcmpTimestamp

Builds a datagram using ICMP headers and Timestamp message formats.

```
function IdRawBuildIcmpTimestamp(AType: byte; ACode: byte; AnId: word;
ASeq: word; AnOtime: TIdNetTime ⚡TIdNetTime; AnRtime: TIdNetTime
⚡TIdNetTime; ATtime: TIdNetTime ⚡TIdNetTime; const APayload;
APayloadSize: integer; var ABuffer): boolean;
```

3.3.40. IdRawBuildIcmpUnreach

Builds a datagram using ICMP headers and ICMP Unreachable message formats.

```
function IdRawBuildIcmpUnreach(AType: byte; ACode: byte; AnOrigLen:
word; AnOrigTos: byte; AnOrigId: word; AnOrigFrag: word; AnOrigTtl:
byte; AnOrigProtocol: byte; AnOrigSource: TIdInAddr; AnOrigDest:
TIdInAddr; const AnOrigPayload: integer; const APayloadSize: integer;
```

```
var ABuffer): boolean;
```

### 3.3.41. IdRawBuildIgmpp

Builds a datagram using IGMP headers and message formats.

```
function IdRawBuildIgmpp(AType: byte; ACode: byte; AnIp: TIdInAddr;
const APayload: integer; const APayloadSize: integer; var ABuffer):
boolean;
```

### 3.3.42. IdRawBuildIp

Builds a datagram using IP headers and message formats.

```
function IdRawBuildIp(ALen: word; ATos: byte; AnId: word; AFrag: word;
ATtl: byte; AProtocol: byte; ASource: TIdInAddr; ADest: TIdInAddr;
const APayload; APayloadSize: integer; var ABuffer): boolean;
```

### 3.3.43. IdRawBuildRip

Builds a datagram using RIP headers and message formats.

```
function IdRawBuildRip(ACommand: byte; AVersion: byte; ARoutingDomain:
word; AnAddressFamily: word; ARoutingTag: word; AnAddr: longword;
AMask: longword; ANextHop: longword; AMetric: longword; const
APayload; APayloadSize: integer; var ABuffer): boolean;
```

### 3.3.44. IdRawBuildTcp

Builds a datagram using TCP headers and message formats.

```
function IdRawBuildTcp(ASourcePort: word; ADestPort: word; ASeq:
longword; AnAck: longword; AControl: byte; AWindowSize: word;
AnUrgent: word; const APayload: integer; const APayloadSize: integer;
var ABuffer): boolean;
```

### 3.3.45. IdRawBuildUdp

Builds a datagram using UDP headers and message formats.

```
function IdRawBuildUdp(ASourcePort: word; ADestPort: word; const
APayload; APayloadSize: integer; var ABuffer): boolean;
```

### 3.3.46. IncludeTrailingBackSlash

Builds a path string that includes a trailing path delimiter.

```
function IncludeTrailingBackSlash(const APath: string): string;
```

**Parameters**

const APath: string

The initial path string.

**Returns**

String - Path including the trailing path delimiter.

**Description**

IncludeTrailingBackSlash is a String function used to construct a path that includes a trailing path delimiter.

IncludeTrailingBackSlash encapsulates the platform-specific procedures or functions used to build a path string using the platform-specific path delimiter character.

IncludeTrailingBackSlash is a convenience function provided for Delphi or C++ Builder products using a VCL version prior to VCL 5.

### 3.3.47. IncQWord

Increments a Word value in a TQWord instance.

```
procedure IncQWord(var QWord: TQWord; IncVal: LongWord);
```

**Parameters**

var QWord: TQWord

Quad word value to increment.



IncVal: LongWord  
Increment amount.

Description

IncQWord is a procedure used to increment the TQWord specified in QWord by the LongWord value in IncVal.

3.3.48. InfoCallback

Allows the SSL socket to perform status notifications.

procedure InfoCallback(sslSocket: PSSL; where: Integer; ret: Integer);

Description

InfoCallback is a procedure used to retrieve application data and status information for a client or server SSL socket. InfoCallback allow the owner of the SSL socket to call it's DoStatusInfo event when the callback is received.

3.3.49. InitializeMime

Initializes the default MIME values for the locale of the local PC.

procedure InitializeMime(var TransferHeader: TTransfer *⚡*TTransfer;  
var HeaderEncoding: char; var MimeCharSet: string);

Parameters

var TransferHeader: TTransfer  
Transfer encoding scheme.

var HeaderEncoding: char  
Header encoding scheme.

var MimeCharSet: string  
Character set for the local system.

Description

InitializeMime is a procedure the specified parameter variables to the MIME values for the Locale for the local computer. TransferHeader is set to 8-bit encoding. HeaderEncoding is set to the Base64 encoding scheme. MimeCharSet is set to the value returned by GetSystemLocale *⚡*GetSystemLocale.

3.3.50. InMainThread

Indicates if the current thread is the main thread of the application.

function InMainThread: boolean;

Returns

Boolean - True if the current thread is the main thread of execution.

Description

InMainThread is a **Boolean** function that identifies if the current thread represents the main thread of execution in an application.

InMainThread isolates the Indy developer for the platform-specific calls needed to identify the current thread and the application main thread. For WIN32, these calls are GetCurrentThreadId and MainThreadId.

InMainThread is used by TIdAntiFreeze *⚡*TIdAntiFreeze to implement DoProcess functionality.

3.3.51. IntToBin

Returns a binary string representation for an Integer value.

function IntToBin(Value: cardinal): string;

Parameters

Value: cardinal  
Cardinal value to be converted.

Returns

String - Binary representation of the Integer value.

**Description**

IntToBin is a String function used to construct a binary representation of a 32-bit Integer value. The return value for IntToBin will contain a string of "0" or "1" characters for each of the bits in the integer Value.

3.3.52. IsCurrentThread

Indicates if the thread is the current thread of execution.

**function** IsCurrentThread(AThread: TThread): boolean;

**Parameters**

AThread: TThread  
Thread to be examined.

**Returns**

**Boolean - True** if the thread in AThread is the current thread of execution.

**Description**

IsCurrentThread is a **Boolean** function that identifies if the thread in *AThread* is the current thread of execution in a multithreaded application. IsCurrentThread isolates the Indy developer for the platform-specific calls needed to identify the current thread of execution in the application. For WIN32, this call is GetCurrentThreadID. IsCurrentThread compares the ThreadID for *AThread* to the value returned by GetCurrentThreadID to determine its return value. IsCurrentThread is used by TIdThreadMgr *↗* *TIdThreadMgr* descendants, TIdThreadMgrDefault *↗* *TIdThreadMgrDefault* and TIdThreadMgrPool *↗* *TIdThreadMgrPool*, to implement ReleaseThread functionality.

3.3.53. IsNumeric

Determines if a character is a numeric digit.

**function** IsNumeric(c: char): Boolean;

**Parameters**

c: char  
Character to be examined.

**Returns**

Boolean - True if the character is a numeric digit.

**Description**

IsNumeric is a Boolean function that indicates if the character in c contains a numeric digit in the range '0'..'9'.

3.3.54. LoadWinsock

**procedure** LoadWinsock;

**Description**

The text for this function has been generated automatically. This means that it is not documented.

3.3.55. LogicalAnd

Performs a logical AND for Integer arguments.

**function** LogicalAnd(A: Integer; B: Integer): Boolean;

**Parameters**

A: Integer  
Base value.

B: Integer  
Mask Value.

**Returns**

Boolean - Result of the Logical AND operation.

**Description**

LogicalAnd is a Boolean function that returns the result of a logical AND operation for the Integer arguments A and B. LogicalAnd returns True when (A and B) = B.

### 3.3.56. MakeAckPkt

Construct a TFTP acknowledgement packet.

```
function MakeAckPkt(const BlockNumber: Word): string;
```


**Parameters**

`const BlockNumber: Word`  
Block number to be acknowledged.

**Returns**

String - Packet created for the protocol.

**Description**

MakeAckPkt is a String function used to construct and populate an acknowledgement packet used by the TrivialFTP protocol. The acknowledgement packet contains the TFTP\_ACK  TFTP\_ACK OpCode and the value from BlockNumber.

### 3.3.57. MakeTempFilename

Constructs a unique name for a temporary file.

```
function MakeTempFilename: string;
```

**Returns**

String - The unique temporary file name.

**Description**

MakeTempFilename is a String function used to construct a unique file name for a temporary file.  
MakeTempFilename uses the variable ATempPath to store the path portion of the temporary file name, and constructs the file name using the prefix " Indy".  
MakeTempFilename encapsulates the platform-specific procedures and functions used to construct a unique file name for an Operating System file. For the Windows platform, the WIN32 API function used is GetTempFileName. On the Linux platform, tempnam is called to create the temporary file name.

### 3.3.58. Max

Determines the larger of two Integer values.

```
function Max(AValueOne: Integer; AValueTwo: Integer): Integer;
```

**Parameters**

`AValueOne: Integer`  
The first Integer value.

`AValueTwo: Integer`  
The second Integer value.

**Returns**

Integer - The larger of the two Integer values.

**Description**

Max is an Integer function used to determine the larger of two Integer values. Max is provided to avoid any dependency on the Borland Math.pas unit.

### 3.3.59. Min

Determines the smaller of two Integer values.

```
function Min(AValueOne: Integer; AValueTwo: Integer): Integer;
```

**Parameters**

`AValueOne: Integer`  
The first Integer value.

`AValueTwo: Integer`  
The second Integer value.

**Returns**

Integer - The smaller of the two Integer values.

**Description**

Min is an Integer function used to determine the smaller of two Integer values. Min is provided to avoid any dependency on the Borland Math.pas unit.

3.3.60. OffsetFromUTC

Determines the local timezone offset from UTC.

**function** OffsetFromUTC: TDateTime;

**Returns**

TDateTime - Native date and time value that is the hour offset from UTC.

**Description**

OffsetFromUTC is a TDateTime function that returns a native date/time value that is the number of hours that the local timezone differs from a date/time expressed in Universal Time Coordinate (UTC).

OffsetFromUTC is often used to determine the time offset for regions and localities that are subject to Daylight Savings Time conventions.

OffsetFromUTC can generate an EldFailedToRetreiveTimeZoneInfo

⚡EldFailedToRetreiveTimeZoneInfo when the timezone cannot be determined.

**Note:** On the Linux platform, OffsetFromUTC uses the value in GOffsetFromUTC

⚡GOffsetFromUTC as the difference from UTC time coordinates.

3.3.61. ParseNewsGroup

Parses a newsgroup from a list of newsgroups.

**procedure** ParseNewsGroup(ALine: **String**; **var** ANewsGroup: **String**; **var** AHi: Cardinal; **var** ALo: Cardinal; **var** AStatus: **String**);

**Parameters**

ALine: **String**

String received by the server.

**var** ANewsGroup: **String**

Lowest index number for an available article.

**var** AHi: Cardinal

Highest index number for an available article.

**var** ALo: Cardinal

Status of a newsgroup.

**var** AStatus: **String**

**Description**

ParseNewsGroup is a procedure that parses a newsgroup list item from server data retrieved using GetNewsgroupList or GetNewGroupsList method with the AList parameter.

AType is set to the newsgroup type which is usually one of these values:

- y - Posting permitted
- n - Read-Only (no posting permitted)
- m - Moderated

**Note:** If ALow is greater than AHigh, no articles are available for that newsgroup.

3.3.62. ParseURI

Parses the components of a URL.

**procedure** ParseURI(URI: **string**; **Var** Protocol: **string**; **Var** Host: **string**; **Var** path: **string**; **Var** Document: **string**; **Var** Port: **string**; **Var** Bookmark: **string**);

**Parameters**

URI: **string**

The URI to be parsed.

**Var** Protocol: **string**

The Internet protocol used to access the resource.

**Var** Host: **string**

The server or computer where the resource is located.

```
Var path: string
    The relative path of the resource.

Var Document: string
    The name of the resource.

Var Port: string
    Port used to access the resource.

Var Bookmark: string
    anchor pointing to a location in a HTML document (used internally by web-browsers).
```

**Description**  
ParseURI is a procedure used to parse the value in URI, and return the component parts of the UTL in the variable parameters Protocol, Host, Path, Document, Port and Bookmark. ParseURI uses an instance of TIdURI *↗TIdURI* to perform the parses on component values.

3.3.63. ParseXOVER

Parses data received from the SendXOVER method.

```
procedure ParseXOVER(Aline: String; var AArticleIndex: Cardinal; var
ASubject: String; var AFrom: String; var ADate: TDateTime; var AMsgId:
String; var AReferences: String; var AByteCount: Cardinal; var
ALineCount: Cardinal; var AExtraData: String);

Parameters
Aline: String
    Article's index number on the server.

var AArticleIndex: Cardinal
    Article's subject.

var ASubject: String
    Article's From header.
```

```
var AFrom: String
    Article's date.

var ADate: TDateTime
    Article's Message ID.

var AMsgId: String
    Article's References header used for threaded article displays.

var AReferences: String
    Number of bytes in the article.

var AByteCount: Cardinal
    Number of lines in the body of the article.

var ALineCount: Cardinal
    Additional XOVER data returned by some news servers.

var AExtraData: String
```

**Description**  
ParseXOVER is a procedure that parses a line of XOVER data (Aline) retrieved using the SendXOVER method.

3.3.64. PasswordCallback

Performs authentication notifications for an SSL context.

```
function PasswordCallback(buf: PChar; size: Integer; rwflag: Integer;
userdata: Pointer): Integer;

Parameters
buf: PChar
    password from the callback.

size: Integer
    number of bytes in the password.

rwflag: Integer
    read write flag.
```

userdata: Pointer  
SSL context for the callback.

**Returns**  
Integer - Length of the password.

**Description**  
PasswordCallback is an Integer function that allows an SSL context to authentication callback notifications for the client or server SSL context specified in userdata. buf will contain the result of the authentication process.

3.3.65. PosInStrArray

Searches an array of strings for an occurrence of the search string.

**function** PosInStrArray(SearchStr: **string**; Contents: **array of string**;  
**const** CaseSensitive: Boolean): Integer;

**Parameters**  
SearchStr: string  
The value to find in the array of strings.

Contents: array of string  
- The array of strings to be searched.

const CaseSensitive: Boolean = True  
Use case sensitivity, Default value is True.

**Returns**  
Integer - Position of the search string in the array, or -1 when not found.

**Description**  
PosInStrArray is an Integer function used find the first occurrence of the search string specified in SearchStr in the array Contents.  
The lower and upper limits of the array Contents is determined using the Low and High functions.  
When CaseSensitive is True, the values in SearchStr and the current array element must be

an exact match. When CaseSensitive is False, the function AnsiSameText *≠AnsiSameText* is used to compare SearchStr to the string in the current array position.  
The return value for the PosInStrArray function will be -1 if the search string does not exist in Contents.

3.3.66. RegisterCoderClass

Identifies and Adds a coder to the coder collection.

**procedure** RegisterCoderClass(ClassType: CIdCoder *≠CIdCoder*;  
CoderType: Byte; CoderPriority: Byte; ContentType: **String**;  
ContentTransferEncoding: **String**);

**Parameters**  
ClassType: CIdCoder  
Coder class instance for the coder.

CoderType: Byte  
Coder operation performed.

CoderPriority: Byte  
Coder threading priority.

ContentType: String  
Content type handled by this coder.

ContentTransferEncoding: String  
Transfer encoding handled by this coder.

**Description**  
RegisterCoderClass is a procedure that allows registration of a coder class instance in Indy. RegisterCoderClass adds a new instance to the coder collection with the values specified in the parameters ClassType, CoderType, CoderPriority, ContentType, and ContentTransferEncoding. RegisterCoderClass allows simultaneous thread execution of coders, and reuse of registered coder class instances.

### 3.3.67. ReturnMIMEType

Constructs the MIME media type, subtype and encoding for the file.

```
function ReturnMIMEType(var MediaType: String; var EncType: String): Boolean;
```

**Parameters**

```
var MediaType: String
    Media type and subtype for the file.
```

```
var EncType: String
    Encoding for the file.
```

**Returns**

Boolean - True when MIME type is found, False when not found.

**Description**

ReturnMIMEType is a Boolean function that constructs the complete MIME media type, subtype, and encoding from the parameters for the function.

### 3.3.68. RightStr

Returns the right-most part of a string.

```
function RightStr(st: String; Len: Integer): String;
```

**Parameters**

```
st: String
    The value to be examined.
```

```
Len: Integer
    The number of characters to be returned.
```

**Returns**

String - The right-most part of the string.

**Description**

RightStr is a String function used to return to the right-most portion of the string st consisting of the number of characters in Len.  
When Len is larger than the length of the string or Len is 0, the unmodified value of the string is returned.

### 3.3.69. ROL

Calculates a rotated-left value.

```
function ROL(val: LongWord; shift: Byte): LongWord;
```

**Parameters**

```
val: LongWord
    Value to be rotated.
```

```
shift: Byte
    The number of bits to be rotated.
```

**Returns**

LongWord - The value of the rotation.

**Description**

ROL is a LongWord function that returns the LongWord value calculated as the Val parameter rotated left by the number of bit positions specified in the shift parameter.

### 3.3.70. ROR

Calculates a rotated-right value.

```
function ROR(val: LongWord; shift: Byte): LongWord;
```

**Parameters**

```
val: LongWord
    The value to rotated right.
```

shift: Byte  
The number of bits to be rotated.

**Returns**  
LongWord - The result of the rotate-right operation.

**Description**  
ROR is a LongWord function that returns the LongWord value calculated as the Val parameter rotated right by the number of bit positions specified in the shift parameter.

3.3.71. RPos

Finds the index of a search string using a specified starting position.

**function** RPos(const ASub: String; const AIn: String; AStart: Integer): Integer;

**Parameters**  
const ASub: String  
The substring to locate.

const AIn: String  
The value to be searched.

AStart: Integer = -1  
The starting position. Default value is -1.

**Returns**  
Integer - Index position of the search string in the string value.

**Description**  
RPos is an Integer function used to locate the token specified by ASub in the string value specified by AIn using the starting position specified by AStart.  
When AStart is greater than 0, the starting position is the character at the index AStart from the beginning of the string AIn. When AStart is less than 0, the starting position is the character at the index AStart from the end of the string AIn.  
The return value of RPos is 0 when the search string in ASub does not occur in the string AIn.

3.3.72. SendError

Build and send a UDP error message.

**procedure** SendError(UDPBase: TIdUDPBase *⚡*TIdUDPBase; APeerIP: string; const APort: Integer; const ErrNumber: Word; ErrorString: string);  
**procedure** SendError(UDPCliet: TIdUDPCliet *⚡*TIdUDPCliet; const ErrNumber: Word; ErrorString: string);  
**procedure** SendError(UDPBase: TIdUDPBase *⚡*TIdUDPBase; APeerIP: string; const APort: Integer; E: Exception);  
**procedure** SendError(UDPCliet: TIdUDPCliet *⚡*TIdUDPCliet; E: Exception);

**Parameters**  
UDPBase: TIdUDPBase  
UDP client generating the exception.  
  
APeerIP: string  
Exception which should be written in the message.

const APort: Integer  
UDP descendant generating the exception.

const ErrNumber: Word  
IP address of the peer connection.

ErrorString: string  
Port number of the peer connection.

**Description**  
SendError *⚡*SendError is an overloaded method used to construct and send UDP protocol error messages. A UDP error message normally contains the error number and text as specified in ErrNumber and ErrorString.  
When the Exception variant is used, the text of the error message will contain Exception.Message.  
When the TIdUDPCliet *⚡*TIdUDPCliet variant is used, the Host and Port properties of UDPCliet are used to send the error message.



3.3.73. SendError

Build and send a UDP error message.

```
procedure SendError(UDPBase: TIdUDPBase ⚡TIdUDPBase; APeerIP: string;
const APort: Integer; const ErrNumber: Word; ErrorString: string);
procedure SendError(UDPCliant: TIdUDPCliant ⚡TIdUDPCliant; const
ErrNumber: Word; ErrorString: string);
procedure SendError(UDPBase: TIdUDPBase ⚡TIdUDPBase; APeerIP: string;
const APort: Integer; E: Exception);
procedure SendError(UDPCliant: TIdUDPCliant ⚡TIdUDPCliant; E:
Exception);
```

**Parameters**  
UDPCliant: TIdUDPCliant  
UDP client generating the exception.

const ErrNumber: Word  
Exception which should be written in the message.

ErrorString: string  
UDP descendant generating the exception.

**Description**  
SendError *⚡*SendError is an overloaded method used to construct and send UDP protocol error messages. A UDP error message normally contains the error number and text as specified in ErrNumber and ErrorString.  
When the Exception variant is used, the text of the error message will contain Exception.Message.  
When the TIdUDPCliant *⚡*TIdUDPCliant variant is used, the Host and Port properties of UDPCliant are used to send the error message.

3.3.74. SendError

Build and send a UDP error message.

```
procedure SendError(UDPBase: TIdUDPBase ⚡TIdUDPBase; APeerIP: string;
```

```
const APort: Integer; const ErrNumber: Word; ErrorString: string);
procedure SendError(UDPCliant: TIdUDPCliant ⚡TIdUDPCliant; const
ErrNumber: Word; ErrorString: string);
procedure SendError(UDPBase: TIdUDPBase ⚡TIdUDPBase; APeerIP: string;
const APort: Integer; E: Exception);
procedure SendError(UDPCliant: TIdUDPCliant ⚡TIdUDPCliant; E:
Exception);
```

**Parameters**  
UDPBase: TIdUDPBase  
UDP client generating the exception.

APeerIP: string  
Exception which should be written in the message.

const APort: Integer  
UDP descendant generating the exception.

E: Exception  
IP address of the peer connection.

**Description**  
SendError *⚡*SendError is an overloaded method used to construct and send UDP protocol error messages. A UDP error message normally contains the error number and text as specified in ErrNumber and ErrorString.  
When the Exception variant is used, the text of the error message will contain Exception.Message.  
When the TIdUDPCliant *⚡*TIdUDPCliant variant is used, the Host and Port properties of UDPCliant are used to send the error message.

3.3.75. SendError

Build and send a UDP error message.

```
procedure SendError(UDPBase: TIdUDPBase ⚡TIdUDPBase; APeerIP: string;
const APort: Integer; const ErrNumber: Word; ErrorString: string);
procedure SendError(UDPCliant: TIdUDPCliant ⚡TIdUDPCliant; const
ErrNumber: Word; ErrorString: string);
procedure SendError(UDPBase: TIdUDPBase ⚡TIdUDPBase; APeerIP: string;
```

```
const APort: Integer; E: Exception);
procedure SendError(UDPCliient: TIdUDPCliient ⚡TIdUDPCliient; E:
Exception);
```

**Parameters**  
UDPCliient: TIdUDPCliient  
UDP client generating the exception.

E: Exception  
Exception which should be written in the message.

**Description**  
SendError *⚡SendError* is an overloaded method used to construct and send UDP protocol error messages. A UDP error message normally contains the error number and text as specified in ErrNumber and ErrorString.  
When the Exception variant is used, the text of the error message will contain Exception.Message.  
When the TIdUDPCliient *⚡TIdUDPCliient* variant is used, the Host and Port properties of UDPCliient are used to send the error message.

3.3.76. SetLocalTime

Updates the time on the local computer.

```
function SetLocalTime(Value: TDateTime): boolean;
```

**Parameters**  
Value: TDateTime  
The date and time to use for the local computer.

**Returns**  
Boolean - True when the local computer date is updated.

**Description**  
SetLocalTime is a Boolean function used to update the time on the local computer system using the TDateTime value in Value.  
SetLocalTime is used by Indy time synchronization components like TIdSNTP *⚡TIdSNTP* and TIdTime *⚡TIdTime*.  
SetLocalTime encapsulates the platform-specific procedures and functions used to update

the time on the local computer system. For the Windows platform, the procedure used is Windows.SetLocalTime. On the Linux platform, SetLocalTime performs no operation and always returns False.

3.3.77. SetThreadPriority

Updates the priority of a thread.

```
procedure SetThreadPriority(AThread: TThread; const APriority:
TThreadPriority);
```

**Parameters**  
AThread: TThread  
The thread to be updated.

const APriority: TThreadPriority  
The thread priority value to be used for the thread.

**Description**  
SetThreadPriority is a procedure used to update the priority of the thread in AThread to the value specified in APriority.  
SetThreadPriority encapsulates the platform-specific procedures and functions used to update the priority of a thread. For the Windows platform, Priority is a published property of TThread.  
**Note:** Since only root is allowed to adjust thread priorities on the Linux platform, calls to SetThreadPriority are ignored regardless of login context.

3.3.78. Sleep

Suspends the current thread for the specified number of milliseconds.

```
procedure Sleep(ATime: cardinal);
```

**Parameters**  
ATime: cardinal  
The number of milliseconds to sleep.

**Description**

Sleep is a procedure used to suspend the current thread of execution for the number of milliseconds specified in ATime. While the current thread is suspended, control passes to other processes with an equal or higher priority.  
Sleep encapsulates the platform-specific procedures or functions used to suspend the current thread of execution. For the Windows platform, the procedure used is Windows.Sleep. On the Linux platform, Sleep tries to select a non-existent socket handle for the specified number of milliseconds.

3.3.79. StrInternetToDateTime

Returns the native date and time for an Internet timestamp.

```
function StrInternetToDateTime(Value: string): TDateTime;
```

**Parameters**  
Value: string  
The Internet timestamp to be converted.

**Returns**  
TDateTime - The native date and time value for the timestamp.

**Description**  
StrInternetToDateTime is a TDateTime function used to convert the Internet timestamp in Value to a native date and time value.  
To convert a value from a TDateTime to an Internet timestamp, use the DateTimeToInternetStr *↗*DateTimeToInternetStr function.

3.3.80. StrToCard

This function converts a string value into a cardinal..

```
function StrToCard(AVal: String): Cardinal;
```

**Parameters**  
AVal: String  
the string containing a numerical value.

**Returns**  
Cardinal - The value the string was converted to.

**Description**  
This function converts a string value into a cardinal (a 32-bit unsigned value) after stripping off spaces and other control characters.

3.3.81. StrToDay

Returns the day number for a short day name.

```
function StrToDay(const ADay: string): Byte;
```

**Parameters**  
const ADay: string  
The 3-character day of week name.

**Returns**  
Byte - The day of week, or -1 if the day name is not valid.

**Description**  
StrToDay is a Byte function used to calculate the day of week for the short day name in ADay. StrToDay will return -1 when ADay is not a valid day name.

3.3.82. StrToMonth

Returns the month number for a given short month name.

```
function StrToMonth(const AMonth: string): Byte;
```

**Parameters**  
const AMonth: string  
The short month name to be converted.

**Returns**

Byte - The month number or -1 when the month name is invalid.

**Description**

StrToMonth is a Byte function used to the month number for the short month name specified in AMonth. StrToMonth will return -1 if AMonth is not a valid short month name.

3.3.83. StrToWorld

Converts a String to a Word data type.

```
function StrToWorld(const Value: String): Word;
```

**Parameters**

const Value: String  
Data to be converted.

**Returns**

Word - Result of the conversion.

**Description**

StrToWorld is a Word function that converts a String specified in Value to a Word data type.

3.3.84. TimeZoneBias

Calculates the local timezone difference from UTC.

```
function TimeZoneBias: Double;
```

**Returns**

Double - Hours and fractional minutes offset from UTC.

**Description**

TimeZoneBias is a Double function that returns the difference between the local timezone for a computer system and Universal Time Coordinates (UTC).  
TimeZoneBias is expressed using any applicable local timezone conventions like Daylight

Savings Time, and represents the number hours and fractional minutes that the timezone differs from UTC.

**Note:** On the Linux platform, TimeZoneBias uses the value from GTimeZoneBias ~~as~~ *GTimeZoneBias* as the local timezone difference for daylight savings time.

3.3.85. UnloadWinsock

```
procedure UnloadWinsock;
```

**Description**

The text for this function has been generated automatically. This means that it is not documented.

3.3.86. UpCaseFirst

Returns a string with the first letter capitalized.

```
function UpCaseFirst(S: string): string;
```

**Parameters**

S: string  
The value to be capitalized.

**Returns**

String - The string with the initial letter capitalized.

**Description**

UpCaseFirst is a String function that returns a string with the first letter capitalized. All other letters in the string S are converted to lower case.

3.3.87. URLDecode

Converts a URL-Encoded string to a US-ASCII string.

```
function URLDecode(psSrc: string): string;
```

Parameters

psSrc: string  
URL-encoded string to be converted.

Returns

String - The un-encoded ASCII representation of the URL.

Description

URLDecode is a **String** function that converts a URL-encoded string to its representation in the US-ASCII character set.  
URLDecode is based on the URL character encoding rules as described in the Internet Standards document Uniform Resource Locators (URL), RFC 1738, by Tim Berners-Lee.  
URLDecode can used to reverse any string encoding performed using URLEncode  
*↗URLEncode.*

3.3.88. URLEncode

Converts a US-ASCII string to a URL-Encoded string.

**function** URLEncode(const psSrc: string): string;

Parameters

const psSrc: string  
US-ASCII string to be encoded.

Returns

String - URL-encoded representation of the string.

Description

URLEncode is a **String** function that converts a US-ASCII string to its representation in the URL Encoding scheme.  
URLEncode is based on the URL character encoding rules as described in the Internet Standards document Uniform Resource Locators (URL), RFC 1738, by Tim Berners-Lee.  
URLEncode performs special handling for characters deemed unsafe in the URL encoding scheme, and any graphical US-ASCII characters in the range 80-FF hexadecimal.  
Use URLDecode *↗URLDecode* to return a URL-encoded string to its US-ASCII

representation.

3.3.89. VerifyCallback

Perform X.509 certificate verifications.

**function** VerifyCallback(Ok: Integer; ctx: PX509\_STORE\_CTX): Integer;

Description

VerifyCallback is an Integer function that allows Verification callbacks using the X.509 certificate specified in PX509\_STORE\_CTX for an SSL context.

3.3.90. WinsockLoaded

**function** WinsockLoaded: Boolean;

Description

The text for this function has been generated automatically. This means that it is not documented.

3.3.91. WordToStr

Converts a Word data type to a String representation.

**function** WordToStr(const Value: Word): WordStr *↗WordStr*;

Parameters

const Value: Word  
Data to be converted.

Returns

WordStr *↗WordStr* - Result of the conversion.

Description

WordToStr is a WordStr *↗WordStr* function that converts the Word data type specified in

Value to it's WordStr *↗WordStr* representation.

3.3.92. WSAGetAsyncBuflen

**function** WSAGetAsyncBuflen(Param: Longint): Word;  
**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.3.93. WSAGetAsyncError

**function** WSAGetAsyncError(Param: Longint): Word;  
**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.3.94. WSAGetSelectError

**function** WSAGetSelectError(Param: Longint): Word;  
**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.3.95. WSAGetSelectEvent

**function** WSAGetSelectEvent(Param: Longint): Word;  
**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.3.96. WSAMakeSelectReply

**function** WSAMakeSelectReply(Event: Word; Error: Word): Longint;  
**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.3.97. WSAMakeSyncReply

**function** WSAMakeSyncReply(Buflen: Word; Error: Word): Longint;  
**Description**  
The text for this function has been generated automatically. This means that it is not documented.

3.4. Types

3.4.1. CIdCoder

Code class reference.

CIdCoder = **class of** TIdCoder *↗TIdCoder*;  
**Description**  
CIdCoder is a TIdCoser class reference used to represent the coder class type in TIdCoderItem *↗TIdCoderItem* collection items.

3.4.2. CSET

Type used to represent a set of characters.

CSET = **set of** Char;  
**Description**  
CSET is a Set type used to represent a set of characters.

3.4.3. PFDSer

PFDSer = ^TFDSer;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.4. PHostEnt

PHostEnt = ^THostEnt *⚡THostEnt*;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.5. PIdArpHdr

PIdArpHdr = ^TIdArpHdr;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.6. PIdBase64Decoder

PIdBase64Decoder = ^TIdBase64Decoder *⚡TIdBase64Decoder*;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.7. PIdBase64Encoder

PIdBase64Encoder = ^TIdBase64Encoder *⚡TIdBase64Encoder*;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.8. PIdCoder

Pointer to a TIdCoder *⚡TIdCoder*.

PIdCoder = ^TIdCoder *⚡TIdCoder*;

Description

PIdCoder is a pointer to a TIdCoder *⚡TIdCoder*.

3.4.9. PIdCoderItem

Pointer to the TIdCoderItem *⚡TIdCoderItem* collection.

PIdCoderItem = ^TIdCoderItem *⚡TIdCoderItem*;

Description

PIdCoderItem is a Pointer to the TIdCoderItem *⚡TIdCoderItem* collection.

3.4.10. PIdDnsHdr

PIdDnsHdr = ^TIdDnsHdr;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.11. PIdEthernetHdr

PIdEthernetHdr = ^TIdEthernetHdr;

Description

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.12. PIdIcmpEcho

PIdIcmpEcho = ^TIdIcmpEcho;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.13. PIdIcmpFrag

PIdIcmpFrag = ^TIdIcmpFrag;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.14. PIdIcmpHdr

PIdIcmpHdr = ^TIdIcmpHdr;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.15. PIdIcmpTs

PIdIcmpTs = ^TIdIcmpTs;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.16. PIdIgmpHdr

PIdIgmpHdr = ^TIdIgmpHdr;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.17. PIdInAddr

Pointer to an internet address.

PIdInAddr = ^TIdInAddr;

**Description**

PIdInAddr is a TIdInAddr pointer that represents a storage structure for the byte, word, and long representations of an internet address.

### 3.4.18. PIdIpHdr

PIdIpHdr = ^TIdIpHdr;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.19. PIdRipHdr

PIdRipHdr = ^TIdRipHdr;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.20. PIdTcpHdr

PIdTcpHdr = ^TIdTcpHdr;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.21. PIdUdpHdr

PIdUdpHdr = ^TIdUdpHdr;

**Description**

The text for this type has been generated automatically. This means that it is not documented.



3.4.22. PIdUUDecoder

PIdUUDecoder = ^TIdUUDecoder *↗TIdUUDecoder*;  
**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.23. PIdUUEncoder

PIdUUEncoder = ^TIdUUEncoder *↗TIdUUEncoder*;  
**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.24. PIdXXDecoder

PIdXXDecoder = ^TIdXXDecoder *↗TIdXXDecoder*;  
**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.25. PIdXXEncoder

PIdXXEncoder = ^TIdXXEncoder *↗TIdXXEncoder*;  
**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.26. PIMFCoderUsage

Represent a message coder.

PIMFCoderUsage = ^TIMFCoderUsage;  
**Description**

PIMFCoderUsage is a Pointer to a TIMFCoderUsage used to decode a message part. PIMFCoderUsage is used by the TMessage.ReceiveBody routine to allow use of multiple coders to decode a message.

3.4.27. PInAddr

PInAddr = ^TInAddr *↗TInAddr*;  
**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.28. PLinger

PLinger = ^TLinger;  
**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.29. PNetEnt

PNetEnt = ^TNetEnt *↗TNetEnt*;  
**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.30. PProtoEnt

PProtoEnt = ^TProtoEnt *↗TProtoEnt*;  
**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.31. PServEnt

PServEnt = ^TServEnt *↗TServEnt*;  
**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.32. Psockaddr

`Psockaddr = ^Tsockaddr ↗Tsockaddr;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.33. PsockaddrIn

`PsockaddrIn = ^TsockaddrIn;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.34. PsockProto

`PsockProto = ^TsockProto ↗TsockProto;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.35. PTimeVal

`PTimeVal = ^TTimeVal ↗TTimeVal;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.36. PTransmitFileBuffers

`PTransmitFileBuffers = ^TTransmitFileBuffers ↗TTransmitFileBuffers;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.37. T\_\_WSAFDIsSetProc

`T__WSAFDIsSetProc = function (s: TSocket ↗TSocket; var FDSet: TFDSet): Bool;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.38. T128BitRecord

`T128BitRecord = array [0..15] of byte;`

**Description**

T128BitRecord is a record type that represents 128-bit values required by Message Digest encoders and decoders.

### 3.4.39. T160BitRecord

`T160BitRecord = array [0..19] of byte;`

**Description**

T160BitRecord is a record type that represents 160-bit values required by Message Digest encoders and decoders.

### 3.4.40. T16x4LongWordRecord

`T16x4LongWordRecord = array[0..15] of LongWord;`

**Description**

T16x4LongWordRecord is a record type that represents 512-bit values composed of 16 32-bit values as required by Message Digest encoders and decoders.

### 3.4.41. T384BitRecord

`T384BitRecord = array [0..47] of byte;`

**Description**

T384BitRecord is a record type that represents 384-bit values as required by Message Digest encoders and decoders.

3.4.42. T4x4LongWordRecord

```
T4x4LongWordRecord = array [0..3] of LongWord;
```

Description

T4x4LongWordRecord is a record type that represents 128-bit values composed of 4 32-bit values as required by Message Digest encoders and decoders.

3.4.43. T4x4x4LongWordRecord

```
T4x4x4LongWordRecord = array[0..3] of T4x4LongWordRecord  
⚡T4x4LongWordRecord;
```

Description

T4x4x4LongWordRecord is a record type that represents 496-bit values accessed as 32-bit values as required by Message Digest encoders and decoders.

3.4.44. T64BitRecord

```
T64BitRecord = array[0..7] of byte;
```

Description

T64BitRecord is a record type that represents 64-bit values as required by Message Digest encoders and decoders.

3.4.45. TAcceptExProc

```
TAcceptExProc = function (sListenSocket, sAcceptSocket: TSocket  
⚡TSocket; lpOutputBuffer: Pointer; dwReceiveDataLength,  
dwLocalAddressLength, dwRemoteAddressLength: DWORD; var  
lpdwBytesReceived: DWORD; lpOverlapped: POverlapped): BOOL;
```

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.46. TAcceptProc

```
TAcceptProc = function (s: TSocket ⚡TSocket; addr: PSockAddr;  
addrlen: PInteger): TSocket ⚡TSocket;
```

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.47. TAccessFileEvent

Specifies an event type for reading or writing to a file on the server.

```
TAccessFileEvent = procedure (Sender: TObject; var FileName: String;  
const PeerInfo: TPeerInfo; var GrantAccess: Boolean; var AStream:  
TStream; var FreeStreamOnComplete: Boolean) of object;
```

Description

TAccessFileEvent is an event type that signalled when a Trivial FTP server needs to access the contents of a file stored on the local file system on the server.

FileName indicates the name of the file on the server file system.

PeerInfo contain the PeerIp and PeerPort for the client connection performing the file access that generated the event notification.

When GrantAccess is True, the file operation for the event has been is permitted. False indicates that the operation was not permitted or an error has occurred. Update GrantAccess in a TAccessFileEvent event handler to reflect the state of the operation.

AStream contains the data to be written to the local file system as a result of the file operation.

When FreeStreamOnCompleteis True, the TAccessFileEvent event handler should free the stream in AStream.

3.4.48. TAuthenticationEvent

Specifies the event type for user authentication in a TELNET connection.

```
TAuthenticationEvent = procedure (AThread: TIdPeerThread
```

```
ATidPeerThread; const AUsername, APassword: string; var
AAuthenticated: Boolean) of object;
Description
TAuthenticationEvent is an event type used to perform authentication of username and
password information provided by a TELNET connection.
AThread is the peer thread representing the TELNET connection generating the
authentication event.
AUsername is the User name supplied for the connection.
APassword is the password supplied for the connection.
AAuthenticated is a variable parameter use to indicate whether the authentication request is
successful.
TAuthenticationEvent is used by TIdTelnetServer ATidTelnetServer in the OnAuthentication
event handler to allow the server implementation to perform user authentication processing.
```

3.4.49. TAuthenticationType

Represents an e-mail authentication type.

```
TAuthenticationType = (atNone, atLogin);
Description
TAuthenticationType is a type used to specify the authentication type used by an E-Mail
server. Currently, this can be one of two values:


- atNone - no authentication is required
- atLogin - simple authentication is required (AUTH LOGIN)

```

You can find the authentication mechanisms supported on an E-Mail server by using the
AuthSchemesSupported.

3.4.50. TBeforeClientConnectEvent

Event type for connection to a remote system using a mapped port.

```
TBeforeClientConnectEvent = procedure (ASender: TComponent; AThread:
TidPeerThread ATidPeerThread; AClient: TIdTCPClient) of object;
```

```
Description
TBeforeClientConnectEvent is an event type that signals when the local connection for
TIdMappedPort is about to connect to the remote system in OnBeforeClientConnect.
```

3.4.51. TBindProc

```
TBindProc = function (s: TSocket ATSocket; var addr: TSockAddr
ATSockAddr; namelen: Integer): Integer;
```

```
Description
The text for this type has been generated automatically. This means that it is not documented.
```

3.4.52. TCallbackEvent

Callback event type for callbacks that require a message.

```
TCallbackEvent = procedure (Msg: string) of object;
Description
TCallbackEvent is an event type for callbacks that require a string messages.
```

3.4.53. TCharBuf

Specifies the ICMP receive and message buffer data type.

```
TCharBuf = array [1..MAX_PACKET_SIZE] of char;
Description
TCharBuf is an Array of Char type that specifies the data type for ICMP receive buffer and
ICMP message buffer used by TIdICMPClient. TCharBuf is indexed from 1 to
MAX_PACKET_SIZE ATMAX_PACKET_SIZE.
```

### 3.4.54. TCharSet

Represents character sets used by GetSystemLocale *↗GetSystemLocale*.

```
TCharSet = (csGB2312, csBig5, csIso2022jp, csEucKR, csIso88591);
```

**Description**

TCharSet is an enumerated type used to represent the valid character sets used by the GetSystemLocal function. Valid value for TCharset includes:

- csGB2312 - Simplified Chinese character set.
- csBig5 - Full Chinese character set.
- csIso2022jp - Japanese character set.
- csEucKR - Korean character set.
- csIso88591 - Default ISO 8859-1 character set.

### 3.4.55. TClassIdException

Represents an exception class for Indy components.

```
TClassIdException = class of EIdException ↗EIdException;
```

**Description**

TClassIdException is used to indicate the type of exception some routines should raise.

### 3.4.56. TClientEvent

Event type that occurs when connecting or disconnecting the TELNET client.

```
TClientEvent = procedure of object;
```

**Description**

TClientEvent is the event type that occurs when the TIdTelnet *↗TIdTelnet* client opens or closes a connection to the host.

The TClientEvent event is used to notify the TIdTelnet *↗TIdTelnet* event handlers OnConnect and OnDisconnect that the corresponding operation has occurred.

### 3.4.57. TClosesocketProc

```
TClosesocketProc = function (s: TSocket): Integer;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.58. TCommandEvent

IMAP4 command event type.

```
TCommandEvent = procedure (Thread : TIdPeerThread ↗TIdPeerThread;  
const Tag, CmdStr: String; var Handled: Boolean) of object;
```

**Description**

TCommandEvent is an event fired for every IMAP4 command which is supported by the TIdMAPI4Server component.

The Thread parameter is the thread for this connection.

Tag is the command tag which was sent from the IMAP4 client.

- This tag is a 4-digit continuous number (e.g. 000A) which is sent by the IMAP client in front of the command. If you "answer" a command you should use this tag to mark the reply. The digits normally running from "0" to "Z". (0-9,A-Z).

Command contains a string of the parameters following the detected command.

- If the command is recognized by the IMAP server (e.g. LOGIN or CAPABILITY), this string contains only the following parameters.
- The list of available IMAP commands and parameters is handled within the RFC2060 available at <http://www.imap.org> .

Set handled to True if you have completely handled the event.

### 3.4.59. TConnectionResult

Result of a connection request.

```
TConnectionResult = (crCanPost, crNoPost, crAuthRequired, crTempUnavailable);
```

**Description**

TConnectionResult is used to store the server response to the connection request. When the IdNNTP client connects to a NNTP server, the server will send a numeric status response to the client. This response can be of the form:

- **200** - NNTP Service Ready, posting permitted.
- **502** - NNTP Service Unavailable.
- **201** - NNTP Service Ready, posting prohibited.
- **400** - NNTP Service temporarily unavailable.

These possible responses are interpreted by the IdNNTP.Connect *↔Connect* method and consequently the appropriate enumerated value is set in TConnectionResult to:

- **200** - crCanPost
- **201** - crNoPost
- **400** - crTempUnavailable

### 3.4.60. TConnectProc

```
TConnectProc = function (s: TSocket ↔TSocket; var name: TSockAddr ↔TSockAddr; namelen: Integer): Integer;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.61. TDataEvent

Specifies the event type for NNTP operations that provide object data.

```
TDataEvent = procedure (AThread: TIdPeerThread ↔TIdPeerThread; AData:TObject) of object;
```

**Description**

TDataEvent is the event type triggered for NNTP Server operations that provide object data for the NNTP operation. *AThread* is the thread of execution performing the NNTP command that triggered the event. *AData* is the TObject descendant that contains data related to the NNTP operation.

### 3.4.62. TDays

```
TDays = (TDaySun, TDayMon, TDayTue, TDayWed, TDayThu, TDayFri, TDaySat);
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.63. TDoByIDEvent

Specifies the event type for NNTP operations that use a message id.

```
TDoByIDEvent = procedure (AThread: TIdPeerThread ↔TIdPeerThread; AActualID:string) of object;
```

**Description**

TDoByIDEvent is the event type triggered for NNTP Server operations that manipulate articles based on their message identifier. *AThread* is the thread of execution performing the NNTP command that triggered the event. *AActualID* is the message id for the news article to be affected by the NNTP command.

### 3.4.64. TDoByNoEvent

Specifies the event type for NNTP operations that use a message number.

```
TDoByNoEvent = procedure (AThread: TIdPeerThread
```

*⚡TidPeerThread;AActualNumber:Cardinal) of object;*

**Description**

TDoByNoEvent is the event type triggered for NNTP Server operations that manipulate articles based on their relative message number.  
*AThread* is the thread of execution performing the NNTP command that triggered the event.  
*AActualNumber* is the relative message number for the news article to be affected by the NNTP command.

3.4.65. TEventNewNewsList

Indicates a new newsgroup article.

TEventNewNewsList = **procedure** (**const** AMsgID: **string**; **var** ACanContinue: Boolean) **of object**;

**Description**

TEventNewNewsList is an event type triggered when a Message-Id has been received using the GetNewNews method without the AList parameter.  
MsgID is the message ID of the new article.  
Setting ACanContinue to false will stop the request.

3.4.66. TEventNewsgroupList

Indicates receipt of a newsgroup listing.

TEventNewsgroupList = **procedure** (**const** ANewsgroup: **string**; **const** ALow, AHigh: Cardinal; **const** AType: **string**; **var** ACanContinue: Boolean) **of object**;

**Description**

TEventNewsgroupList is an event type triggered when a newsgroup is retrieved using the GetNewsgroupList or GetNewGroupsList method without the AList parameter.  
AType usually is one of these values:

- y - Posting permitted
- n - Read-Only (no posting permitted)
- m - Moderated

**Note:** If ALow is greater than AHigh, no articles are available in this newsgroup.

3.4.67. TEventStreaming

Event for receipt or transmission of streaming news articles.

TEventStreaming = **procedure** (**const** AMesgID: **string**; **var** AAccepted: Boolean) **of object**;

**Description**

TEventStreaming is an event type triggered by servers or news forwarding clients that support streaming mode transmission of news articles.  
Set AAccepted to **False** to inform the peer that the new article is not accepted or already on the server.

3.4.68. TGetAcceptExSockaddrsProc

TGetAcceptExSockaddrsProc = **procedure** (lpOutputBuffer: Pointer; dwReceiveDataLength, dwLocalAddressLength, dwRemoteAddressLength: DWORD; **var** LocalSockaddr: TSocketAddr ⚡TSocketAddr; **var** LocalSockaddrLength: Integer; **var** RemoteSockaddr: TSocketAddr ⚡TSocketAddr; **var** RemoteSockaddrLength: Integer);

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.69. TGetEvent

Specifies the event type for NNTP operations that do not require parameters.

TGetEvent = **procedure** (AThread: TidPeerThread ⚡TidPeerThread; ALookup: **string**) **of object**;

**Description**

TGetEvent is the event type triggered for NNTP Server operations that do not require

parameters.  
*AThread* is the thread of execution performing the NNTP command that triggered the event.

### 3.4.70. TGetHostByAddrProc

```
TGetHostByAddrProc = function (addr: Pointer; len, Struct: Integer):
    PHostEnt ⚡PHostEnt;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.71. TGetHostByNameProc

```
TGetHostByNameProc = function (name: PChar): PHostEnt ⚡PHostEnt;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.72. TGetHostNameProc

```
TGetHostNameProc = function (name: PChar; len: Integer): Integer;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.73. TGetPeerNameProc

```
TGetPeerNameProc = function (s: TSocket ⚡TSocket; var name: TSocketAddr
    ⚡TSocketAddr; var namelen: Integer): Integer;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.74. TGetProtoByNameProc

```
TGetProtoByNameProc = function (name: PChar): PProtoEnt ⚡PProtoEnt;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.75. TgetProtoByNumberProc

```
TgetProtoByNumberProc = function (proto: Integer): PProtoEnt
    ⚡PProtoEnt;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.76. TGetServByNameProc

```
TGetServByNameProc = function (name, proto: PChar): PServEnt
    ⚡PServEnt;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.77. TGetServByPortProc

```
TGetServByPortProc = function (port: Integer; proto: PChar): PServEnt
    ⚡PServEnt;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.78. TGetSockNameProc

```
TGetSockNameProc = function (s: TSocket ⚡TSocket; var name: TSocketAddr
    ⚡TSocketAddr; var namelen: Integer): Integer;
```

**Description**



The text for this type has been generated automatically. This means that it is not documented.

3.4.79. TGetSockOptProc

```
TGetSockOptProc = function (s: TSocket ⚡TSocket; level, optname: Integer; optval: PChar; var optlen: Integer): Integer;
```

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.80. TGroupEvent

Specifies the event type for NNTP newsgroup operations.

```
TGroupEvent = procedure (AThread: TIdPeerThread ⚡TIdPeerThread; AGroup: string) of object;
```

Description

TGroupEvent is the event type triggered for NNTP Server operations that require a newsgroup name as a parameter.  
*AThread* is the thread of execution performing the NNTP command that triggered the event.  
*AGroup* is the name of the newsgroup to be affected by the NNTP command.

3.4.81. THostEnt

```
THostEnt = hostent;
```

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.82. THostNameGetEvent

Event type for HostName commands without parameters.

```
THostNameGetEvent = procedure (Thread: TIdPeerThread) of object;
```

Description

THostNameGetEvent is an event type generated for TIdHostNameServer *⚡*TIdHostNameServer commands that do not have parameters, including:

- ALL
- HELP
- VERSION
- ALLOLD
- DOMAINS
- ALLDOM
- ALLINGWAY

TIdHostNameServer *⚡*TIdHostNameServer provides event handlers for the event notification to allow the server to respond to the HostName commands.

3.4.83. THostNameOneParmEvent

Event type for HostName commands with parameters.

```
THostNameOneParmEvent = procedure (Thread: TIdPeerThread ⚡TIdPeerThread; Parm: string) of object;
```

Description

THostNameOneParmEvent is an event type generated for TIdHostNameServer *⚡*TIdHostNameServer commands that have parameters, including:

- HNAME
- HADDR

TIdHostNameServer *⚡*TIdHostNameServer provides event handlers for the event notification to allow the server to respond to the HostName commands.

3.4.84. THtonIProc

```
THtonIProc = function (hostlong: u_long): u_long ⚡u_long;
```

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.85. THtonsProc

```
THtonsProc = function (hostshort: u_short): u_short u_short;
```

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.86. TICMPDataBuffer

Specifies the data type for ICMP message data.

```
TICMPDataBuffer = array [1..iDEFAULTPACKETSIZE] of byte;
```

Description

TICMPDataBuffer is an Array of Byte type that specifies the data type for ICMP message data received in an ICMP response message.

3.4.87. TIdCardAddressAttributes

Describes a VCard address type.

```
TIdCardAddressAttributes = set of ( tatHome, tatDomestic, tatInternational, tatPostal, tatParcel, tatWork, tatPreferred );
```

Description

TIdCardAddressAttributes is a set type used to describe an address and can include the following:.

- tatHome - home delivery address
- tatDomestic - a domestic address
- tatInternational - an international address
- tatPostal - a postal delivery address
- tatParcel - a parcel delivery address
- tatWork - a work delivery address
- tatPreferred - preferred address

3.4.88. TIdDICTAuthEvent

Specifies the event type for the DICT AUTH and SASLAUTH commands.

```
TIdDICTAuthEvent = procedure ( Thread: TIdPeerThread TIdPeerThread; Username, authstring : String ) of object;
```

Description

TIdDICTAuthEvent specifies the event type for the DICT AUTH and SASLAUTH commands, and is used by the TIdDICTServer ~~TIdDICTServer~~ OnCommandAuth and OnCommandSASLAAuth event handlers to perform user authentication.

Username contains the user name to be authenticated.

For the DICT AUTH command, AuthString will contain an APOP-style MD5 Checksum as described in RFC 1939.

For the DICT SASLAUTH command, AuthString will contain the BASE64-encoded initial SASL response as described in RFC 2045. All responses to the SASLAUTH server challenge should use the SASLRESP command and a BASE64-encoded parameter.

Authentication is an optional server capability. The AUTH and SASLAUTH commands may be implemented by a DICT server.

3.4.89. TIdDICTDefineEvent

Specifies the event type for the DICT DEFINE command.

```
TIdDICTDefineEvent = procedure ( Thread: TIdPeerThread TIdPeerThread; Database, WordToFind : String ) of object;
```

Description

TIdDICTDefineEvent specifies the event type for the DICT DEFINE command, and is used by the OnCommandDefine event handler to perform a search for the specified word in the indicated database(s).

Database may contain a name in the form returned by the DICT SHOW DB command, or one of the following wildcard characters:

- "!" - Search all databases until a match is found.
- "\*" - Search the current database for all matching entries.

### 3.4.90. TIdDICTGetEvent

Specifies the event type for the DICT HELP, QUIT, and STATUS commands.

```
TIdDICTGetEvent = procedure ( Thread: TIdPeerThread) of object;
```

**Description**

TIdDICTGetEvent specifies the event type for the DICT HELP, QUIT, and STATUS commands. TIdDICTGetEvent is used to notify the corresponding event handler that the operation should be performed.

### 3.4.91. TIdDICTMatchEvent

Specifies the event type for the DICT MATCH command.

```
TIdDICTMatchEvent = procedure ( Thread: TIdPeerThread ≍TIdPeerThread;  
Database, Strategy,WordToFind : String ) of object;
```

**Description**

TIdDICTMatchEvent specifies the event type for the DICT MATCH command, and is used by the TIdDICTServer *≍*TIdDICTServer OnCommandMatch event handler to search the dictionary index and report the words found using a particular strategy. All DICT servers **must** implement the MATCH command, and **must** support the "exact" and "prefix" strategies. The "exact" strategy performs a case-insensitive comparison. The "prefix" strategy is similar to "exact", except that it only compares the first part of the word. These strategies are easily implemented and are generally the most useful. Other strategies are server dependent. Other strategies that might be considered are matches based on substring, suffix, regular expressions, soundex [KNUTH73], and Levenshtein [PZ85] algorithms. These last two are especially useful for correcting spelling errors. Other useful strategies perform some sort of "reverse" lookup (i.e., by searching definitions to find the word that the query suggests). Strategy may also contain the special character ".", which indicates that the server should use the default strategy for that server. This is usually a derivative of the Levenshtein algorithm [PZ85]. Database contains the name of the database to be searched, in a form returned by SHOW DB, or one the following wildcard characters:

- "!" - Search all databases until a match is found.

- "\*" - Search for all matches in the current database.

### 3.4.92. TIdDICTOtherEvent

Specifies the event type for the DICT OPTION or other unrecognized commands.

```
TIdDICTOtherEvent = procedure ( Thread: TIdPeerThread ≍TIdPeerThread;  
Command, Parm:String ) of object;
```

**Description**

TIdDICTOtherEvent specifies the event type for the DICT OPTION or other unrecognized commands. TIdDICTServer *≍*TIdDICTServer uses the OnCommandOption event handler to respond to the DICT OPTION command, which informs the server that the client wishes to utilize one of the defined protocol options. At the current time, OPTION MIME is the only recognized option. In the future, standard extensions to this protocol should be proposed which allow the client to request certain content types or encodings. OPTION MIME requests that all text responses be prefaced by a MIME header, as described in RFC2045, and followed by a single blank line (CRLF). If a client requests this option, then the client **MUST** be able to parse Content-Type and Content-transfer-encoding headers, and **MUST** be able to ignore textual responses which have an unsupported content or encoding. A client **MUST** support the UTF-8 encoding described in RFC2044. OPTION MIME is a **REQUIRED** server capability, and all DICT servers **MUST** implement this command.

### 3.4.93. TIdDICTShowEvent

Specifies the event type for the DICT SHOW command.

```
TIdDICTShowEvent = procedure ( Thread: TIdPeerThread ≍TIdPeerThread;  
Command : String ) of object;
```

**Description**

TIdDICTShowEvent specifies the event type for the DICT SHOW command, and is used by the TIdDICTServer *≍*TIdDICTServer OnCommandShow event handler to implement the server command.

The DICT SHOW command accepts parameters to indicate the information to be generated by the server, including the following:

- SHOW DB - Displays the list of currently accessible databases.
- SHOW DATABASES - Displays the list of currently accessible databases.
- SHOW STRAT - Displays the list of currently supported search strategies.
- SHOW STRATEGIES - Displays the list of currently supported search strategies.
- SHOW INFO database - Displays the source, copyright, and licensing information about the specified database.
- SHOW SERVER - Displays local server herald banner.

3.4.94. TIdExceptionEvent

Specifies the event handler for thread exceptions.

```
TIdExceptionEvent = procedure (Sender: TObject; E: Exception) of
object;
```

Description

TIdExceptionEvent is an event handler that allows TIdThread.OnException event notifications when exceptions are raised in a thread of execution.

3.4.95. TIdFingerGetEvent

Represents the Finger event for a user information request.

```
TIdFingerGetEvent = procedure (AThread: TIdPeerThread ≠TIdPeerThread;
const AUserName: String) of object;
```

Description

TIdFingerGetEvent is the event triggered when a request for user information is detected. TIdFingerEvent is used by the TIdFingerServer *≠TIdFingerServer* OnCommandFinger and OnCommandVerboseFinger event handlers. AThread is the current thread of execution in the Finger server. AUserName is the user token to be handled by the Finger server. UserName can contain optional tokens as identified in RFC 1288.

3.4.96. TIdFTPTransferType

File transfer type.

```
TIdFTPTransferType = (ftBinary, ftASCII);
```

Description

TIdFTPTransferType is the type is used to indicate the file transfer type for sending and receiving files and can be one of these values:

- ftBinary - Binary (8 bit) file transfers
- ftASCII - ASCII (7 bit) file transfers

3.4.97. TIdGopherMenuEvent

Event type for Gopher menu display.

```
TIdGopherMenuEvent = procedure ( Sender : TObject; MenuItem :
TIdGopherMenuItem ≠TIdGopherMenuItem ) of object;
```

Description

TIdGopherMenuEvent is an event type triggered for each Gopher menu item retrieved with the GetMenu and GetExtendedMenu methods. TIdGopherMenuEvent is used to display Gopher menu items while the complete menu is downloaded.

3.4.98. TIdHTTPGetEvent

Specifies an event type for HTTP transfer commands.

```
TIdHTTPGetEvent = procedure (AThread: TIdPeerThread ≠TIdPeerThread;
RequestInfo: TIdHTTPRequestInfo ≠TIdHTTPRequestInfo; ResponseInfo:
TIdHTTPResponseInfo) of object;
```

Description

TIdHTTPGetEvent represents an event type used to signal receipt of a HTTP command that transmits HTTP resources. TIdHTTPGetEvent is used for HTTP GET, POST, and HEAD commands.

AThread is the thread of execution triggering the TIdHTTPGetEvent event.  
RequestInfo represents the header values to be used for a GET, HEAD or POST command as they are captured from the HTTP client.  
ResponseInfo represents the header values to be returned to the HTTP client.  
TIdHTTPServer *↗* *TIdHTTPServer* provides the OnCommandGet event handler to respond to the TIdHTTPGetEvent notification.

### 3.4.99. TIdHTTPMethod

Represents the HTTP Method for a request.

```
TIdHTTPMethod = (hmHead, hmGet, hmPost);
```

**Description**

TIdHTTPMethod is an enumerated type used to represent the HTTP method for a request.  
TIdHTTPMethod can be one of the following values:

- hmHead - Head
- hmGet - Get
- hmPost - Post

TIdHTTPMethod is used by TIdHTTP *↗* *TIdHTTP* in the DoRequest method.

### 3.4.100. TIdHTTPOnRedirectEvent

Event type triggered for a HTTP redirection request.

```
TIdHTTPOnRedirectEvent = procedure (Sender: TObject; var dest: String;  
var NumRedirect: Integer; var Handled: boolean) of object;
```

**Description**

TIdHTTPOnRedirectEvent is a event type triggered when an HTTP client is notified of a redirection request from the HTTP server. TIdHTTPOnRedirectEvent is used in conjunction with an event handler to respond to the event.  
Sender is the HTTP client that has received the redirection request.  
Dest is the new location or URL for the resource. Dest is often used to update a GUI

application with the current URL.  
NumRedirects is a variable used to accumulate the number of redirection requests that the HTTP client has received.  
Handled indicates that the event handler should respond to the redirection request. Set Handled to False if the application is unable to respond to the redirection request.

### 3.4.101. TIdHTTPOtherEvent

Specifies an event type for unrecognized HTTP commands.

```
TIdHTTPOtherEvent = procedure (Thread: TIdPeerThread ↗ TIdPeerThread;  
const asCommand, asData, asVersion: string) of object;
```

**Description**

TIdHTTPOtherEvent represents the event type used to signal receipt of a HTTP command other than GET, POST, or HEAD.  
TIdHTTPServer *↗* *TIdHTTPServer* provides the OnCommandOther event handler to respond to the IdHTTPOtherEvent notification.

### 3.4.102. TIdHTTPProtocolVersion

Represents the HTTP Protocol version for a request.

```
TIdHTTPProtocolVersion = (pv1_0, pv1_1);
```

**Description**

TIdHTTPProtocolVersion is an enumerated type used to represent the HTTP Protocol version for a request. TIdHTTPProtocolVersion can be one of the following values:

- pv1\_0 - HTTP 1.0
- pv1\_1 - HTTP 1.1

TIdHTTPProtocolVersion identifies the protocol version that an HTTP server must support in order to respond to the Get, Head, Post, or PostFromStream request.

### 3.4.103. TidlrcFiveParmEvent

Event type for IRC commands having 5 parameters.

```
TIdIrcFiveParmEvent = procedure (Thread: TIdPeerThread
  ⚡TIdPeerThread; Parm1, Parm2, Parm3, Parm4, Parm5 : String) of
object;
```

**Description**

TidlrcFiveParmEvent is an event type for IRC commands having 5 parameters.

### 3.4.104. TidlrcGetEvent

Event type for IRC commands that do not have parameters.

```
TIdIrcGetEvent = procedure ( Thread: TIdPeerThread) of object;
```

**Description**

TidlrcGetEvent is the event type generated for IRC commands that do not have parameters.

### 3.4.105. TidlrcOneParmEvent

Event type for IRC commands having a single parameter.

```
TIdIrcOneParmEvent = procedure (Thread: TIdPeerThread ⚡TIdPeerThread;
  Parm : String) of object;
```

**Description**

TidlrcOneParmEvent is an event type for IRC server commands having a single parameter.

### 3.4.106. TidlrcOtherEvent

Specifies the event type for unknown IRC commands.

```
TIdIrcOtherEvent = procedure ( Thread: TIdPeerThread ⚡TIdPeerThread;
  Command, Parm : String) of object;
```

**Description**

TidlrcOtherEvent is the event type for unknown IRC commands.

### 3.4.107. TidlrcServerEvent

Event type for the IRC SERVER command.

```
TIdIrcServerEvent = procedure ( Thread: TIdPeerThread ⚡TIdPeerThread;
  ServerName, Hopcount, Info : String) of object;
```

**Description**

TidlrcServerEvent is an event type for the IRC SERVER command.

### 3.4.108. TidlrcThreeParmEvent

Event type for IRC commands having 3 parameters.

```
TIdIrcThreeParmEvent = procedure (Thread: TIdPeerThread
  ⚡TIdPeerThread; Parm1, Parm2, Parm3 : String) of object;
```

**Description**

TidlrcThreeParmEvent is an event type for IRC commands having 3 parameters.

### 3.4.109. TIdIrcTwoParmEvent

Event type for IRC commands having 2 parameters.

```
TIdIrcTwoParmEvent = procedure (Thread: TIdPeerThread ≍TIdPeerThread;  
Parm1, Parm2 : String) of object;
```

**Description**

TIdIrcTwoParmEvent is an event type for IRC commands having 2 parameters.

### 3.4.110. TIdIrcUserEvent

Event type for the IRC User command.

```
TIdIrcUserEvent = procedure ( Thread: TIdPeerThread ≍TIdPeerThread;  
UserName, HostName, ServerName, RealName : String) of object;
```

**Description**

TIdIrcUserEvent is the event type for the IRC User command.

### 3.4.111. TIdLinger

Linger data type for Indy.

```
TIdLinger = TLinger;
```

**Description**

TIdLinger represents the TLinger type used in Indy protocol stack interface implementations.

### 3.4.112. TIdLogDebugTarget

Specifies the target for TIdLogDebug *≍*TIdLogDebug log messages.

```
TIdLogDebugTarget = (ItFile, ItDebugOutput, ItEvent);
```

**Description**

TIdLogDebugTarget is an enumerated type that specifies the target for TIdLogDebug *≍*TIdLogDebug log messages. TIdLogDebugTarget can have the following values:

- ItFile - Write log messages to a file.
- ItDebugOutput - Write log messages to Debugger Output.
- ItEvent - Do not output messages, fire events only.

TIdLogDebugTarget is used to specify the value of the TIdLogDebug.Target property and the target for log messages in TIdLogDebug *≍*TIdLogDebug.

### 3.4.113. TIdMessageEvent

Specified the event type for message handling.

```
TIdMessageEvent = procedure (ASender : TComponent; var AMsg :  
TIdMessage) of object;
```

**Description**

TIdMessageEvent is an event type used to signal events that occur when handling the TIdMessage *≍*TIdMessage specified in AMsg.

### 3.4.114. TIdMessagePartClass

```
TIdMessagePartClass = class of TIdMessagePart ≍TIdMessagePart;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.115. TIdMessagePriority

Specifies the message priority.

```
TIdMessagePriority = (mpHighest, mpHigh, mpNormal, mpLow, mpLowest);
```

**Description**

TidMessagePriority is an enumerated type used to describe the priority of a message. TidMessagePriority can contain one of the following values:

- mpHighest - The message should be given the highest priority.
- mpHigh - This message should be given a high priority.
- mpNormal - This message should be given the normal priority. This is the default value.
- mpLow - This message should be given a low priority.
- mpLowest - This message should be given the lowest priority.

TidMessagePriority is used by TidMessage *≧*TidMessage to represent the value of either the "Priority:" or "X-Priority:" headers in a RFC 822 message.

3.4.116. TidNetTime

TidNetTime = longword;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.117. TidPhoneAttributes

Describes a VCard telephone number.

TidPhoneAttributes = **set of** ( tpaHome, tpaVoiceMessaging, tpaWork, tpaPreferred, tpaVoice, tpaFax, paCellular, tpaVideo, tpaBBS, tpaModem, tpaCar, tpaISDN, tpaPCS, tpaPager);

**Description**

TidPhoneAttributes is an enumerated type that describes a telephone number type, and can include the following values:

- tpaHome - home telephone number
- tpaVoiceMessaging - voice messaging is available
- tpaWork - work telephone number
- tpaPreferred - preferred telephone number
- tpaVoice - voice telephone number

- tpaFax - facsimile telephone number
- tpaCellular - cellular telephone number
- tpaVideo - video-phone number
- tpaBBS - Bulletin Board System
- tpaModem - Modem telephone number
- tpaCar - Car telephone
- tpaISDN - ISDN telephone number
- tpaPCS - personal communication services telephone number
- tpaPager - pager telephone number

TidPID

3.4.118. TidPID

Represents an Indy process identifier.

TidPID = LongWord;

**Description**

TidPID is the type used to represent Indy Process identifiers, and encapsulates the platform-specific data type used to represent a process identifier. For the Windows platform, TidPID is defined as a LongWord value as returned by the CurrentProcessID function. For the Linux platform, TidPID is defined as an Integer value as returned by the CurrentProcessID function.

3.4.119. TidQOTDGetEvent

Specifies an event type for TidQuotdServer commands.

TidQOTDGetEvent = **procedure** ( Thread: TidPeerThread *≧*TidPeerThread ) of object;

**Description**

TidQOTDGetEvent is an event type that allows TidQuotdServer to respond to the OnCommandQOTD event. Thread is the thread for the client connection.



3.4.120. TIdServeFile

Function type for an optimized file transmission procedure.

```
TIdServeFile = function (ASocket: TIdStackSocketHandle
  ⚡TIdStackSocketHandle; AFileName: string): cardinal;
Description
TIdServeFile is a Cardinal function type that represents the function used to perform file
transmission. TIdServeFile abstracts the ServeFile function under Windows NT that is
optimized for sequential read-only access. GServeFile is the application global instance of the
function for Indy.
```

3.4.121. TIdServerThreadEvent

Specifies an event type for connections on a TCP server.

```
TIdServerThreadEvent = procedure (AThread: TIdPeerThread) of object;
Description
TIdServerThreadEvent is a event type that allows the TCP server to respond to requests
generated by a TIdPeerThread ⚡TIdPeerThread connections on the server.
TIdServerThreadEvent is generated when a peer thread requests a connection, begins
executing, or intends to close the connection on the TCP server.
```

3.4.122. TIdSSLAction

SSL socket action allowed.

```
TIdSSLAction = (sslRead, sslWrite);
Description
TIdSSLAction is an enumerated type that represents the action allowed using the SSL socket.
```

3.4.123. TIdSSLErrorMode

Error type for the SSL exception.

```
TIdSSLErrorMode = (sslemClient, sslemServer);
Description
TIdSSLErrorMode is an enumerated type that indicates the error type for the SSL exception.
```

3.4.124. TIdSSLMode

SSL context mode values.

```
TIdSSLMode = (sslmUnassigned, sslmClient, sslmServer, sslmBoth);
Description
TIdSSLMode is an enumerated type that represents the valid values for the mode of the SSL
context.
```

3.4.125. TIdSSLVerifyMode

Valid verify mode values for an SSL context.

```
TIdSSLVerifyMode = (sslvrfPeer, sslvrfFailIfNoPeerCert,
  sslvrfClientOnce);
Description
TIdSSLVerifyMode is an enumerated type that represents the valid values for verify mode of
an SSL context.
```

3.4.126. TIdSSLVerifyModeSet

Set of verify mode values for the SSL context.

```
TIdSSLVerifyModeSet = set of TIdSSLVerifyMode ⚡TIdSSLVerifyMode;
```

**Description**

TIdSSLVerifyModeSet is a Set type that contains TIdSSLVerifyMode *↗TIdSSLVerifyMode* values for the SSL context.

3.4.127. TIdSSLVersion

SSL version numbers.

```
TIdSSLVersion = (sslvSSLv2, sslvSSLv23, sslvSSLv3, sslvTLSv1);
```

**Description**

TIdSSLVersion is an enumerated type that represents the valid SSL Version numbers for Indy OpenSSL components.

3.4.128. TIdStackSocketHandle

```
TIdStackSocketHandle = TSocket ↗TSocket;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.129. TIdStatisticsOperation

Represents the statistical operation for the Tunnel component.

```
TIdStatisticsOperation = (soIncrease, soDecrease);
```

**Description**

TIdStatisticsOperation is an enumerated type that represents the type of statistical operation to be performed for a Tunnel component.

3.4.130. TIdStatus

Represents the connection status for a component.

```
TIdStatus = (hsResolving, hsConnecting, hsConnected, hsDisconnecting, hsDisconnected, hsText);
```

**Description**

TIdStatus is an enumerated type that represents the connection status for an Indy component. TIdStatus can contain one of the following values:

- hsResolving - A host name is being resolved to an IP Address
- hsConnecting - A connection is being opened
- hsConnected - A connection has been made
- hsDisconnecting - The connection is being closed
- hsDisconnected - The connection has been closed
- hsText - The connection is generating an informational message

3.4.131. TIdStatusEvent

Represents a connection status event for the component.

```
TIdStatusEvent = procedure (axSender: TObject; const axStatus: TIdStatus ↗TIdStatus; const asStatusText: string) of object;
```

**Description**

TIdStatusEvent is an event type triggered during a change in the connection status for the component. TIdStatusEvent is triggered by the DoStatus method. axStatus is the TIdStatus *↗TIdStatus* value for the current connection, and can contain one of the following values:

- hsResolving - A host name is being resolved for an IP address
- hsConnecting - A connection is being opened
- hsConnected - A connection has been made
- hsDisconnecting - The connection is being closed
- hsDisconnected - The connection has been closed
- hsText - The connection has generated an informational message

axStatusText is the suggested text for displaying the component connection status.

### 3.4.132. TIdStringMessageEvent

Specified the event type for string events.

```
TIdStringMessageEvent = procedure (ASender : TComponent; const AString
: String; var AMsg : TIdMessage) of object;
```

**Description**

TIdStringMessageEvent is the event type used to signal events that occur for the TIdMessage *↗TIdMessage* that can be identified by the value specified in AString.

### 3.4.133. TIdTelnetNegotiateEvent

Specifies an event type for TELNET option negotiation.

```
TIdTelnetNegotiateEvent = procedure (AThread: TIdPeerThread) of
object;
```

**Description**

TIdTelnetNegotiateEvent is an event type used to signal option negotiation features as describe in the TELNET protocol specifications. TELNET negotiated options allow a TELNET server to provide services that enhance the basic features available in a Network Virtual Terminal (NVT). Negotiated options utilize the "DO, DON'T, WILL, WON'T" conversation mechanism that allows either party (or both) to initiate an option request, accept or request an option request, and implement the requested option.  
AThread is the peer thread generating the option negotiation event.  
TIdTelnetServer *↗TIdTelnetServer* uses the OnNegotiate event handler to respond to TIdTelnetNegotiateEvent event notifications for connections.

### 3.4.134. TIdTFTPMode

Specifies the transfer mode for data on TFTP connection.

```
TIdTFTPMode = (tfNetAscii, tfOctet);
```

**Description**

TIdTFTPMode is an enumerated type that specifies the transfer mode for data on TFTP

connection, and can contain the values tfNetAscii and tfOctet.  
tfNetAscii indicates that the connection will transmit data using the netascii, or "USA Standard Code for Information Interchange", format defined in the Telnet protocol specification. tfNetAscii will contain 8-bit ASCII data.  
tfOctet indicates that the connection will transmit raw 8-bit data, or binary data.

### 3.4.135. TIdThreadClass

Class type for new thread instances.

```
TIdThreadClass = class of TIdThread ↗TIdThread;
```

**Description**

TIdThreadClass is a TIdThread *↗TIdThread* class type used by TIdThreadMgr *↗TIdThreadMgr* and descendants to represent the class type instance created by the thread manager using CreateNewThread.

### 3.4.136. TIdThreadStopMode

Identifies the manner used to halt a thread.

```
TIdThreadStopMode = (smTerminate, smSuspend);
```

**Description**

TIdThreadStopMode is an enumerated type that identifies the valid values for TIdThread.StopMode, and indicates the manner used to halt execution of TIdThread.Run.  
TIdThreadStopMode can contain one of the following values:

- smTerminate - Thread was terminated.
- smSuspend - Thread was suspended.

### 3.4.137. TIdVCardEMailType

Describes a VCard e-mail address type.

```
TIdVCardEMailType = (ematAOL, ematAppleLink, ematATT, ematCIS,
emateWorld, ematInternet, ematIBMMail, ematMCIMail, ematPowerShare,
ematProdigy, ematTelex, ematX400);
```

**Description**

TIdVCardEMailType is used to indicate a type of E-Mail address and can be one of these values:

- ematAOL - America On-Line
- ematAppleLink - AppleLink
- ematATT - AT&T Mail
- ematCIS - CompuServe Information Service
- emateWorld - eWorld
- ematInternet -Internet SMTP
- ematIBMMail - IBM Mail
- ematMCIMail - MCI Mail
- ematPowerShare - PowerShare
- ematProdigy - Prodigy Information Service
- ematTelex - Telex number
- ematX400 - X.400 service

### 3.4.138. TInAddr

```
TInAddr = in_addr;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.139. TInet\_AddrProc

```
TInet_AddrProc = function (cp: PChar): u_long u_long;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.140. TInet\_NtoaProc

```
TInet_NtoaProc = function (inaddr: TInAddr): PChar;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.141. TIntStringEvent

Event type for coders that allow Integer and String event parameters.

```
TIntStringEvent = procedure (ASender: TComponent; AVal : Integer;
const AOut: String) of object;
```

**Description**

TIntStringEvent is am event type used for coder event notifications that allow both the Integer result code in AVal and the string content in AOut.

### 3.4.142. TIoctlSocketProc

```
TIoctlSocketProc = function (s: TSocket TSocket; cmd: DWORD; var
arg: u_long): Integer;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.143. TListenProc

```
TListenProc = function (s: TSocket TSocket; backlog: Integer):
Integer;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.144. TLogItemEvent

Event type triggered during message logging.

```
TLogItemEvent = procedure (ASender: TComponent; var AText: string) of object;
```

#### Description

TLogItemEvent is an event type triggered when messages are written using the Log method. In TIdLogDebug *↗TIdLogDebug*, the TIdLogEvent can be used to updated GUI applications when one of the logged operations from TIdLogBase *↗TIdLogBase* has occurred. Assign an event handler to OnLogItem to capture and process TLogEventItem events.

### 3.4.145. TModeSetResult

Indicates the result from a an NNTP mode change.

```
TModeSetResult = (mrCanStream, mrNoStream, mrCanIHAVE, mrNoIHAVE, mrCanPost, mrNoPost);
```

#### Description

TModeSetResult is used to indicated the result from a request to change the mode of an NNTP connection. TModeSetResult values are generally updated in methods that establish an NNTP connection, or prepare a connection for transmission of articles.

Valid result values for a change to NNTP mode include the following:

- mrCanStream - Connection can send multiple articles.
- mrNoStream - Connection cannot send multiple articles.
- mrCanIHAVE - Connection can perform negotiated article exchange.
- mrNoIHAVE - Connection cannot perform negotiated article exchange.
- mrCanPost - Connection can post new articles.
- mrNoPost - Connection cannot post new articles.

TModeType *↗TModeType* determines the range of valid TModeSetResult values that can apply to the NNTP connection. For example:

- mtStream - mrCanStream, mrNoStream

- mtIHAVE - mrCanIHAVE, mrNoIHAVE
- mtReader - mrCanPost, mrNoPost

### 3.4.146. TModeType

Indicates the NNTP connection mode for an NNTP connection.

```
TModeType = (mtStream, mtIHAVE, mtReader);
```

#### Description

TModeType indicates the NNTP connection mode to be used by an NNTP client or server. Valid values for TModeType includes the following:

- mtStream
- mtIHAVE
- mtReader

mtStream indicates that the NNTP connection supports a stream of forwarded NNTP messages. mtStream is used to send a group of articles to an NNTP server, or in server to server NNTP news distribution.

mtIHAVE that the NNTP connection wants to perform negotiated article transmissions for articles already posted to an NNTP host. mtIHAVE informs a server that an article is available, and if the server desires a copy of that article, it will return a response instructing the client to send the entire article. If the server does not want the article (if, for example, the server already has a copy of it), a response indicating that the article is not wanted will be returned. mtReader is the NNTP connection mode normally used by NNTP clients (or news readers).

### 3.4.147. TMonths

```
TMonths = (TMthJan, TMthFeb, TMthMar, TMthApr, TMthMay, TMthJun, TMthJul, TMthAug, TMthSep, TMthOct, TMthNov, TMthDec);
```

#### Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.148. TNetEnt

TNetEnt = netent;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.149. TNetworkClass

Specifies valid network class values used by TIdNetworkCalculator *↗TIdNetworkCalculator*.

TNetworkClass = (ID\_NET\_CLASS\_A, ID\_NET\_CLASS\_B, ID\_NET\_CLASS\_C, ID\_NET\_CLASS\_D, ID\_NET\_CLASS\_E);

Description

TNetwork class is an enumerated type that identifies the valid values that can be used in TIdNetworkCalculator *↗TIdNetworkCalculator*. Values in the TNetworkClass enumeration correspond to the network addressing scheme described in the Internet Standards document An IP Address Extension Proposal, RFC 1365 , by K. Siyan.

The 32-bit IP address is defined as having a network part and a local address part. The division between the network part and the local address part has been defined in terms of 5 address classes: class A, B, C, D, E. Only class A, B, and C addresses are assigned to hosts. Class D is used for multicasting. Class E is reserved.

The enumerated values and meanings of TNetworkClass are:

- ID\_NET\_CLASS\_A - Class A has the highest order bit set to 0, a 7 bit network number and a 24 bit host address.
- ID\_NET\_CLASS\_B - Class B has the two higher order bits set to 10, a 14 bit network number and a 16 bit host address.
- ID\_NET\_CLASS\_C - Class C has the three higher order bit set to 110, a 21 bit network number and a 8 bit host address.
- ID\_NET\_CLASS\_D - Class D has the four higher order bits set to 1110.
- ID\_NET\_CLASS\_E- Class E has four higher address bits set to 1111.

3.4.150. TNewsEvent

Specifies the event type for NNTP operations with parameters and return values.

TNewsEvent = **procedure** (AThread: TIdPeerThread *↗TIdPeerThread*;AParm:String) of object;

Description

TNewsEvent is the event type triggered for NNTP Server operations that accept one or more parameters, and return one or more values as a result of the NNTP command.

AThread is the thread of execution performing the NNTP command that triggered the event.

AParm is the string containing an optional list of parameters for the NNTP command.

3.4.151. TNewsTransportEvent

Event triggered during article receipt or transmission.

TNewsTransportEvent = **procedure** (AMsg: TStringList) of object;

Description

TNewsTransportEvent is an event triggered during news during receipt or transmission of one or more NNTP news articles. TNewsTransportEvent provides a mechanism to override the default handling of article transmission in methods like SendCheck, SendIHAVE and SendTakeThis.

AMsg contains one or more Message IDs, one Message ID per line, to be handled by the event.

**Note:** If a TNewsTransportEvent is assigned for a SendXXX method, it is assumed that all message handling and transport will be assumed by the event handler.

3.4.152. TNtohlProc

TNtohlProc = **function** (netlong: u\_long): u\_long *↗u\_long*;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.153. TNtohsProc

TNtohsProc = **function** (netshort: u\_short): u\_short *↯u\_short*;  
**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.154. TOnGetMessagePartStream

Specifies the event type for reading a message part from a stream.

TOnGetMessagePartStream = **procedure** (AStream: TStream) **of object**;  
**Description**  
TOnGetMessagePartStream is an event type used to signal events that occur while handling a message part that resides in a stream.  
TOnGetMessagePartStream is the event type used by the OnGetMessagePartStream event handler of TIdMessagePart *↯TIdMessagePart*.

3.4.155. TOnReplyEvent

Specifies the event type for receipt of an ICMP reply.

TOnReplyEvent = **procedure** (ASender: TComponent; **const** AReplyStatus: TReplyStatus) **of object**;  
**Description**  
TOnReplyEvent is an event type that allows an allows a TIdIcmpClient *↯TIdIcmpClient* to receive and respond to receipt of an ICMP reply message. The TOnReplyEvent is triggered when TIdIcmpClient *↯TIdIcmpClient* has interpreted the reply status.

3.4.156. TOnSessionEndEvent

Specifies an event type for a terminating HTTP session.

TOnSessionEndEvent = **procedure** (Sender: TIdHTTPSession) **of object**;  
**Description**  
TOnSessionEndEvent is an event type used to signal that an HTTP session will be removed from the Indy session list. TOnSessionEndEvent is used by TIdHTTPServer *↯TIdHTTPServer* and TIdHTTPSessionList *↯TIdHTTPSessionList* to implement maintaining session state for TIdHTTPSessions.

3.4.157. TOnSessionStartEvent

Specifies an event type for a new HTTP session.

TOnSessionStartEvent = **procedure** (Sender: TIdHTTPSession) **of object**;  
**Description**  
TOnSessionStartEvent is an event type used to signal that an HTTP session will be created and added to the Indy session list. TOnSessionStartEvent is used by TIdHTTPServer *↯TIdHTTPServer* and TIdHTTPSessionList *↯TIdHTTPSessionList* to implement maintaining session state for TIdHTTPSessions.

3.4.158. TOnTelnetCommand

Event type that occurs when a TELNET command is received.

TOnTelnetCommand = **procedure** (Sender: TComponent; Status: TTelnetCommand) **of object**;  
**Description**  
TOnTelnetCommand specifies the event type triggered when a TELNET command is received by the TIdTelnet *↯TIdTelnet* client.  
Status is the TTelnetCommand *↯TTelnetCommand* received by the client, and provided to the OnTelnetCommand event handler in the TIdTelnet *↯TIdTelnet* client.

### 3.4.159. TOtherEvent

Specifies the event type for NNTP AUTHINFO and other operations.

```
TOtherEvent = procedure (AThread: TIdPeerThread  
  ⚡TIdPeerThread; ACommand: String; AParm: String; var AHandled: Boolean) of  
object;
```

**Description**

TOtherEvent is the event type triggered for the NNTP Server command AUTHINFO and the default handler for unknown NNTP commands.  
AThread is the thread of execution performing the NNTP command that triggered the event.  
ACommand is the command to be executed.  
AParm is the string containing an optional list of parameters for the NNTP command.  
AHandled indicates if the command was recognized and processed by the event handler.

### 3.4.160. TPasswordEvent

Event type for password callbacks.

```
TPasswordEvent = procedure (var Password: String) of object;
```

**Description**

TPasswordEvent is an event type for password callback functions that receive the password as a String value.

### 3.4.161. TPlusRequestEvent

Gopher+ request event.

```
TPlusRequestEvent = procedure (AThread: TIdPeerThread  
  ⚡TIdPeerThread; ARequest: String; APlusData : String) of object;
```

**Description**

TPlusRequestEvent is the event type used to signal a Gopher request for a Gopher connection.  
APlusData includes the Gopher+ command in the first byte, and additional information such

as a requested View for an item or ASK block data sent from the client.  
TIdGopherServer.OnPlusRequest is the event handler used to allow the application to respond to the event notification.

### 3.4.162. TPosProc

Specifies an integer function type for IndyPos ⚡IndyPos implementation.

```
TPosProc = function (const Substr, S: string): Integer;
```

**Description**

TPosProc is an Integer function type used to specify the implementation for the IndyPos ⚡IndyPos function. TPosProc accepts two constant string arguments used as parameters for the IndyPos ⚡IndyPos implementation.

### 3.4.163. TProceduralEvent

Event type for procedural events.

```
TProceduralEvent = procedure of object;
```

**Description**

TProceduralEvent is an event type used for procedural events.

### 3.4.164. TProtoEnt

```
TProtoEnt = protoent;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.



3.4.165. TRANSMIT\_FILE\_BUFFERS

TRANSMIT\_FILE\_BUFFERS = \_TRANSMIT\_FILE\_BUFFERS;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.166. TRecvFromProc

TRecvFromProc = **function** (s: TSocket *↯*TSocket; **var** Buf; len, flags: Integer; **var** from: TSockAddr *↯*TSockAddr; **var** fromlen: Integer): Integer;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.167. TRecvProc

TRecvProc = **function** (s: TSocket *↯*TSocket; **var** Buf; len, flags: Integer): Integer;

Description

The text for this type has been generated automatically. This means that it is not documented.

3.4.168. TReplyStatusTypes

Identifies values for an ICMP Reply Status.

TReplyStatusTypes = (rsEcho, rsError, rsTimeOut, rsErrorUnreachable, rsErrorTTLExceeded);

Description

TReplyStatusTypes is an enumerated type that identifies the response type values that can be received for an ICMP reply. TReplyStatusTypes can contain one of the following values:

- rsEcho - An Echo was received.
- rsError - An error has occurred.
- rsTimeOut - Timeout occurred before a response was received.

- rsErrorUnreachable - The address for the ICMP message is not available.
- rsErrorTTLExceeded - Time-To-Live exceeded for an ICMP response.

3.4.169. TRequestedRecord

Resource type desired for the requested resource record.

TRequestedRecord = cA *↯*cA ..cStar *↯*cStar;

Description

TRequestedRecord is an enumerated value that represent a valid resource type constant.

3.4.170. TRequestedRecords

Set of requested records for the DNS Query.

TRequestedRecords = **set of** TRequestedRecord *↯*TRequestedRecord;

Description

TRequestedRecords is a Set type that contains TRequestedRecord *↯*TRequestedRecord values for a DNS query.

3.4.171. TRequestEvent

Gopher request event.

TRequestEvent = **procedure** (AThread: TIdPeerThread *↯*TIdPeerThread; ARequest: **String**) **of object**;

Description

TRequestEvent is an event type used to signal a Gopher request from a connection on a Gopher server.

TIdGopherServer.OnRequest is the event handler provided to allow the application to respond to the client request.

3.4.172. TSelectProc

```
TSelectProc = function (nfd: Integer; readfds, writefds, exceptfds:
PFDSets; timeout: PTimeVal): Longint;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.173. TSendMsgEvent

```
TSendMsgEvent = procedure (Thread: TIdPeerThread; var
CustomMsg: String) of object;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.174. TSendMsgEventC

```
TSendMsgEventC = procedure (var CustomMsg: String) of object;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.175. TSendProc

```
TSendProc = function (s: TSocket; var Buf: Integer): Integer;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.176. TSendToProc

```
TSendToProc = function (s: TSocket; var Buf: Integer; var
addrto: TSocketAddr; tolen: Integer): Integer;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.177. TSendTrnEvent

```
TSendTrnEvent = procedure (Thread: TIdPeerThread; var
Header: TIdHeader; var CustomMsg: String) of object;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.178. TSendTrnEventC

```
TSendTrnEventC = procedure (var Header: TIdHeader; var
CustomMsg: String) of object;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.179. TServEnt

```
TServEnt = servent;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

3.4.180. TSetSockOptProc

```
TSetSockOptProc = function (s: TSocket; level, optname:
Integer; optval: PChar; optlen: Integer): Integer;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.181. TShutDownProc

```
TShutDownProc = function (s: TSocket ↗TSocket; how: Integer): Integer;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.182. TSockAddr

```
TSockAddr = sockaddr_in;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.183. TSocket

```
TSocket = u_int ↗u_int;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.184. TSocketProc

```
TSocketProc = function (af, Struct, protocol: Integer): TSocket ↗TSocket;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.185. TSockProto

```
TSockProto = sockproto;
```

**Description**  
The text for this type has been generated automatically. This means that it is not documented.

### 3.4.186. TSocksAuthentication

Specifies the Socks Authentication type.

```
TSocksAuthentication = (saNoAuthentication, saUsernamePassword);
```

**Description**  
TSocksAuthentication is an enumerated type that specifies the type of authentication required for the Socks proxy.  
TSocksAuthentication may contain one of these following values:

- saNoAuthentication - No authentication is required.
- saUsernamePassword - User Name and Password authentication.

### 3.4.187. TSocksVersion

Represents Socks protocol version values.

```
TSocksVersion = (svNoSocks, svSocks4, svSocks4A, svSocks5);
```

**Description**  
TSocksVersion is an enumerated type used to specify the version of the proxy protocol required for a client connection.  
TSocksVersion may contain one of these following values:

- svNoSocks - No socks functionality.
- svSocks4 - SOCKS version 4.0.
- svSocks4A - SOCKS version 4.0A.
- svSocks5 - SOCKS version 5.0.

### 3.4.188. TStringEvent

Coder event type for string notifications.

```
TStringEvent = procedure (ASender: TComponent; const AString: string);
```

**Description**

TStringEvent is an event type used to allow notification of coder events using the string message specified in AOut.

### 3.4.189. TTelnetCommand

Specifies the commands recognized by the TELNET command event handler.

```
TTelnetCommand = (tncNoLocalEcho, tncLocalEcho, tncEcho);
```

**Description**

TTelnetCommand is an enumerated type that identifies the TELNET commands recognized by the OnTelnetCommand event handler in TIdTelnet *↗TIdTelnet* during suboption negotiations. The valid values for TTelnetCommand include the following values:

- tncNoLocalEcho - The NVT will not echo characters locally.
- tncLocalEcho - The NVT will echo characters locally.
- tncEcho - The NVT will allow the remote host to echo characters.

### 3.4.190. TThreadPriority

Identifies thread priority constants.

```
TThreadPriority = (tpIdle, tpLowest, tpLower, tpNormal, tpHigher, tpHighest, pTimeCritical);
```

**Description**

TThreadPriority is an enumerated type that identifies the valid thread priority constant values on the Linux platform.

### 3.4.191. TTimeVal

```
TTimeVal = timeval;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.192. TTnDataAvail

Event type that occurs when data is received by the TELNET client.

```
TTnDataAvail = procedure (Buffer : String) of object;
```

**Description**

TTnDataAvail is the event type triggered when data is read from the host connection for a TIdTelnet *↗TIdTelnet* client.

Buffer represents the data read from the TIdTelnetReadThread *↗TIdTelnetReadThread* and provided to the OnDataAvailable event handler in TIdTelnet *↗TIdTelnet*.

### 3.4.193. TTnState

Specifies the TELNET client states.

```
TTnState = (tnsDATA, tnsIAC, tnsIAC_SB, tnsIAC_WILL, tnsIAC_DO, tnsIAC_WONT, tnsIAC_DONT, tnsIAC_SBIAC, tnsIAC_SBDATA, tnsSBDATA_IAC);
```

**Description**

TTnState is an enumerated type that identifies the valid states for the TIdTelnet *↗TIdTelnet* client. TTnState is updated in the TELNET client to reflect the nature and use of the data received by the client.

The following values are valid for TTnState:

- tnsDATA - Data follows.
- tnsIAC - Interpret as Command.
- tnsIAC\_SB - Suboption begin.
- tnsIAC\_WILL - Desire to perform or confirm the option.
- tnsIAC\_DO - Request the peer to perform or confirm the option.
- tnsIAC\_WONT - Desire to cease or confirm stopping the option.
- tnsIAC\_DONT - Request the peer to cease or confirm stopping the option.
- tnsIAC\_SBIAC - Suboption Interpret as Command.
- tnsIAC\_SBDATA - Suboption Interpret as Data.
- tnsSBDATA\_IAC - Suboption Data extension to Interpret as Command.

### 3.4.194. TTransfer

Represents transfer encoding values.

```
TTransfer = (bit7, bit8, iso2022jp);
```

**Description**

TTransfer is an enumerated type that represents the valid MIME transfer encoding values for header encoding and decoding operations. TTransfer can contain one of the following values:

- bit7
- bit8
- iso2022jp

### 3.4.195. TTransferCompleteEvent

Specifies the event type for completed TFTP transfers.

```
TTransferCompleteEvent = procedure (Sender: TObject; const Success: Boolean; const PeerInfo: TPeerInfo; AStream: TStream; const WriteOperation: Boolean) of object;
```

**Description**

TTransferCompleteEvent is an event type signalled when a TrivialFTP server transfer operation has been completed. TTransferCompleteEvent allows the TFTP server to perform operations that may be required for the peer connection or data received on the connection.

### 3.4.196. TTransmitFileBuffers

```
TTransmitFileBuffers = _TRANSMIT_FILE_BUFFERS;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.197. TTransmitFileProc

```
TTransmitFileProc = function (hSocket: TSocket ⚡TSocket; hFile: THandle; nNumberOfBytesToWrite: DWORD; nNumberOfBytesPerSend: DWORD; lpOverlapped: POverlapped; lpTransmitBuffers: PTransmitFileBuffers ⚡PTransmitFileBuffers; dwReserved: DWORD): BOOL;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.198. TTunnelEvent

Specified the event type for slave thread notifications.

```
TTunnelEvent = procedure (Thread: TSlaveThread) of object;
```

**Description**

TTunnelEvent is an event type that identifies the TSlaveThread *⚡TSlaveThread* affected by the event notification. TTunnelEvent is the event type generated for the OnTransformRead and OnTunnelDisconnect event handlers.

### 3.4.199. TTunnelEventC

```
TTunnelEventC = procedure (Receiver: TReceiver) of object;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

### 3.4.200. TUDPReadEvent

Specifies the event type for reading data from a UDP socket binding.

```
TUDPReadEvent = procedure (Sender: TObject; AData: TStream; ABinding: TIdSocketHandle) of object;
```

**Description**

TUDPReadEvent is an event type used to notify a TIdUDPServer *⚡TIdUDPServer* event

handler that data has been received from a socket binding and is available in the buffer. TIdUDPServer *↗*TIdUDPServer provides the DoUDPRead event handler to respond to TUDPReadEvent notifications.

3.4.201. TVerifyPeerEvent

X.509 Certificate verification event type.

TVerifyPeerEvent = **function** (Certificate: TIdX509): Boolean **of object**;  
**Description**  
TVerifyPeerEvent is an event type that allows respond to SSL verification request using an X.509 certificate.

3.4.202. TWKSBits

Represents flags for Well Known Service entries.

TWKSBits = **array**[0..7] **of** byte;  
**Description**  
TWKSBits is an Array of Byte values used to represent bit flags for Well Known Service entries.

3.4.203. TWorkBeginEvent

Indicates a pending read or write operation.

TWorkBeginEvent = **procedure** (Sender: TObject; AWorkMode: TWorkMode *↗*TWorkMode; **const** AWorkCountMax: Integer) **of object**;  
**Description**  
TWorkBeginEvent is the event used to indicate that the component is about to read data from the peer or send data to the peer.  
AWorkMode is the TWorkMode *↗*TWorkMode value that indicate the operation to be performed.  
AWorkCountMax is the number of bytes expected for this operation, or 0 when no value is

provided.

3.4.204. TWorkEndEvent

Indicates that a read or write operation is complete.

TWorkEndEvent = **procedure** (Sender: TObject; AWorkMode: TWorkMode) **of object**;  
**Description**  
This type of event occurs in components when receiving or sending data has been completed. AWorkMode is a TWorkMode *↗*TWorkMode value that indicates if the component is receiving or sending data.

3.4.205. TWorkEvent

Indicates that the read or write operation is being performed.

TWorkEvent = **procedure** (Sender: TObject; AWorkMode: TWorkMode *↗*TWorkMode; **const** AWorkCount: Integer) **of object**;  
**Description**  
This event often fires when a component is receiving or sending data and this event is often used to update progress indicators.  
AWorkMode indicates if the component is receiving or sending data and can be one of two values:

- wmRead - The component is reading data from the peer.
- wmWrite - The component is sending data to the peer.

AWorkCount indicates the number of bytes sent or received.

3.4.206. TWorkMode

Indicates the work mode or operation.

```
TWorkMode = (wmRead, wmWrite);
```

**Description**

TWorkMode is an enumerated type that indicates the work mode or operation, and is used in the TWorkBeginEvent *↗TWorkBeginEvent*, TWorkEvent *↗TWorkEvent*, and TWorkEndEvent *↗TWorkEndEvent* events. TWorkMode can contain the values:

- wmRead - The component is reading data from the peer.
- wmWrite - The component is sending data to the peer.

3.4.207. TWSAAsyncGetHostByAddrProc

```
TWSAAsyncGetHostByAddrProc = function (HWindow: HWND; wMsg: u_int  
↗u_int; addr: PChar; len, Struct: Integer; buf: PChar; buflen:  
Integer): THandle;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.208. TWSAAsyncGetHostByNameProc

```
TWSAAsyncGetHostByNameProc = function (HWindow: HWND; wMsg: u_int  
↗u_int; name, buf: PChar; buflen: Integer): THandle;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.209. TWSAAsyncGetProtoByNameProc

```
TWSAAsyncGetProtoByNameProc = function (HWindow: HWND; wMsg: u_int  
↗u_int; name, buf: PChar; buflen: Integer): THandle;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.210. TWSAAsyncGetProtoByNumberProc

```
TWSAAsyncGetProtoByNumberProc = function (HWindow: HWND; wMsg: u_int  
↗u_int; number: Integer; buf: PChar; buflen: Integer): THandle;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.211. TWSAAsyncGetServByNameProc

```
TWSAAsyncGetServByNameProc = function (HWindow: HWND; wMsg: u_int  
↗u_int; name, proto, buf: PChar; buflen: Integer): THandle;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.212. TWSAAsyncGetServByPortProc

```
TWSAAsyncGetServByPortProc = function ( HWindow: HWND; wMsg, port:  
u_int ↗u_int; proto, buf: PChar; buflen: Integer): THandle;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.213. TWSAAsyncSelectProc

```
TWSAAsyncSelectProc = function (s: TSocket ↗TSocket; HWindow: HWND;  
wMsg: u_int ↗u_int; lEvent: Longint): Integer;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.214. TWSACancelAsyncRequestProc

```
TWSACancelAsyncRequestProc = function (hAsyncTaskHandle: THandle):  
Integer;
```

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.215. TWSACancelBlockingCallProc

TWSACancelBlockingCallProc = **function** : Integer;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.216. TWSACleanupProc

TWSACleanupProc = **function** : Integer;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.217. TWSAData

TWSAData = WSADData;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.218. TWSAGetLastErrorProc

TWSAGetLastErrorProc = **function** : Integer;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.219. TWSAIsBlockingProc

TWSAIsBlockingProc = **function** : BOOL;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.220. TWSARecvExProc

TWSARecvExProc = **function** (s: TSocket ~~TSocket~~; **var** buf; len: Integer; **var** flags: Integer): Integer;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.221. TWSASetBlockingHookProc

TWSASetBlockingHookProc = **function** (lpBlockFunc: TFarProc): TFarProc;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.222. TWSASetLastErrorProc

TWSASetLastErrorProc = **procedure** (iError: Integer);

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.223. TWSAStartupProc

TWSAStartupProc = **function** (wVersionRequired: word; **var** WSADData: TWSADData): Integer;

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.224. TWSAUnhookBlockingHookProc

TWSAUnhookBlockingHookProc = **function** : Integer;



**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.225. **u\_char**

`u_char = Char;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.226. **u\_int**

`u_int = Integer;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.227. **u\_long**

`u_long = DWORD;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.228. **u\_short**

`u_short = Word;`

**Description**

The text for this type has been generated automatically. This means that it is not documented.

3.4.229. **WordStr**

Specifies the data type for WordToStr *↯ WordToStr* conversions.

`WordStr = string[2];`

**Description**

WordStr specifies the data type for WordToStr *↯ WordToStr* conversions.

3.5. **Variables**

3.5.1. **\_\_WSAFDIsSet**

`__WSAFDIsSet: T__WSAFDIsSetProc ↯ T__WSAFDIsSetProc = nil;`

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

3.5.2. **Accept**

`Accept: TAcceptProc ↯ TAcceptProc = nil;`

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

3.5.3. **AcceptEx**

`AcceptEx: TAcceptExProc ↯ TAcceptExProc = nil;`

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

3.5.4. **Bind**

`Bind: TBindProc ↯ TBindProc = nil;`

**Description**

The text for this variable has been generated automatically. This means that it is not

documented.

3.5.5. CloseSocket

CloseSocket: TCloseSocketProc = nil;

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.6. CoderCollective

Represents the Indy coder collection.

CoderCollective: TIdCoderCollection *⚡TIdCoderCollection*;

Description

CoderCollective is a TIdCoderCollection *⚡TIdCoderCollection* unit variable that represents the Indy coder collection. CoderCollective is pre-loaded with a CT\_CREATION and a CT\_REALISATION coder class instance in the initialization section of the unit. CoderCollective is also update in calls to RegisterCoderClass *⚡RegisterCoderClass*.

3.5.7. Connect

Connect: TConnectProc *⚡TConnectProc* = nil;

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.8. GAntiFreeze

Application global variable used to the AntiFreeze instance.

GAntiFreeze: TIdAntiFreezeBase *⚡TIdAntiFreezeBase* = nil;

Description

TIdAntiFreezeBase *⚡TIdAntiFreezeBase* uses the global variable GAntiFreeze, declared in the TIdAntiFreezeBase *⚡TIdAntiFreezeBase* unit, to determine if another instance has already been created. Only one instance of TIdAntiFreezeBase *⚡TIdAntiFreezeBase* is allowed per application.

GAntiFreeze is initialized in the Create constructor of the first instance of TAntiFreezeBase. GAntiFreeze is set to nil in the destructor of TAntiFreezeBase. Refer to TIdAntiFreezebase.Destroy for more information.

3.5.9. GetAcceptExSockaddrs

GetAcceptExSockaddrs: TGetAcceptExSockaddrsProc

*⚡TGetAcceptExSockaddrsProc* = nil;

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.10. GetHostByAddr

GetHostByAddr: TGetHostByAddrProc *⚡TGetHostByAddrProc* = nil;

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.11. GetHostByName

GetHostByName: TGetHostByNameProc *⚡TGetHostByNameProc* = nil;

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.12. GetHostName

```
GetHostName: TGetHostNameProc ↗TGetHostNameProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

3.5.13. GetPeerName

```
GetPeerName: TGetPeerNameProc ↗TGetPeerNameProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

3.5.14. GetProtoByname

```
GetProtoByname: TGetProtoByNameProc ↗TGetProtoByNameProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

3.5.15. GetProtoByNumber

```
GetProtoByNumber: TGetProtoByNumberProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

3.5.16. GetServByName

```
GetServByName: TGetServByNameProc ↗TGetServByNameProc = nil;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

3.5.17. GetServByPort

```
GetServByPort: TGetServByPortProc ↗TGetServByPortProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

3.5.18. GetSockName

```
GetSockName: TGetSockNameProc ↗TGetSockNameProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

3.5.19. GetSockOpt

```
GetSockOpt: TGetSockOptProc ↗TGetSockOptProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

3.5.20. GOffsetFromUTC

Time difference from UTC time coordinates.

```
GOffsetFromUTC: TDateTime = 0;
```

**Description**  
GOffsetFromUTC is a TDateTime constant variable that represents the hours and minutes difference from Universal Time Coordinates (UTC). **Note:** GOffsetFromUTC is defined for the Linux platform only.

### 3.5.21. GServeFileProc

Indy optimized file transfer routine for Windows NT.

```
GServeFileProc: TIdServeFile ⚡TIdServeFile = nil;
```

**Description**

GServeFileProc is a Cardinal function that represents the routine used to perform optimized file transmission under Windows NT. GServeFile uses the ServeFile function under Windows NT that is optimized for sequential read-only access. GServeFile is the application global instance of the function for Indy.

### 3.5.22. GStack

The Indy global stack instance.

```
GStack: TIdStack ⚡TIdStack = nil;
```

**Description**

GStack is a global variable that represents the current stack object for the Indy component set. GStack should never be changed, and is created and destroyed as needed. This is mentioned only because there a few useful low-level properties and methods.

### 3.5.23. GSystemLocale

Character for the system locale.

```
GSystemLocale: TCharSet ⚡TCharSet = csIso88591;
```

**Description**

GSystemLocale is a constant variable that represenents the system locale character set. The default for GSystemLocale is cscsIso88591 on the Linux platform. **Note:** GSystemLocale is defined for the Linux platform only.

### 3.5.24. GTimeZoneBias

Daylight saving time adjustment for the current time zone.

```
GTimeZoneBias: Double = 0;
```

**Description**

GTimeZoneBias is a Double constant variable that represents the time adjustment for daylight savings time for the current time zone. **Note:** GTimeZoneBias is defined for the Linux platform only.

### 3.5.25. Htonl

```
Htonl: THtonlProc ⚡THtonlProc = nil;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.26. Htons

```
Htons: THtonsProc ⚡THtonsProc = nil;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.27. Id\_SO\_False

```
Id_SO_False: Integer = 0;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.28. Id\_SO\_True

```
Id_SO_True: Integer = 1;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.29. IndyPos

Specifies the Pos function handler for single- or multi-byte character sets.

```
IndyPos: TPosProc ↗TPosProc = nil;
```

**Description**

IndyPos is a unit global TPosProc *↗*TPosProc variable that determines the function used for single-byte and multi-byte equivalents of the RTL Pos function. IndyPos is assigned the initialization section of the IdGlobal.pas *↗*IdGlobal.pas unit, and will contain a pointer to the function used for run-time support.

For single-byte character sets, where the leading byte is #0 (Decimal 0), IndyPos will use the address of SBPos. SBPos implements the Pos function hidden by Object Pascal "compiler magic".

For multi-byte character sets, IndyPos will use the address of AnsiPos.

### 3.5.30. Inet\_Addr

```
Inet_Addr: TInet_AddrProc ↗TInet_AddrProc = nil;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.31. Inet\_Ntoa

```
Inet_Ntoa: TInet_NtoaProc ↗TInet_NtoaProc = nil;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.32. IoctlSocket

```
IoctlSocket: TIoctlSocketProc ↗TIoctlSocketProc = nil;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.33. Listen

```
Listen: TListenProc ↗TListenProc = nil;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.34. MIMEMediaType

```
MIMEMediaType: array [0..MaxMIMEMType] of String;
```

**Description**

MIMEMediaType represents the array of Strings constructed for valid MIME media type, subtype and encoding constant combinations.

### 3.5.35. Ntohl

```
Ntohl: TNtohlProc ↗TNtohlProc = nil;
```

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

3.5.36. Ntohs

```
Ntohs: TNtohsProc ↪TNtohsProc = nil;
```

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.37. Recv

```
Recv: TRecvProc ↪TRecvProc = nil;
```

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.38. RecvFrom

```
RecvFrom: TRecvFromProc ↪TRecvFromProc = nil;
```

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.39. Select

```
Select: TSelectProc ↪TSelectProc = nil;
```

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.40. Send

```
Send: TSendProc ↪TSendProc = nil;
```

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.41. SendTo

```
SendTo: TSendToProc ↪TSendToProc = nil;
```

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.42. SetSockOpt

```
SetSockOpt: TSetSockOptProc ↪TSetSockOptProc = nil;
```

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.43. ShutDown

```
ShutDown: TShutDownProc ↪TShutDownProc = nil;
```

Description

The text for this variable has been generated automatically. This means that it is not documented.

3.5.44. Socket

```
Socket: TSocketProc ↪TSocketProc = nil;
```

Description

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.45. TransmitFile

TransmitFile: TTransmitFileProc *⚡TTransmitFileProc* = **nil**;

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.46. WSAAsyncGetHostByAddr

WSAAsyncGetHostByAddr: TWSAAsyncGetHostByAddrProc  
*⚡TWSAAsyncGetHostByAddrProc* = **nil**;

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.47. WSAAsyncGetHostByName

WSAAsyncGetHostByName: TWSAAsyncGetHostByNameProc  
*⚡TWSAAsyncGetHostByNameProc* = **nil**;

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.48. WSAAsyncGetProtoByName

WSAAsyncGetProtoByName: TWSAAsyncGetProtoByNameProc  
*⚡TWSAAsyncGetProtoByNameProc* = **nil**;

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.49. WSAAsyncGetProtoByNumber

WSAAsyncGetProtoByNumber: TWSAAsyncGetProtoByNumberProc  
*⚡TWSAAsyncGetProtoByNumberProc* = **nil**;

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.50. WSAAsyncGetServByName

WSAAsyncGetServByName: TWSAAsyncGetServByNameProc  
*⚡TWSAAsyncGetServByNameProc* = **nil**;

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.51. WSAAsyncGetServByPort

WSAAsyncGetServByPort: TWSAAsyncGetServByPortProc  
*⚡TWSAAsyncGetServByPortProc* = **nil**;

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.52. WSAAsyncSelect

WSAAsyncSelect: TWSAAsyncSelectProc *⚡TWSAAsyncSelectProc* = **nil**;

**Description**

The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.53. WSACancelAsyncRequest

```
WSACancelAsyncRequest: TWSACancelAsyncRequestProc  
⚡TWSACancelAsyncRequestProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.54. WSACancelBlockingCall

```
WSACancelBlockingCall: TWSACancelBlockingCallProc  
⚡TWSACancelBlockingCallProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.55. WSACleanup

```
WSACleanup: TWSACleanupProc ⚡TWSACleanupProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.56. WSAGetLastError

```
WSAGetLastError: TWSAGetLastErrorProc ⚡TWSAGetLastErrorProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.57. WSAIsBlocking

```
WSAIsBlocking: TWSAIsBlockingProc ⚡TWSAIsBlockingProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.58. WSARecvEx

```
WSARecvEx: TWSARecvExProc ⚡TWSARecvExProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.59. WSASetBlockingHook

```
WSASetBlockingHook: TWSASetBlockingHookProc ⚡TWSASetBlockingHookProc  
= nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

### 3.5.60. WSASetLastError

```
WSASetLastError: TWSASetLastErrorProc ⚡TWSASetLastErrorProc = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.



3.5.61. WSAStartup

```
WSAStartup: TWSAStartupProc ↗TWSAStartupProc  = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

3.5.62. WSAUnhookBlockingHook

```
WSAUnhookBlockingHook: TWSAUnhookBlockingHookProc  
↗TWSAUnhookBlockingHookProc  = nil;
```

**Description**  
The text for this variable has been generated automatically. This means that it is not documented.

3.6. Constants

3.6.1. AF\_APPLETALK

```
AF_APPLETALK = 16;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.2. AF\_BAN

```
AF_BAN = 21;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.3. AF\_CCITT

```
AF_CCITT = 10;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.4. AF\_CHAOS

```
AF_CHAOS = 5;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.5. AF\_DATAKIT

```
AF_DATAKIT = 9;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.6. AF\_DECnet

```
AF_DECnet = 12;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.7. AF\_DLI

```
AF_DLI = 13;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.8. AF\_ECMA

AF\_ECMA = 8;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.9. AF\_FIREFOX

AF\_FIREFOX = 19;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.10. AF\_HYLINK

AF\_HYLINK = 15;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.11. AF\_IMPLINK

AF\_IMPLINK = 3;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.12. AF\_INET

AF\_INET = 2;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.13. AF\_IPX

AF\_IPX = 6;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.14. AF\_ISO

AF\_ISO = 7;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.15. AF\_LAT

AF\_LAT = 14;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.16. AF\_MAX

AF\_MAX = 22;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.17. AF\_NETBIOS

AF\_NETBIOS = 17;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.18. AF\_NS

AF\_NS = 6;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.19. AF\_OSI

AF\_OSI = AF\_ISO ~~≠~~AF\_ISO;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.20. AF\_PUP

AF\_PUP = 4;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.21. AF\_SNA

AF\_SNA = 11;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.22. AF\_UNIX

AF\_UNIX = 1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.23. AF\_UNKNOWN1

AF\_UNKNOWN1 = 20;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.24. AF\_UNSPEC

AF\_UNSPEC = 0;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.25. AF\_VOICEVIEW

AF\_VOICEVIEW = 18;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.26. BACKSPACE

Represents the US-ASCII Backspace character.

BACKSPACE = #8;

Description

BACKSPACE is the constant Integer value used to represent the US-ASCII Backspace character.

3.6.27. base64\_tbl

Represents the Base64 alphabet.

base64\_tbl: **array** [0..63] **of** Char = (  
'A','B','C','D','E','F','G','H', 'I','J','K','L','M','N','O','P',  
'Q','R','S','T','U','V','W','X', 'Y','Z','a','b','c','d','e','f',  
'g','h','i','j','k','l','m','n', 'o','p','q','r','s','t','u','v',  
'w','x','y','z','0','1','2','3', '4','5','6','7','8','9','+','/');

Description

base64\_tbl is an Array of Char type that represents the Base64 alphabet used in Base64 encoding and decoding operations. base64\_tbl is used in B64 and EncodeHeader *↗EncodeHeader*.

3.6.28. Base64CodeTable

Represents the character encoding table for Base64 encoding.

Base64CodeTable: **string** =  
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/';

Description

Base64CodeTable is a constant String value that represents the character encoding table for

Base64 encoding. Base64CodeTable is used by TldBase64Encoder *↗TldBase64Encoder* and TldBase64Decoder *↗TldBase64Decoder*.

3.6.29. BUFFERLEN

Default buffer length used by Tunnel communication classes.

BUFFERLEN = \$4000;

Description

Default buffer length used by Tunnel communication classes.

3.6.30. BytesReadType

Number of bytes read for the Tunnel component.

BytesReadType = 5;

Description

BytesReadType is a constant value that represents the statistical operation for updating the number of bytes read for the Tunnel component.

3.6.31. BytesWriteType

Number of bytes written for the Tunnel component.

BytesWriteType = 6;

Description

BytesWriteType is a constant value that represents the statistical operation for updating the number of bytes written for the Tunnel component.

3.6.32. cA

Host Address resource type.

cA = 1;

Description

cA is a constant value used in TIdDNSResolver *TIdDNSResolver* for requesting a simple Host Address. Thus, a query for wwnvm.wvnet.edu will result in the IP address 129.71.2.4 in the TIdDNSResourceItem.RData.HostAddrStr property.

3.6.33. cAABit

Valid in Responses Authoritative Answer bit mask.

cAABit = \$0400;

Description

cAABit is a constant value that represents a bit mask used to access the Valid in Responses Authoritative Answer bit.

3.6.34. cAAMask

Valid in Responses Authoritative Answer result mask.

cAAMask = \$FBFF;

Description

cAAMask is a constant value that represents a mask used to determine the Valid in Responses Authoritative Answer result value.

3.6.35. cAXFR

Question type for entire zone transfer.

cAXFR = 252;

Description

cAXFR is a constant value that represents the question type for a request for the Transfer of an entire zone.

3.6.36. cCH

Resource class.

cCH = 3;

Description

cCH is a constant value that represents the resource class for "The Chaos Claee".

3.6.37. cCS

Resource class constant.

cCS = 2;

Description

cCS is a constant value that represents the resource class for CSNet (Obsolete).

3.6.38. CHAR0

Represents the Null character.

CHAR0 = #0;

Description

CHAR0 is a constant value used to represent the Null character.

3.6.39. CHAR32

Represents the Space character.

CHAR32 = #32;

Description

CHAR32 is the constant value used to represent the Space character.

3.6.40. cHINFO

Resource type constant.

cHINFO = 13;

Description

cHINFO is a constant value that represents the resource type for Host Information.

3.6.41. cHS

Resource class constant.

cHS = 4;

Description

cHS is a constant value that represents the resource class for Hesiod [Dyer 87].

3.6.42. cIN

Resource class constant.

cIN = 1;

Description

cIN is a constant value that represents the resource class for Internet resource records.

3.6.43. cMAILA

Resource type constant.

cMAILA = 254;

Description

cMAILA is a constant value that represents the resource type for mail agent resource records.  
(Obsoleted by cMX)

3.6.44. cMAILB

Resource type constant.

cMAILB = 253;

Description

cMAILB is a constant value that represents the resource type for mailbox related records.

3.6.45. cMB

Resource type constant.

cMB = 7;

Description

cMB is a constant value that represents the resource type for a mail box domain name.(Experimental)

3.6.46. cMD

Resource type constant.

cMD = 3;

**Description**

cMD is a constant value that represents the resource type for a mail destination. (obsoleted by cMX)

**3.6.47. cMF**

Resource type constant.

cMF = 4;

**Description**

cMF is a constant value that represents the resource type for a mail forwarder. (obsoleted by cMX)

**3.6.48. cMG**

Resource type constant.

cMG = 8;

**Description**

cMG is a constant value that represents the resource type for a mail group member. (Experimental)

**3.6.49. cMINFO**

Resource type constant.

cMINFO = 14;

**Description**

cMINFO is a constant value that represents the resource type for Mailbox or Mail List Information.

**3.6.50. cMR**

Resource type constant.

cMR = 9;

**Description**

cMR is a constant value that represents the resource type for a mail Rename Domain Name. (Experimental)

**3.6.51. cMX**

Resource type constant.

cMX = 15;

**Description**

cMX is a constant value that represents the resource type for a Mail Exchange.

**3.6.52. CN\_CODED\_DATA**

Coder notification message for generation of encoded data.

CN\_CODED\_DATA = 0;

**Description**

CN\_CODED\_DATA is a constant value that represents the generic coder notification message that signals output generation of encoded data.

**3.6.53. CN\_CODING\_ENDED**

Coder notification message for completion of coding.

CN\_CODING\_ENDED = 4;

**Description**

CN\_CODING\_ENDED is a constant value that represents the generic coder notification message that signals completion of a coding operation.

3.6.54. CN\_CODING\_STARTED

Coder notification message for execution of coding.

CN\_CODING\_STARTED = 3;

Description

CN\_CODING\_STARTED is a constant value that represents the generic coder notification message that signals execution of a coding operation.

3.6.55. CN\_DATA\_END\_FOUND

Coder notification message for completion of input data.

CN\_DATA\_END\_FOUND = 2;

Description

CN\_CODING\_STARTED ~~CN\_CODING\_STARTED~~ is a constant value that represents the generic coder notification message for completion of coder input data.

3.6.56. CN\_DATA\_START\_FOUND

Coder notification message for initial input data.

CN\_DATA\_START\_FOUND = 1;

Description

CN\_DATA\_START\_FOUND is a constant value that represents the generic coder notification message for initial processing of coder input data.

3.6.57. CN\_IMF\_BODY\_PART\_END

Coder notification for when an IMF body has been detected.

CN\_IMF\_BODY\_PART\_END = CN\_IMF\_CODER\_START ~~CN\_IMF\_CODER\_START~~ + 2;

Description

CN\_IMF\_BODY\_PART\_END is a constant value that represents the coder notification message issued when an Internet Message Format coder has located the end of the message body.

3.6.58. CN\_IMF\_BODY\_START

Coder notification for when the start of an IMF body has been detected.

CN\_IMF\_BODY\_START = CN\_IMF\_CODER\_START ~~CN\_IMF\_CODER\_START~~ + 1;

Description

CN\_IMF\_BODY\_START is a constant value that represents the coder notification message issued when an Internet Message Format coder has located the beginning of the message body.

3.6.59. CN\_IMF\_CODER\_START

Initial Coder notification for Internet Message Format coders.

CN\_IMF\_CODER\_START = 20;

Description

CN\_IMF\_CODER\_START is a constant value that represents the initial coder notification message for Internet Message Format coders. CN\_IMF\_CODER\_START is not actually used for IMF coder notifications.



3.6.60. CN\_IMF\_DATA\_END

Coder notification for end of message data.

CN\_IMF\_DATA\_END = CN\_IMF\_CODER\_START *≠*CN\_IMF\_CODER\_START + 6;

Description

CN\_IMF\_DATA\_END is a constant value that represents the coder notification message issued when an Internet Message Format coder has located the end of the data for the message.

3.6.61. CN\_IMF\_END\_MULTIPART

Coder notification for a boundary in a multipart message.

CN\_IMF\_END\_MULTIPART = CN\_IMF\_CODER\_START *≠*CN\_IMF\_CODER\_START + 5;

Description

CN\_IMF\_END\_MULTIPART is a constant value that represents the coder notification message when a boundary is encountered in a multipart message.

3.6.62. CN\_IMF\_HEAD\_VALUE

Coder notification for header values in message data.

CN\_IMF\_HEAD\_VALUE = CN\_IMF\_CODER\_START *≠*CN\_IMF\_CODER\_START + 3;

Description

CN\_IMF\_HEAD\_VALUE is a constant value that represents the coder notification message issued when an Internet Message Format coder has located a header value in the message.

3.6.63. CN\_IMF\_NEW\_FILENAME

Coder notification for a file name from the message.

CN\_IMF\_NEW\_FILENAME = CN\_NEW\_FILENAME *≠*CN\_NEW\_FILENAME;

Description

CN\_IMF\_NEW\_FILENAME is a constant value that represents the coder notification message issued when an Internet Message Format coder has located a file name in the message.

3.6.64. CN\_IMF\_NEW\_MULTIPART

Coder notification for boundary markers in a message.

CN\_IMF\_NEW\_MULTIPART = CN\_IMF\_CODER\_START *≠*CN\_IMF\_CODER\_START + 4;

Description

CN\_IMF\_NEW\_MULTIPART is a constant value that represents the coder notification message issued when an Internet Message Format coder has located a boundary marker in a multi-part message.

3.6.65. CN\_NEW\_FILENAME

Coder notification for a file name from input data.

CN\_NEW\_FILENAME = 5;

Description

CN\_NEW\_FILENAME is a constant value that represents the generic coder notification message issued when a coder has located a file name in the input data.

3.6.66. CN\_UU\_BEGIN\_FOUND

Coder notification for a UUEncode header in data.

CN\_UU\_BEGIN\_FOUND = CN\_UU\_CODER\_START *≠*CN\_UU\_CODER\_START + 2;

Description

CN\_UU\_BEGIN\_FOUND is a constant value that represents the UUEncoder notification message issued when UUEncode Header is located in data.

### 3.6.67. CN\_UU\_CODER\_START

Initial notification message for UUEncode coders.

CN\_UU\_CODER\_START = 40;

**Description**

CN\_UU\_CODER\_START is a constant value that represents the initial notification message for UUEncode coders.

### 3.6.68. CN\_UU\_END\_FOUND

Coder notification message for the end of the UUEncode header.

CN\_UU\_END\_FOUND = CN\_UU\_CODER\_START *⌘*CN\_UU\_CODER\_START + 5;

**Description**

CN\_UU\_END\_FOUND is a constant value that represents the notification message for UUEncode coders indicating the end of the UUEncode header.

### 3.6.69. CN\_UU\_LAST\_CHAR\_FOUND

Coder notification message for the start of possible UUEncode padding.

CN\_UU\_LAST\_CHAR\_FOUND = CN\_UU\_CODER\_START *⌘*CN\_UU\_CODER\_START + 4;

**Description**

CN\_UU\_LAST\_CHAR\_FOUND is a constant value that represents the notification message for UUEncode coders indicating that all input data has been processed, and UUEncode padding needs to be calculated from the last character position.

### 3.6.70. CN\_UU\_NEW\_FILENAME

Coder notification for a file name from data.

CN\_UU\_NEW\_FILENAME = CN\_NEW\_FILENAME *⌘*CN\_NEW\_FILENAME;

**Description**

CN\_UU\_NEW\_FILENAME is a constant value that represents the coder notification message issued when a UUEncode coder has located a file name in the data.

### 3.6.71. CN\_UU\_PRIVILEGE\_ERROR

Coder notification for ioncorrect privilege for the coder.

CN\_UU\_PRIVILEGE\_ERROR = CN\_UU\_CODER\_START *⌘*CN\_UU\_CODER\_START + 8;

**Description**

CN\_UU\_PRIVILEGE\_ERROR is a constant value that represents the UUEncode coder notification message issued when the UUEncode coder does not have the necessary UNIX privileges identified in the UUEncode header.

### 3.6.72. CN\_UU\_PRIVILEGE\_FOUND

Coder notification for UNIX privileges from the UUEncode header.

CN\_UU\_PRIVILEGE\_FOUND = CN\_UU\_CODER\_START *⌘*CN\_UU\_CODER\_START + 7;

**Description**

CN\_UU\_PRIVILEGE\_FOUND is a constant value that represents the coder notification message issued when a UUEncode coder encounters the UNIX privilege value in the UUEncode header.

### 3.6.73. CN\_UU\_TABLE\_BEGIN\_ABORT

Coder notificaiton for UUEncode header errors.

CN\_UU\_TABLE\_BEGIN\_ABORT = CN\_UU\_CODER\_START *⌘*CN\_UU\_CODER\_START + 3;

**Description**

CN\_UU\_TABLE\_BEGIN\_ABORT is a constant value that represents the coder notification message issued when a error has occurred while handling UUEncode header data.

### 3.6.74. CN\_UU\_TABLE\_CHANGED

Coder notification for a change to the UUEncode alphabet.

```
CN_UU_TABLE_CHANGED = CN_UU_CODER_START ⚡CN_UU_CODER_START + 6;
```

**Description**  
CN\_UU\_TABLE\_CHANGED is a constant value that represents the coder notification issued when the UUEncode alphabet has been altered in the UUEncode header.

### 3.6.75. CN\_UU\_TABLE\_FOUND

Notification message for a missing UUEncode header.

```
CN_UU_TABLE_FOUND = CN_UU_CODER_START ⚡CN_UU_CODER_START + 1;
```

**Description**  
CN\_UU\_TABLE\_FOUND is a constant value that represents the coder notification issued when the UUEncode header is not found in the data for the coder.

### 3.6.76. cName

Resource type constant.

```
cName = 5;
```

**Description**  
cName is a constant value that represents the resource type for the canonical name for an alias.

### 3.6.77. cNS

Resource type constant.

```
cNS = 2;
```

**Description**  
cNS is a constant value that represents the resource type for an Authoritative name server.

### 3.6.78. cNULL

Resource type constant.

```
cNULL = 10;
```

**Description**  
cNULL is a constant value that represents the resource type for a Resource Record. (Experimental)

### 3.6.79. CompressedBytesType

Total compressed bytes handled by the Tunnel component.

```
CompressedBytesType = 4;
```

**Description**  
CompressedBytesType is a constant value that represents the statistical operation for updating the total number of compressed bytes handled by the Tunnel component.

### 3.6.80. CompressionRatioType

Ratio of total byes to total compressed bytes for the Tunnel component.

```
CompressionRatioType = 3;
```

**Description**  
CompressionRatioType is a constant value that represents the statistical operation for updating the Ratio of total byes to total compressed bytes for the Tunnel component.

### 3.6.81. ConstBoundary

Represents the boundary marker keyword in an IMF message.

```
ConstBoundary = 'boundary';
```

**Description**

ConstBoundary is a constant value that represents the RFC 822-compliant header atom value for the boundary keyword in Internet Mail Format having multipart alternative boundaries.

### 3.6.82. ConstContentDisposition

Represents the content-disposition header name.

```
ConstContentDisposition = 'content-disposition';
```

**Description**

ConstContentDisposition is a constant value that represents the RFC 822-compliant header name for Internet Mail Format having a Content-Disposition header.

### 3.6.83. ConstContentMD5

Represents the header MD5-encoded messages.

```
ConstContentMD5 = 'content-md5';
```

**Description**

ConstContentMD5 is a constant value that represents the RFC 822-compliant header for Internet Mail Format messages using the Message Digest 5 encoding scheme.

### 3.6.84. ConstContentTransferEncoding

Represents the content-transfer-encoding literal in an IMF message.

```
ConstContentTransferEncoding = 'content-transfer-encoding';
```

**Description**

ConstContentTransferEncoding is a constant value that represents the RFC 822-compliant header name for Internet Mail Format messages using a content-transfer-encoding header value.

### 3.6.85. ConstContentType

Represents the header name for content type header in an IMF message.

```
ConstContentType = 'content-type';
```

**Description**

ConstContentType is a constant value that represents the RFC 822-compliant header name for Internet Mail Format messages using a content-type: header value.

### 3.6.86. ConstFileName

Header name for inline content file names.

```
ConstFileName = 'filename';
```

**Description**

ConstFileName is a constant value that represents the RFC 822-compliant header name for Internet Mail Format messages using content-disposition to identify an inline file.

### 3.6.87. ConstIMFBoundaryEnd

State value for the end of a message part in a multipart message.

```
ConstIMFBoundaryEnd = 2;
```

**Description**

ConstIMFBoundaryEnd is a constant value that represents the IIdIMFDecoder state value set when the decoder has reached the end of a message part in a multipart Internet Mail Format message.

3.6.88. ConstIMFMessageStart

State value indicating that processing of an IMF message has started.

```
ConstIMFMessageStart = 1;
```

Description

ConstIMFMessageStart is a constant value that represents the TIdIMFDecoder *↗TIdIMFDecoder* state value assigned when the decoder has starter to process an Internet Mail Format message.

3.6.89. ConstIMFStart

State value indicating readiness to process an IMF message.

```
ConstIMFStart = 0;
```

Description

ConstIMFStart is a constant value that represents the TIdIMFDecoder *↗TIdIMFDecoder* state value assigned when the decoder is ready to begin processing an Internet Mail Format message.

3.6.90. ConstName

Represents the header name for inline files in an IMF message.

```
ConstName = 'name';
```

Description

ConstName is a constant value that represents the RFC 822-compliant header name for inline files in an Internet Mail Format message.

3.6.91. cOpCodeBits

Bit mask constant.

```
cOpCodeBits = $7800;
```

Description

cOpCodeBits is a constant value that represents the bit mask for the OpCode bits in a DNS OpCode.

3.6.92. cOpCodeMask

Bit mask constant.

```
cOpCodeMask = $87FF;
```

Description

cOpCodeMask is a constant value that represents the mask for the value of OpCode bits in a DNS Query header.

3.6.93. cOPCodeStrs

OpCode names.

```
cOPCodeStrs: Array[cResQuery ↗cResQuery..cResStatus] Of String[7] = ('Query', 'IQuery', 'Status');
```

Description

cOPCodeStrs is a constant array that represents the names for DNS OpCode constants.

3.6.94. CP\_FALLBACK

Fallback Coder Priority.

```
CP_FALLBACK = 0;
```

Description

CP\_FALLBACK is the constant value for the Fallback Coder Priority.

### 3.6.95. CP\_IMF

Coder priority for Internet Mail coders.

```
CP_IMF = 1;
```

**Description**

CP\_IMF is a constant value that represents the coder priority for Internet Mail coders.

### 3.6.96. CP\_STANDARD

Standard Coder Priority.

```
CP_STANDARD = 8;
```

**Description**

CP\_STANDARD is the constant value that represents the Standard Coder Priority.

### 3.6.97. cPTR

Resource type constant.

```
cPTR = 12;
```

**Description**

cPTR is a constant used in the TIdDNSResolver *↗TIdDNSResolver* for requesting a DNS name from an IP Address (Reverse DNS). Thus, a query for 129.71.2.4 will result in the IP address 129.71.2.4 in the TIdDNSResourceItem.RData.DomainName property.

### 3.6.98. cQClassStr

Resource class names.

```
cQClassStr: Array[cIN ↗cIN..CHs] Of String[3] =  
( 'IN', 'CS', 'CH', 'HS' );
```

**Description**

cQClassStr is a constant array that represents the names for resource class constants.

### 3.6.99. cQRBit

Bit mask constant.

```
cQRBit = $8000;
```

**Description**

cQRBit is a constant value that represents the bit mask for the Query or Response bit flag in a DNS header. When QR is 0, the header is for a Query. When QR is 1, the header is for a Response.

### 3.6.100. cQRMask

Bit mask constant.

```
cQRMask = $FFFF;
```

**Description**

cQRMask is a constant value that represents the mask for accessing the value of the QR flag in a DNS header.

### 3.6.101. CR

Represents the Carriage Return character.

```
CR = #13;
```

**Description**

CR is the constant value used to represent the Carriage Return character.

3.6.102. cRABit

Bit mask flag.

```
cRABit = $0080;
```

Description

cRABit is a constant value that represents the bit mask for the recursive search available flag in a DNS header. When RA is 1, the server supports Recursive Search.

3.6.103. cRAMask

Bit mask constant.

```
cRAMask = $FF7F;
```

Description

cRAMask is a constant value that represents the mask for the Recursive Search Available value in a DNS Header.

3.6.104. cRCodeBits

Bit mask constant.

```
cRCodeBits = $000F;
```

Description

cRCodeBits is a constant value that represents the bit mask for the DNS Response Code in a DNS header.

3.6.105. cRCodeFormatErr

Response code constant.

```
cRCodeFormatErr = 1;
```

Description

cRCodeFormatErr is a constant value that represents the response code for an invalid or malformed DNS query packet.

3.6.106. cRCodeMask

Bit mask constant.

```
cRCodeMask = $FFF0;
```

Description

cRCodeMask is a constant value that represents the mask for the Response Code value in a DNS query response.

3.6.107. cRCodeNameErr

Response code constant.

```
cRCodeNameErr = 3;
```

Description

cRCodeNameErr is a constant value that represents the response code for an invalid or malformed Domain Name in the DNS query packet.

3.6.108. cRCodeNoError

Response code constant.

```
cRCodeNoError = 0;
```

Description

cRCodeNoError is a constant value that represents the response code for successful execution of the DNS query packet.

### 3.6.109. cRCodeNotImplemented

Response code constant.

```
cRCodeNotImplemented = 4;
```

**Description**

cRCodeNotImplemented is a constant value that represents the response code for a DNS query packet containing an OpCode or Question not implemented on the DNS server.

### 3.6.110. cRCodeRefused

Response code constant.

```
cRCodeRefused = 5;
```

**Description**

cRCodeRefused is a constant value that represents the response code for a DNS query packet that has been refused by the DNS server.

### 3.6.111. cRCodeServerError

Response code constant.

```
cRCodeServerError = 2;
```

**Description**

cRCodeServerError is a constant value that represents the response code for a DNS query packet that has resulted in an error on the DNS server.

### 3.6.112. cRCodeStrs

Response Code names.

```
cRCodeStrs: Array[cRCodeNoError ↗cRCodeNoError..cRCodeRefused] of
```

```
String = (RSCodeNoError ↗RSCodeNoError, RSCodeQueryFormat  
↗RSCodeQueryFormat, RSCodeQueryServer ↗RSCodeQueryServer,  
RSCodeQueryName ↗RSCodeQueryName, RSCodeQueryNotImplemented  
↗RSCodeQueryNotImplemented, RSCodeQueryQueryRefused);
```

**Description**

cRCodeStrs is a constant array that represents the names for Response Code constant values.

### 3.6.113. cRDBit

Bit mask constant.

```
cRDBit = $0100;
```

**Description**

cRDBit is a constant value that represents the bit mask for the Recursive Search Requested by Query flag in the DNS query header.

### 3.6.114. cRDMask

Bit mask constant.

```
cRDMask = $FEFF;
```

**Description**

cRDMask is a constant value that represents the mask for the Recursive Search Requested by Query flag value in the DNS query header.

### 3.6.115. cResIQuery

DNS query OpCode constant.

```
cResIQuery = 1;
```

**Description**



cResIQuery is a constant value that represents the DNS Query OpCode for the Inverse Query request.

3.6.116. cResQuery

DNS Query OpCode constant.

cResQuery = 0;

Description

cResQuery is a constant value that represents the DNS Query OpCode for a standard resource query request.

3.6.117. cResStatus

DNS Query OpCode constant.

cResStatus = 2;

Description

cResStatus is a constant value that represents the DNS Query OpCode for a DNS Server Status request.

3.6.118. cSOA

Resource type constant.

cSOA = 6;

Description

cSOA is a constant value that represents the resource type that marks the start of a zone of authority resource type.

3.6.119. csSPECIALS

3.6.120. cStar

Resource type constant.

cStar = 255;

Description

cStar is a constant value that represents the resource type used for a request retrieving all Resource Records for the target.

3.6.121. CT\_Creation

Indicates the Coder type for RegisterCoderClass ↗RegisterCoderClass.

Value:

0.

CT\_Creation = 0;

Description

CT\_Creation is a **Byte** constant that represents the Coder type for RegisterCoderClass ↗RegisterCoderClass.

CT\_Creation is used when registering new class instances of a coder class descendant.

3.6.122. CT\_Realisation

Indicates the Coder type for RegisterCoderClass ↗RegisterCoderClass.

Value:

\$80 (80 Hex, 128 Decimal).

```
CT_Realisation = $80;
```

**Description**

CT\_Realisation is a **Byte** constant that represents the Coder type for RegisterCoderClass *↗RegisterCoderClass*.

CT\_Realisation is used when realizing new instances of a coder class descendant.

### 3.6.123. cTCBit

Bit mask constant.

```
cTCBit = $0200;
```

**Description**

cTCBit is a constant value that represents the bit mask for the Truncation Bit in the DNS Query header. cTCBit indicates a message that has been truncated for length.

### 3.6.124. cTCMask

Bit mask value.

```
cTCMask = $FDFF;
```

**Description**

cTCMask is a constant value that represents the mask for the Truncation Bit flag value in a DNS Query header.

### 3.6.125. CTL3To4

Code table length for 3-to-4 coders.

```
CTL3To4 = 64;
```

**Description**

CTL3To4 is the code table length for 3-to-4 coders.

### 3.6.126. cTXT

Resource type constant.

```
cTXT = 16;
```

**Description**

cTXT is a constant value that represents the resource type for a Text String.

### 3.6.127. cWKS

Resource type constant.

```
cWKS = 11;
```

**Description**

cWKS is a constant value that represents the resource type for a well known service description.

### 3.6.128. DEF\_PACKET\_SIZE

Specifies the default packet size for ICMP messages.

```
DEF_PACKET_SIZE = 32;
```

**Description**

DEF\_PACKET\_SIZE is a constant value that specifies the default packet size for ICMP request messages, like an Echo Request.

### 3.6.129. EADDRINUSE

```
EADDRINUSE = WSAEADDRINUSE ↗WSAEADDRINUSE;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.130. EADDRNOTAVAIL

EADDRNOTAVAIL = WSAEADDRNOTAVAIL *⚡WSAEADDRNOTAVAIL*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.131. EAFNOSUPPORT

EAFNOSUPPORT = WSAEAFNOSUPPORT *⚡WSAEAFNOSUPPORT*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.132. EALREADY

EALREADY = WSAEALREADY *⚡WSAEALREADY*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.133. ECONNABORTED

ECONNABORTED = WSAECONNABORTED *⚡WSAECONNABORTED*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.134. ECONNREFUSED

ECONNREFUSED = WSAECONNREFUSED *⚡WSAECONNREFUSED*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.135. ECONNRESET

ECONNRESET = WSAECONNRESET *⚡WSAECONNRESET*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.136. EDESTADDRREQ

EDESTADDRREQ = WSAEDESTADDRREQ *⚡WSAEDESTADDRREQ*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.137. EDQUOT

EDQUOT = WSAEDQUOT *⚡WSAEDQUOT*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.138. EHOSTDOWN

EHOSTDOWN = WSAEHOSTDOWN *⚡WSAEHOSTDOWN*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.139. EHOSTUNREACH

EHOSTUNREACH = WSAEHOSTUNREACH *⚡WSAEHOSTUNREACH*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.140. EINPROGRESS

EINPROGRESS = WSAEINPROGRESS *⚡WSAEINPROGRESS*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.141. EISCONN

EISCONN = WSAEISCONN *⚡WSAEISCONN*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.142. ELOOP

ELOOP = WSAELOOP *⚡WSAELOOP*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.143. EMSGSIZE

EMSGSIZE = WSAEMSGSIZE *⚡WSAEMSGSIZE*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.144. ENAMETOOLONG

ENAMETOOLONG = WSAENAMETOOLONG *⚡WSAENAMETOOLONG*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.145. ENETDOWN

ENETDOWN = WSAENETDOWN *⚡WSAENETDOWN*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.146. ENETRESET

ENETRESET = WSAENETRESET *⚡WSAENETRESET*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.147. ENETUNREACH

ENETUNREACH = WSAENETUNREACH *⚡WSAENETUNREACH*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.148. ENOBUFS

ENOBUFS = WSAENOBUFS *⚡WSAENOBUFS*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.149. ENOPROTOOPT

ENOPROTOOPT = WSAENOPROTOOPT *⚡WSAENOPROTOOPT*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.150. ENOTCONN

ENOTCONN = WSAENOTCONN *⚡WSAENOTCONN*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.151. ENOTEMPTY

ENOTEMPTY = WSAENOTEMPTY *⚡WSAENOTEMPTY*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.152. ENOTSOCK

ENOTSOCK = WSAENOTSOCK *⚡WSAENOTSOCK*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.153. EOL

Represents the End-of-Line character sequence.

EOL = CR *⚡*CR + LF *⚡*LF;

Description

EOL is the constant value used to represent the End-of-Line character sequence commonly used by many Internet protocols.

3.6.154. EOPNOTSUPP

EOPNOTSUPP = WSAEOPNOTSUPP *⚡WSAEOPNOTSUPP*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.155. EPFNOSUPPORT

EPFNOSUPPORT = WSAEPFNOSUPPORT *⚡WSAEPFNOSUPPORT*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.156. EPROCLIM

EPROCLIM = WSAEPROCLIM *⚡WSAEPROCLIM*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.157. EPROTONOSUPPORT

EPROTONOSUPPORT = WSAEPROTONOSUPPORT *⚠WSAEPROTONOSUPPORT*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.158. EPROTOTYPE

EPROTOTYPE = WSAEPROTOTYPE *⚠WSAEPROTOTYPE*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.159. EREMOTE

EREMOTE = WSAEREMOTE *⚠WSAEREMOTE*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.160. ErrAccessViolation

Error number for server read file or write file errors.

ErrAccessViolation = 2;

**Description**

ErrAccessViolation is the constant error number for EldTFTPAccessViolation *⚠EldTFTPAccessViolation* Exceptions raised when CanRead is set to FALSE in a TIdTrivialFTPServer *⚠TIdTrivialFTPServer* onReadFile or onWriteFile.

### 3.6.161. ErrAllocationExceeded

Exception raised when a TFTP file writing fails.

ErrAllocationExceeded = 3;

**Description**

ErrAllocationExceeded is the constant error number for the EldTFTPAllocationExceeded *⚠EldTFTPAllocationExceeded* Exception raised when a TFTP file write error occurs.

### 3.6.162. ErrFileAlreadyExists

Error number for the TFTP File Exists error.

ErrFileAlreadyExists = 6;

**Description**

ErrFileAlreadyExists represents the constant error number for the TrivialFTP EldTFTPFileAlreadyExists *⚠EldTFTPFileAlreadyExists* exception.

### 3.6.163. ErrFileNotFound

Error number for File Not Found errors.

ErrFileNotFound = 1;

**Description**

ErrFileNotFoundError is the constant error number for File Not Found errors.

### 3.6.164. ErrIllegalOperation

Error number for illegal operation messages.

ErrIllegalOperation = 4;

**Description**

ErrIllegalOperation is the constant error number for exceptions raised by several methods when the parameter provided does not map to an enumerated type, such as when the mode setting is not octet or netascii.

3.6.165. ErrNoSuchUser

Error number for TFTP user authentication failure.

```
ErrNoSuchUser = 7;
```

Description

ErrNoSuchUser is a constant value that represents the error number for unknown user name or password on a TrivialFTP connection.

3.6.166. ErrOptionNegotiationFailed

Error number for TFTP option negotiation.

```
ErrOptionNegotiationFailed = 8;
```

Description

ErrOptionNegotiationFailed is a constant value that specifies the error number for an error that occurs while negotiating TrivialFTP options.

3.6.167. ErrUndefined

Error number for undefined TrivialFTP errors.

```
ErrUndefined = 0;
```

Description

ErrUndefined is a constant value the specifies the error number for undefined TrivialFTP errors.

3.6.168. ErrUnknownTransferID

Error number for an unknown TFTP transfer ID.

```
ErrUnknownTransferID = 5;
```

Description

ErrUnknownTransferID is a constant value that represents the error number for an unknown TrivialFTP transfer ID.

3.6.169. ESHUTDOWN

```
ESHUTDOWN = WSAESHUTDOWN ⚡WSAESHUTDOWN;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.170. ESOCKTNOSUPPORT

```
ESOCKTNOSUPPORT = WSAESOCKTNOSUPPORT ⚡WSAESOCKTNOSUPPORT;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.171. ESTALE

```
ESTALE = WSAESTALE ⚡WSAESTALE;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.172. ETIMEDOUT

ETIMEDOUT = WSAETIMEDOUT *⚡WSAETIMEDOUT* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.173. ETOOMANYREFS

ETOOMANYREFS = WSAETOOMANYREFS *⚡WSAETOOMANYREFS* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.174. EUSERS

EUSERS = WSAEUSERS *⚡WSAEUSERS* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.175. EWOULDBLOCK

EWOULDBLOCK = WSAEWOULDBLOCK *⚡WSAEWOULDBLOCK* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.176. FD\_ACCEPT

FD\_ACCEPT = \$08 ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.177. FD\_CLOSE

FD\_CLOSE = \$20 ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.178. FD\_CONNECT

FD\_CONNECT = \$10 ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.179. FD\_OOB

FD\_OOB = \$04 ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.180. FD\_READ

FD\_READ = \$01 ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.



### 3.6.181. FD\_SETSIZE

```
FD_SETSIZE = 64;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.182. FD\_WRITE

```
FD_WRITE = $02;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.183. FIOASYNC

```
FIOASYNC = IOC_IN IOC_IN or ((Longint(SizeOf(Longint)) and  
IOCPARM_MASK) shl 16) or (Longint(Byte('f')) shl 8) or 125;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.184. FIONBIO

```
FIONBIO = IOC_IN IOC_IN or ((Longint(SizeOf(Longint)) and  
IOCPARM_MASK) shl 16) or (Longint(Byte('f')) shl 8) or 126;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.185. FIONREAD

```
FIONREAD = IOC_OUT IOC_OUT or ((Longint(SizeOf(Longint)) and  
IOCPARM_MASK) shl 16) or (Longint(Byte('f')) shl 8) or 127;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.186. GContentType

This is the default value for the TIdHTTPResponseInfo.ContentType property.

```
GContentType = 'text/html';
```

**Description**

GContentType is the default value for the TIdHTTPResponseInfo.ContentType property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.187. GFContentLength

This is the default value for the TIdHTTPResponseInfo.ContentLength property.

```
GFContentLength = -1;
```

**Description**

GFContentLength is the default value for the TIdHTTPResponseInfo.ContentLength property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.188. GMaxAge

This is the default value for the TIdCookie.MaxAge property.

```
GMaxAge = -1;
```

**Description**

GMaxAge is the default value for the TIdCookie.MaxAge property. This is used in the class definition and the constructor. Changing this value will change the default value for that property.

### 3.6.189. GRequestedBlockSize

Default TrivialFTP request block size.

```
GRequestedBlockSize = 1500;
```

**Description**

GRequestedBlockSize is a constant Integer value that represents the global default block size used TFTP requests. TIdTrivialFTP.RequestedBlockSize receives the value of GRequestedBlockSize in the constructor for the object instance.

### 3.6.190. GFTTL

Default Time-To-Live for TIdRawBase *↗TIdRawBase* object instances.

```
GFTTL = 128;
```

**Description**

GFTTL is a constant Integer value that represents the default Time-To-Live assigned to the TIdRawBase.TTL property in the constructor for the object instance.

### 3.6.191. GLoginAttempts

Maximum login attempts for a threaded connection to the Telnet server.

```
GLoginAttempts = 3;
```

**Description**

GLoginAttempts is a constant Integer value that represents the maximum number of login attempts permitted when attempting a connection to TIdTelnetServer *↗TIdTelnetServer*. GLoginAttempts is assigned to the TIdTelnetServer.LoginAttempts property in the constructor for the server instance, and is used for all TIdPeerThread *↗TIdPeerThread* connections to the server.

### 3.6.192. GPathSep

```
=====
```

### 3.6.193. GReceiveTimeout

Default timeout value for read operations in TIdRawBase *↗TIdRawBase* object instances.

```
GReceiveTimeout = 4000;
```

**Description**

GReceiveTimeout is a constant Integer value that represents the default timeout value for receive operations. GReceiveTimeout is assigned to the TIdRawBase.ReceiveTimeout property in the constructor for the object instance, and is used to determine when the socket connection is readable.

### 3.6.194. GRecvBufferSizeDefault

Default value for the TIdTCPConnection.SendBufferSize property.

```
GRecvBufferSizeDefault = 32768;
```

**Description**

GRecvBufferSizeDefault is the default value for the TIdTCPConnection.SendBufferSize property. GRecvBufferSizeDefault is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.195. GResponseNo

This is the default value for the TIdHTTPResponseInfo.ResponseNo property.

```
GResponseNo = 200;
```

**Description**

GResponseNo is the default value for the TIdHTTPResponseInfo.ResponseNo property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.196. GSendBufferSizeDefault

Default value for the TIdTCPConnection.RecvBufferSize property.

```
GSendBufferSizeDefault = 32768;
```

**Description**

GSendBufferSizeDefault is the default value for the TIdTCPConnection.RecvBufferSize property. GSendBufferSizeDefault is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.197. GServerSoftware

This is the default value for the TIdHTTPResponseInfo.ServerSoftware property.

```
GServerSoftware = gsIdProductName ⚡gsIdProductName + '/' +  
gsIdVersion ⚡gsIdVersion;
```

**Description**

GServerSoftware is the default value for the TIdHTTPResponseInfo.ServerSoftware property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.198. gsIdProductName

Identifies the product name.

```
gsIdProductName = 'Indy';
```

**Description**

gsIdProductName is a constant String value that contains the name of this product ('Indy').

### 3.6.199. gsIdVersion

Identifies the product version.

```
gsIdVersion = '8.0.21';
```

**Description**

gsIdVersion is a constant String value that identifies the version of this product.

### 3.6.200. GTransferMode

Default TrivialFTP transfer mode.

```
GTransferMode = tfOctet;
```

**Description**

GTransferMode is a constant value that represents the default TIdTFTPMode ⚡TIdTFTPMode file transfer mode used by the TIdTrivialFTP.TransferMode property.

### 3.6.201. HalfCodeTable

Length of the coding table as represented in a UUEncode or XXEncode header.

```
HalfCodeTable = CTL3To4 ⚡CTL3To4 div 2;
```

**Description**  
HalfCodeTable is the length of the coding table as represented in a UUEncode or XXEncode header.

3.6.202. hdrsize

Bytes in a TFTP Header.

hdrsize = 4;

**Description**  
hdrsize represents the number of bytes used in a TrivialFTP header construct for TFTP data packets.

3.6.203. HOST\_NOT\_FOUND

HOST\_NOT\_FOUND = WSAHOST\_NOT\_FOUND ~~WSAHOST\_NOT\_FOUND~~;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.204. ICMP\_MIN

Specifies the minimum number of data bytes in an ICMP reply.

ICMP\_MIN = 8;

**Description**  
ICMP\_MIN is a constant value that specifies the minimum number of data bytes in an ICMP reply.

3.6.205. Id\_ARP\_HSIZE

Id\_ARP\_HSIZE = \$1C;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.206. Id\_ARPHRD\_ETHER

Id\_ARPHRD\_ETHER = 1;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.207. Id\_ARPOP\_INVREPLY

Id\_ARPOP\_INVREPLY = 9;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.208. Id\_ARPOP\_INVREQUEST

Id\_ARPOP\_INVREQUEST = 8;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.209. Id\_ARPOP\_REPLY

Id\_ARPOP\_REPLY = 2;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.210. Id\_ARPOP\_REQUEST

```
Id_ARPOP_REQUEST = 1;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.211. Id\_ARPOP\_REVREPLY

```
Id_ARPOP_REVREPLY = 4;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.212. Id\_ARPOP\_REVREQUEST

```
Id_ARPOP_REVREQUEST = 3;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.213. ID\_Default\_TIdAntiFreezeBase\_Active

This is the default value for the TIdAntiFreezeBase.Active property.

```
ID_Default_TIdAntiFreezeBase_Active = True;
```

Description

ID\_Default\_TIdAntiFreezeBase\_Active is the default value for the TIdAntiFreeze.Active property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the

default value for that property.

3.6.214. ID\_Default\_TIdAntiFreezeBase\_ApplicationHasPriority

```
ID_Default_TIdAntiFreezeBase_ApplicationHasPriority = True;
```

Description

ID\_Default\_TIdAntiFreezeBase\_ApplicationHasPriority

3.6.215. ID\_Default\_TIdAntiFreezeBase\_IdleTimeOut

This is the default value for the TIdAntiFreezeBase.IdleTimeOut property.

```
ID_Default_TIdAntiFreezeBase_IdleTimeOut = 250;
```

Description

ID\_Default\_TIdAntiFreezeBase\_IdleTimeOut is the default value for the TIdAntiFreeze.IdleTimeOut property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.216. ID\_Default\_TIdAntiFreezeBase\_OnlyWhenIdle

This is the default value for the TIdAntiFreezeBase.OnlyWhenIdle property.

```
ID_Default_TIdAntiFreezeBase_OnlyWhenIdle = True;
```

Description

ID\_Default\_TIdAntiFreezeBase\_OnlyWhenIdle is the default value for the TIdAntiFreeze.OnlyWhenIdle property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.217. Id\_DNS\_HSIZE

Id\_DNS\_HSIZE = \$0C;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.218. Id\_ETH\_HSIZE

Id\_ETH\_HSIZE = \$0E;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.219. Id\_ETHER\_ADDR\_LEN

Id\_ETHER\_ADDR\_LEN = 6;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.220. Id\_ETHERTYPE\_ARP

Id\_ETHERTYPE\_ARP = \$0806;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.221. Id\_ETHERTYPE\_IP

Id\_ETHERTYPE\_IP = \$0800;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.222. Id\_ETHERTYPE\_LOOPBACK

Id\_ETHERTYPE\_LOOPBACK = \$9000;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.223. Id\_ETHERTYPE\_PUP

Id\_ETHERTYPE\_PUP = \$0200;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.224. Id\_ETHERTYPE\_REVARP

Id\_ETHERTYPE\_REVARP = \$8035;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.225. Id\_ETHERTYPE\_VLAN

Id\_ETHERTYPE\_VLAN = \$8100;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.226. Id\_ICMP\_ECHO

Id\_ICMP\_ECHO = 8;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.227. Id\_ICMP\_ECHO\_HSIZE

Id\_ICMP\_ECHO\_HSIZE = \$08;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.228. Id\_ICMP\_ECHOREPLY

Id\_ICMP\_ECHOREPLY = 0;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.229. Id\_ICMP\_HSIZE

Id\_ICMP\_HSIZE = \$04;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.230. Id\_ICMP\_IREQ

Id\_ICMP\_IREQ = 15;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.231. Id\_ICMP\_IREQREPLY

Id\_ICMP\_IREQREPLY = 16;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.232. Id\_ICMP\_MASK\_HSIZE

Id\_ICMP\_MASK\_HSIZE = \$0C;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.233. Id\_ICMP\_MASKREPLY

Id\_ICMP\_MASKREPLY = 18;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.234. Id\_ICMP\_MASKREQ

Id\_ICMP\_MASKREQ = 17;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.235. Id\_ICMP\_PARAMPROB

Id\_ICMP\_PARAMPROB = 12;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.236. Id\_ICMP\_PARAMPROB\_OPTABSENT

Id\_ICMP\_PARAMPROB\_OPTABSENT = 1;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.237. Id\_ICMP\_REDIRECT

Id\_ICMP\_REDIRECT = 5;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.238. Id\_ICMP\_REDIRECT\_HOST

Id\_ICMP\_REDIRECT\_HOST = 1;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.239. Id\_ICMP\_REDIRECT\_HSIZE

Id\_ICMP\_REDIRECT\_HSIZE = \$08;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.240. Id\_ICMP\_REDIRECT\_NET

Id\_ICMP\_REDIRECT\_NET = 0;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.241. Id\_ICMP\_REDIRECT\_TOSHOST

Id\_ICMP\_REDIRECT\_TOSHOST = 3;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.242. Id\_ICMP\_REDIRECT\_TOSNET

Id\_ICMP\_REDIRECT\_TOSNET = 2;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.243. Id\_ICMP\_ROUTERADVERT

Id\_ICMP\_ROUTERADVERT = 9;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.



3.6.244. Id\_ICMP\_ROUTERSOLICIT

Id\_ICMP\_ROUTERSOLICIT = 10;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.245. Id\_ICMP\_SOURCEQUENCH

Id\_ICMP\_SOURCEQUENCH = 4;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.246. Id\_ICMP\_TIMEXCEED\_HSIZE

Id\_ICMP\_TIMEXCEED\_HSIZE = \$08;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.247. Id\_ICMP\_TIMXCEED

Id\_ICMP\_TIMXCEED = 11;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.248. Id\_ICMP\_TIMXCEED\_INTRANS

Id\_ICMP\_TIMXCEED\_INTRANS = 0;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.249. Id\_ICMP\_TIMXCEED\_REASS

Id\_ICMP\_TIMXCEED\_REASS = 1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.250. Id\_ICMP\_TS\_HSIZE

Id\_ICMP\_TS\_HSIZE = \$14;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.251. Id\_ICMP\_TSTAMP

Id\_ICMP\_TSTAMP = 13;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.252. Id\_ICMP\_TSTAMPREPLY

Id\_ICMP\_TSTAMPREPLY = 14;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.253. Id\_ICMP\_UNREACH

Id\_ICMP\_UNREACH = 3;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.254. Id\_ICMP\_UNREACH\_FILTER\_PROHIB

Id\_ICMP\_UNREACH\_FILTER\_PROHIB = 13;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.255. Id\_ICMP\_UNREACH\_HOST

Id\_ICMP\_UNREACH\_HOST = 1;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.256. Id\_ICMP\_UNREACH\_HOST\_PRECEDENCE

Id\_ICMP\_UNREACH\_HOST\_PRECEDENCE = 14;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.257. Id\_ICMP\_UNREACH\_HOST\_PROHIB

Id\_ICMP\_UNREACH\_HOST\_PROHIB = 10;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.258. Id\_ICMP\_UNREACH\_HOST\_UNKNOWN

Id\_ICMP\_UNREACH\_HOST\_UNKNOWN = 7;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.259. Id\_ICMP\_UNREACH\_HSIZE

Id\_ICMP\_UNREACH\_HSIZE = \$08;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.260. Id\_ICMP\_UNREACH\_ISOLATED

Id\_ICMP\_UNREACH\_ISOLATED = 8;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.261. Id\_ICMP\_UNREACH\_NEEDFRAG

Id\_ICMP\_UNREACH\_NEEDFRAG = 4;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.262. Id\_ICMP\_UNREACH\_NET

Id\_ICMP\_UNREACH\_NET = 0;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.263. Id\_ICMP\_UNREACH\_NET\_PROHIB

Id\_ICMP\_UNREACH\_NET\_PROHIB = 9;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.264. Id\_ICMP\_UNREACH\_NET\_UNKNOWN

Id\_ICMP\_UNREACH\_NET\_UNKNOWN = 6;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.265. Id\_ICMP\_UNREACH\_PORT

Id\_ICMP\_UNREACH\_PORT = 3;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.266. Id\_ICMP\_UNREACH\_PRECEDENCE\_CUTOFF

Id\_ICMP\_UNREACH\_PRECEDENCE\_CUTOFF = 15;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.267. Id\_ICMP\_UNREACH\_PROTOCOL

Id\_ICMP\_UNREACH\_PROTOCOL = 2;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.268. Id\_ICMP\_UNREACH\_SRCFAIL

Id\_ICMP\_UNREACH\_SRCFAIL = 5;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.269. Id\_ICMP\_UNREACH\_TOSHOST

Id\_ICMP\_UNREACH\_TOSHOST = 12;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.270. Id\_ICMP\_UNREACH\_TOSNET

Id\_ICMP\_UNREACH\_TOSNET = 11;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.271. Id\_IGMP\_HSIZE

Id\_IGMP\_HSIZE = \$08;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.272. Id\_IGMP\_LEAVE\_GROUP

Id\_IGMP\_LEAVE\_GROUP = \$17;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.273. Id\_IGMP\_MEMBERSHIP\_QUERY

Id\_IGMP\_MEMBERSHIP\_QUERY = \$11;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.274. Id\_IGMP\_V1\_MEMBERSHIP\_REPORT

Id\_IGMP\_V1\_MEMBERSHIP\_REPORT = \$12;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.275. Id\_IGMP\_V2\_MEMBERSHIP\_REPORT

Id\_IGMP\_V2\_MEMBERSHIP\_REPORT = \$16;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.276. Id\_INADDR\_ANY

Id\_INADDR\_ANY = INADDR\_ANY;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.277. Id\_INADDR\_NONE

Id\_INADDR\_NONE = INADDR\_NONE;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.278. Id\_INVALID\_SOCKET

Id\_INVALID\_SOCKET = INVALID\_SOCKET *⚡INVALID\_SOCKET*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.279. Id\_IP\_DF

Id\_IP\_DF = \$4000;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.280. Id\_IP\_HSIZE

Id\_IP\_HSIZE = \$14;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.281. Id\_IP\_MAXPACKET

Id\_IP\_MAXPACKET = 65535;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.282. Id\_IP\_MF

Id\_IP\_MF = \$2000;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.283. Id\_IP\_OFFMASK

Id\_IP\_OFFMASK = \$1FFF;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.284. Id\_IP\_RF

Id\_IP\_RF = \$8000;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.285. Id\_IP\_TTL

Id\_IP\_TTL = IP\_TTL *⚡IP\_TTL*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.286. Id\_IPPROTO\_ICMP

Id\_IPPROTO\_ICMP = IPPROTO\_ICMP *⚡IPPROTO\_ICMP*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.287. Id\_IPPROTO\_IGMP

Id\_IPPROTO\_IGMP = IPPROTO\_IGMP *⚡IPPROTO\_IGMP*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.288. Id\_IPPROTO\_IP

Id\_IPPROTO\_IP = IPPROTO\_IP *⚡IPPROTO\_IP*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.289. Id\_IPPROTO\_MAX

Id\_IPPROTO\_MAX = IPPROTO\_MAX *⚡IPPROTO\_MAX*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.290. Id\_IPPROTO\_RAW

Id\_IPPROTO\_RAW = IPPROTO\_RAW *⚡IPPROTO\_RAW*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.291. Id\_IPPROTO\_TCP

Id\_IPPROTO\_TCP = IPPROTO\_TCP *⚡IPPROTO\_TCP*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.292. Id\_IPPROTO\_UDP

Id\_IPPROTO\_UDP = IPPROTO\_UDP *⚡IPPROTO\_UDP*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.293. ID\_LOGBASE\_Active

Default value for the TIdLogBase.Active property.

ID\_LOGBASE\_Active = False;

**Description**

ID\_LOGBASE\_Active is the default value for the TIdLogBase.Active property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.294. ID\_LOGBASE\_LogTime

Default value for the TIdLogBase.LogTime property.

ID\_LOGBASE\_LogTime = True;

**Description**

ID\_LOGBASE\_LogTime is the default value for the TIdLogBase.LogTime property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.295. ID\_MAPPED\_PORT\_TCP\_PORT

Default value for the TIdMappedPortTCP.MappedPort property.

ID\_MAPPED\_PORT\_TCP\_PORT = 0;

**Description**

ID\_MAPPED\_PORT\_TCP\_PORT is the default value for the TIdMappedPortTCP.MappedPort property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.296. Id\_MAX\_IPOPTLEN

Id\_MAX\_IPOPTLEN = 40;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.297. ID\_MSG\_NODECODE

Default value for the TIdMessage.NoDecode property.

```
ID_MSG_NODECODE = False;
```

**Description**

ID\_MSG\_NODECODE is the default value for the TIdMessage.NoDecode property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.298. ID\_MSG\_PRIORITY

Default value for the TIdMessage.Priority property.

```
ID_MSG_PRIORITY = mpNormal;
```

**Description**

ID\_MSG\_PRIORITY is the default value for the TIdMessage.Priority property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.299. ID\_MSG\_USENOWFORDATE

Default value for the TIdMessage.UseNowForDate property.

```
ID_MSG_USENOWFORDATE = True;
```

**Description**

ID\_MSG\_USENOWFORDATE is the default value for the TIdMessage.UseNowForDate property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.300. ID\_NC\_MASK\_LENGTH

Default length of network masks for TIdNetworkCalculator *↗TIdNetworkCalculator* object instances.

```
ID_NC_MASK_LENGTH = 32;
```

**Description**

ID\_NC\_MASK\_LENGTH is a constant Cardinal value that represents the length of the network mask used in calculating IP addresses for TIdNetworkCalculator *↗TIdNetworkCalculator*. ID\_NC\_MASK\_LENGTH is assigned to the TIdNetworkCalculator.NetworkMaskLength property in the constructor for the object instance.

### 3.6.301. ID\_NETWORKCLASS

Default network class for IP addresses in TIdNetworkCalculator *↗TIdNetworkCalculator* instances.

```
ID_NETWORKCLASS = ID_NET_CLASS_A;
```

**Description**

ID\_NETWORKCLASS is a constant value that represents the default TNetworkClass *↗TNetworkClass* value assigned to the TIdNetworkCalculator.NetworkClass property in the constructor for the object instance.

### 3.6.302. Id\_PF\_INET

```
Id_PF_INET = PF_INET ↗PF_INET;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.303. Id\_RIP\_HSIZE

Id\_RIP\_HSIZE = \$18;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.304. Id\_RIPCMD\_MAX

Id\_RIPCMD\_MAX = 7;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.305. Id\_RIPCMD\_POLL

Id\_RIPCMD\_POLL = 5;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.306. Id\_RIPCMD\_POLLENTRY

Id\_RIPCMD\_POLLENTRY = 6;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.307. Id\_RIPCMD\_REQUEST

Id\_RIPCMD\_REQUEST = 1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.308. Id\_RIPCMD\_RESPONSE

Id\_RIPCMD\_RESPONSE = 2;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.309. Id\_RIPCMD\_TRACEOFF

Id\_RIPCMD\_TRACEOFF = 4;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.310. Id\_RIPCMD\_TRACEON

Id\_RIPCMD\_TRACEON = 3;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.311. Id\_RIPVER\_0

Id\_RIPVER\_0 = 0;

Description

The text for this constant has been generated automatically. This means that it is not documented.



3.6.312. Id\_RIPVER\_1

Id\_RIPVER\_1 = 1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.313. Id\_RIPVER\_2

Id\_RIPVER\_2 = 2;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.314. Id\_SD\_Both

Id\_SD\_Both = 2;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.315. Id\_SD\_Recv

Id\_SD\_Recv = 0;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.316. Id\_SD\_Send

Id\_SD\_Send = 1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.317. ID\_SIMPLE\_SERVER\_BOUND\_PORT

Default port number for TIdSimpleServer *↗TIdSimpleServer* object instances.

ID\_SIMPLE\_SERVER\_BOUND\_PORT = 0;

Description

ID\_SIMPLE\_SERVER\_BOUND\_PORT is a constant Integer value that represents the default port number assigned to the TIdSimpleServer.BoundPort property in the constructor for the object instance.

3.6.318. Id\_SO\_BROADCAST

Id\_SO\_BROADCAST = SO\_BROADCAST *↗SO\_BROADCAST*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.319. Id\_SO\_DEBUG

Id\_SO\_DEBUG = SO\_DEBUG *↗SO\_DEBUG*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.320. Id\_SO\_DONTRROUTE

Id\_SO\_DONTRROUTE = SO\_DONTRROUTE *↗SO\_DONTRROUTE*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.321. Id\_SO\_KEEPALIVE

```
Id_SO_KEEPALIVE = SO_KEEPALIVE ⚡SO_KEEPALIVE;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.322. Id\_SO\_LINGER

```
Id_SO_LINGER = SO_LINGER ⚡SO_LINGER;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.323. Id\_SO\_OOBINLINE

```
Id_SO_OOBINLINE = SO_OOBINLINE ⚡SO_OOBINLINE;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.324. Id\_SO\_RCVBUF

```
Id_SO_RCVBUF = SO_RCVBUF ⚡SO_RCVBUF;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.325. Id\_SO\_RCVTIMEO

```
Id_SO_RCVTIMEO = SO_RCVTIMEO ⚡SO_RCVTIMEO;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.326. Id\_SO\_REUSEADDR

```
Id_SO_REUSEADDR = SO_REUSEADDR ⚡SO_REUSEADDR;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.327. Id\_SO\_SNDBUF

```
Id_SO_SNDBUF = SO_SNDBUF ⚡SO_SNDBUF;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.328. Id\_SO\_SNDTIMEO

```
Id_SO_SNDTIMEO = SO_SNDTIMEO ⚡SO_SNDTIMEO;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.329. Id SOCK\_DGRAM

```
Id SOCK_DGRAM = Integer(SOCK_DGRAM);
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.330. Id SOCK\_RAW

Id SOCK\_RAW = Integer(SOCK\_RAW);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.331. Id SOCK\_STREAM

Id SOCK\_STREAM = Integer(SOCK\_STREAM);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.332. Id SOCKET\_ERROR

Id SOCKET\_ERROR = SOCKET\_ERROR *↗*SOCKET\_ERROR;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.333. ID SOCKS\_AUTH

Default Socks Authentication for TSocksInfo *↗*TSocksInfo instances.

ID SOCKS\_AUTH = saNoAuthentication;

**Description**

ID SOCKS\_AUTH is a constant value that represents the default TSocksAuthentication *↗*TSocksAuthentication value assigned to the TSocksInfo.Authentication property in the constructor for the object instance.

### 3.6.334. ID SOCKS\_PORT

Default port number for TSocksInfo *↗*TSocksInfo object instances.

ID SOCKS\_PORT = 0;

**Description**

ID SOCKS\_PORT is a constant Integer value that represents the default port number assigned to the TSocksInfo.Port property in the constructor for a TSocksInfo *↗*TSocksInfo object instance.

### 3.6.335. ID SOCKS\_VER

Default Socks version for TSocksInfo *↗*TSocksInfo instances.

ID SOCKS\_VER = svNoSocks;

**Description**

ID SOCKS\_VER is a constant value that represents the default TSocksVersion *↗*TSocksVersion assigned to the TSocksInfo.Version property in the constructor for the object instance.

### 3.6.336. Id SOL\_SOCKET

Id SOL\_SOCKET = SOL\_SOCKET *↗*SOL\_SOCKET;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.337. Id TCP\_ACK

Id TCP\_ACK = \$10;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.338. Id\_TCP\_FIN

Id\_TCP\_FIN = \$01;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.339. Id\_TCP\_HSIZE

Id\_TCP\_HSIZE = \$14;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.340. Id\_TCP\_NODELAY

Id\_TCP\_NODELAY = TCP\_NODELAY *≠TCP\_NODELAY*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.341. Id\_TCP\_PUSH

Id\_TCP\_PUSH = \$08;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.342. Id\_TCP\_RST

Id\_TCP\_RST = \$04;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.343. Id\_TCP\_SYN

Id\_TCP\_SYN = \$02;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.344. Id\_TCP\_URG

Id\_TCP\_URG = \$20;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.345. Id\_TId\_HTTPAutoStartSession

This is the default value for the TIdHTTPServer.AutoStartSession property.

Id\_TId\_HTTPAutoStartSession = False;

**Description**

Id\_TId\_HTTPAutoStartSession is the default value for the TIdHTTPServer.AutoStartSession property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.346. Id\_TId\_HTTPServer\_ParseParams

This is the default value for the TIdHTTPServer.ParseParams property.

Id\_TId\_HTTPServer\_ParseParams = True;

**Description**

Id\_TId\_HTTPServer\_ParseParams is the default value for the TIdHTTPServer.ParseParams property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.347. Id\_TId\_HTTPServer\_SessionState

This is the default value for the TIdHTTPServer.SessionState property.

```
Id_TId_HTTPServer_SessionState = False;
```

**Description**

ID\_Default\_TIdAntiFreezeBase\_Active *≠* ID\_Default\_TIdAntiFreezeBase\_Active is the default value for the TIdHTTPServer.SessionState property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.348. Id\_TId\_HTTPSessionTimeout

This is the default value for the TIdHTTPServer.SessionTimeout property.

```
Id_TId_HTTPSessionTimeout = 0;
```

**Description**

Id\_TId\_HTTPSessionTimeout is the default value for the TIdHTTPServer.SessionTimeout property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.349. Id\_TIdFinger\_VerboseOutput

This is the default value for the TIdFinger.VerboseOutput property.

```
Id_TIdFinger_VerboseOutput = False;
```

**Description**

Id\_TIdFinger\_VerboseOutput is the default value for the TIdFinger.VerboseOutput property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.350. Id\_TIdFTP\_Passive

This is the default value for the TIdFTP.Passive property.

```
Id_TIdFTP_Passive = False;
```

**Description**

Id\_TIdFTP\_Passive is the default value for the TIdFTP.Passive property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.351. Id\_TIdFTP\_TransferType

This is the default value for the TIdFTP.TransferType property.

```
Id_TIdFTP_TransferType = ftBinary;
```

**Description**

Id\_TIdFTP\_TransferType is the default value for the TIdFTP.TransferType property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.352. Id\_TIdGopherServer\_TruncateLength

This is the default value for the TIdGopherServer.TruncateLength property.

```
Id_TIdGopherServer_TruncateLength = 70;
```

**Description**

Id\_TIdGopherServer\_TruncateLength is the default value for the TIdGopherServer.TruncateLength property. This is used in the class definition and the

constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.353. Id\_TIdGopherServer\_TruncateUserFriendly

This is the default value for the TIdGopherServer.TruncateUserFriendly property.

```
Id_TIdGopherServer_TruncateUserFriendly = True;
```

Description

Id\_TIdGopherServer\_TruncateUserFriendly is the default value for the TIdGopherServer.TruncateUserFriendly property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.354. Id\_TIdHTTP\_HandleRedirects

This is the default value for the TIdHTTP.HandleRedirects property.

```
Id_TIdHTTP_HandleRedirects = False;
```

Description

Id\_TIdHTTP\_HandleRedirects is the default value for the TIdHTTP.HandleRedirects property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.355. Id\_TIdHTTP\_ProtocolVersion

This is the default value for the TIdHTTP.ProtocolVersion property.

```
Id_TIdHTTP_ProtocolVersion = pv1_1;
```

Description

Id\_TIdHTTP\_ProtocolVersion is the default value for the TIdHTTP.ProtocolVersion property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for

that property.

3.6.356. Id\_TIdHTTP\_RedirectMax

This is the default value for the TIdHTTP.RedirectMax property.

```
Id_TIdHTTP_RedirectMax = 15;
```

Description

Id\_TIdHTTP\_RedirectMax is the default value for the TIdHTTP.RedirectMax property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.357. Id\_TIDICMP\_ReceiveTimeout

This is the default value for the TIdIcmpClient.ReceiveTimeout property.

```
Id_TIDICMP_ReceiveTimeout = 5000;
```

Description

Id\_TIDICMP\_ReceiveTimeout is the default value for the TIdIcmpClient.ReceiveTimeout property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

3.6.358. ID\_TIDLOGDEBUG\_TARGET

Default value for the TIdLogDebug.Target property.

```
ID_TIDLOGDEBUG_TARGET = ltFile;
```

Description

ID\_TIDLOGDEBUG\_TARGET is the default value for the TIdLogDebug.Target property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.359. Id\_TIdRawBase\_BufferSize

Default buffer size for TIdRawBase *↗TIdRawBase* object instances.

```
Id_TIdRawBase_BufferSize = 8192;
```

**Description**

Id\_TIdRawBase\_BufferSize is a constant Integer value that represents the default buffer size assigned to the TIdRawBase.BufferSize property in the constructor for the object instance.

### 3.6.360. Id\_TIdRawBase\_Port

Default port number for TIdRawBase *↗TIdRawBase* object instances.

```
Id_TIdRawBase_Port = 0;
```

**Description**

Id\_TIdRawBase\_Port is a constant Integer value that represents the default port number assigned to the TIdRawBase.Port property in the constructor for the object instance.

### 3.6.361. ID\_TIDSMTP\_AUTH\_TYPE

Default Authentication Type for TIdSMTP *↗TIdSMTP* object instances.

```
ID_TIDSMTP_AUTH_TYPE = atNone;
```

**Description**

ID\_TIDSMTP\_AUTH\_TYPE is a constant value that represents the default TAuthenticationType *↗TAuthenticationType* value assigned to the TIdSMTP.AuthenticationType property in the constructor for a TIdSMTP *↗TIdSMTP* object instance.

### 3.6.362. ID\_UDP\_BUFFERSIZE

Default UDP buffer size.

```
ID_UDP_BUFFERSIZE = 8192;
```

**Description**

ID\_UDP\_BUFFERSIZE is a constant Integer value that represents the default value for the TIdUDPBase.BufferSize property.

### 3.6.363. Id\_UDP\_HSIZE

```
Id_UDP_HSIZE = $08;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.364. Id\_WSAEACCES

```
Id_WSAEACCES = WSAEACCES ↗WSAEACCES;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.365. Id\_WSAEADDRINUSE

```
Id_WSAEADDRINUSE = WSAEADDRINUSE ↗WSAEADDRINUSE;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.366. Id\_WSAEADDRNOTAVAIL

```
Id_WSAEADDRNOTAVAIL = WSAEADDRNOTAVAIL ↗WSAEADDRNOTAVAIL;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.367. Id\_WSAEAFNOSUPPORT

Id\_WSAEAFNOSUPPORT = WSAEAFNOSUPPORT *⚡WSAEAFNOSUPPORT* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.368. Id\_WSAEALREADY

Id\_WSAEALREADY = WSAEALREADY *⚡WSAEALREADY* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.369. Id\_WSAEBADF

Id\_WSAEBADF = WSAEBADF *⚡WSAEBADF* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.370. Id\_WSAECONNABORTED

Id\_WSAECONNABORTED = WSAECONNABORTED *⚡WSAECONNABORTED* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.371. Id\_WSAECONNREFUSED

Id\_WSAECONNREFUSED = WSAECONNREFUSED *⚡WSAECONNREFUSED* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.372. Id\_WSAECONNRESET

Id\_WSAECONNRESET = WSAECONNRESET *⚡WSAECONNRESET* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.373. Id\_WSAEDESTADDRREQ

Id\_WSAEDESTADDRREQ = WSAEDESTADDRREQ *⚡WSAEDESTADDRREQ* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.374. Id\_WSAEFAULT

Id\_WSAEFAULT = WSAEFAULT *⚡WSAEFAULT* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.375. Id\_WSAEHOSTDOWN

Id\_WSAEHOSTDOWN = WSAEHOSTDOWN *⚡WSAEHOSTDOWN* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.



### 3.6.376. Id\_WSAEHOSTUNREACH

Id\_WSAEHOSTUNREACH = WSAEHOSTUNREACH *⚡WSAEHOSTUNREACH*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.377. Id\_WSAEINPROGRESS

Id\_WSAEINPROGRESS = WSAEINPROGRESS *⚡WSAEINPROGRESS*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.378. Id\_WSAEINTR

Id\_WSAEINTR = WSAEINTR *⚡WSAEINTR*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.379. Id\_WSAEINVAL

Id\_WSAEINVAL = WSAEINVAL *⚡WSAEINVAL*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.380. Id\_WSAEISCONN

Id\_WSAEISCONN = WSAEISCONN *⚡WSAEISCONN*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.381. Id\_WSAELOOP

Id\_WSAELOOP = WSAELOOP *⚡WSAELOOP*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.382. Id\_WSAEMFILE

Id\_WSAEMFILE = WSAEMFILE *⚡WSAEMFILE*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.383. Id\_WSAEMSGSIZE

Id\_WSAEMSGSIZE = WSAEMSGSIZE *⚡WSAEMSGSIZE*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.384. Id\_WSAENAMETOOLONG

Id\_WSAENAMETOOLONG = WSAENAMETOOLONG *⚡WSAENAMETOOLONG*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.385. Id\_WSAENETDOWN

Id\_WSAENETDOWN = WSAENETDOWN *⚡WSAENETDOWN* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.386. Id\_WSAENETRESET

Id\_WSAENETRESET = WSAENETRESET *⚡WSAENETRESET* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.387. Id\_WSAENETUNREACH

Id\_WSAENETUNREACH = WSAENETUNREACH *⚡WSAENETUNREACH* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.388. Id\_WSAENOBUFFS

Id\_WSAENOBUFFS = WSAENOBUFFS *⚡WSAENOBUFFS* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.389. Id\_WSAENOPROTOOPT

Id\_WSAENOPROTOOPT = WSAENOPROTOOPT *⚡WSAENOPROTOOPT* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.390. Id\_WSAENOTCONN

Id\_WSAENOTCONN = WSAENOTCONN *⚡WSAENOTCONN* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.391. Id\_WSAENOTEMPTY

Id\_WSAENOTEMPTY = WSAENOTEMPTY *⚡WSAENOTEMPTY* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.392. Id\_WSAENOTSOCK

Id\_WSAENOTSOCK = WSAENOTSOCK *⚡WSAENOTSOCK* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.393. Id\_WSAEOPNOTSUPP

Id\_WSAEOPNOTSUPP = WSAEOPNOTSUPP *⚡WSAEOPNOTSUPP* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.394. Id\_WSAEPFNOSUPPORT

Id\_WSAEPFNOSUPPORT = WSAEPFNOSUPPORT *⚡WSAEPFNOSUPPORT* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.395. Id\_WSAEPROTONOSUPPORT

Id\_WSAEPROTONOSUPPORT = WSAEPROTONOSUPPORT *⚡WSAEPROTONOSUPPORT* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.396. Id\_WSAEPROTOTYPE

Id\_WSAEPROTOTYPE = WSAEPROTOTYPE *⚡WSAEPROTOTYPE* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.397. Id\_WSAESHUTDOWN

Id\_WSAESHUTDOWN = WSAESHUTDOWN *⚡WSAESHUTDOWN* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.398. Id\_WSAESOCKTNOSUPPORT

Id\_WSAESOCKTNOSUPPORT = WSAESOCKTNOSUPPORT *⚡WSAESOCKTNOSUPPORT* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.399. Id\_WSAETIMEDOUT

Id\_WSAETIMEDOUT = WSAETIMEDOUT *⚡WSAETIMEDOUT* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.400. Id\_WSAETOOMANYREFS

Id\_WSAETOOMANYREFS = WSAETOOMANYREFS *⚡WSAETOOMANYREFS* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.401. Id\_WSAEWOULDBLOCK

Id\_WSAEWOULDBLOCK = WSAEWOULDBLOCK *⚡WSAEWOULDBLOCK* ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.402. IdBeatsInDay

IdBeatsInDay = 1000 ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.403. IdDayNames

```
IdDayNames: array[0..IdDaysInWeek] of string = ( '', SLongDayNameSun,
SLongDayNameMon, SLongDayNameTue, SLongDayNameWed, SLongDayNameThu,
SLongDayNameFri, SLongDayNameSat );
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.404. IdDayShortNames

```
IdDayShortNames: array[0..IdDaysInWeek] of string = ( '',
SShortDayNameSun, SShortDayNameMon, SShortDayNameTue,
SShortDayNameWed, SShortDayNameThu, SShortDayNameFri, SShortDayNameSat
);
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.405. IdDaysInCentury

```
IdDaysInCentury = (25 * IdDaysInFourYears) - 1;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.406. IdDaysInFourYears

```
IdDaysInFourYears = IdDaysInShortLeapYearCycle
⌊IdDaysInShortLeapYearCycle;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.407. IdDaysInLeapCentury

```
IdDaysInLeapCentury = IdDaysInCentury ⌊IdDaysInCentury + 1;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.408. IdDaysInLeapYear

```
IdDaysInLeapYear = 366;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.409. IdDaysInLeapYearCycle

```
IdDaysInLeapYearCycle = IdDaysInCentury ⌊IdDaysInCentury * 4 + 1;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.410. IdDaysInMonth

```
IdDaysInMonth: array[1..IdMonthsInYear] of byte = ( 31, 28, 31, 30,
31, 30, 31, 31, 30, 31, 30, 31 );
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.411. IdDaysInShortLeapYearCycle

```
IdDaysInShortLeapYearCycle = IdDaysInLeapYear ⚡IdDaysInLeapYear +  
(IdDaysInYear ⚡IdDaysInYear * 3);
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.412. IdDaysInShortNonLeapYearCycle

```
IdDaysInShortNonLeapYearCycle = IdDaysInYear ⚡IdDaysInYear *  
IdYearsInShortLeapYearCycle ⚡IdYearsInShortLeapYearCycle;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.413. IdDaysInWeek

```
IdDaysInWeek = 7;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.414. IdDaysInYear

```
IdDaysInYear = 365;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.415. IdDNSResolver\_ReceiveTimeout

This is the default value for the TIdDNSResolver.ReceiveTimeout property.

```
IdDNSResolver_ReceiveTimeout = 4000;
```

**Description**  
IdDNSResolver\_ReceiveTimeout is the default value for the TIdDNSResolver.ReceiveTimeout property. This is used in the class definition and the constructor. This is done to prevent problems with Delphi's form streaming mechanism. Changing this value will change the default value for that property.

### 3.6.416. iDEFAULTPACKETSIZE

Specifies the default packet size for ICMP data buffers.

```
iDEFAULTPACKETSIZE = 128;
```

**Description**  
iDEFAULTPACKETSIZE is a constant values that specifies the default packet size for ICMP data buffers, like TICMPDataBuffer *⚡TICMPDataBuffer*.

### 3.6.417. iDEFAULTREPLYBUFSIZE

Specifies the default size for ICMP reply buffers.

```
iDEFAULTREPLYBUFSIZE = 1024;
```

**Description**  
iDEFAULTREPLYBUFSIZE is a constant value that specifies the default size for ICMP reply buffers.

### 3.6.418. IdGopherItem\_Binary

Item is a binary file.

IdGopherItem\_Binary = '9';

**Description**

Item is a binary file. The client must read until the TCP connection closes. Beware.

### 3.6.419. IdGopherItem\_BinDOS

DOS binary archive file.

IdGopherItem\_BinDOS = '5';

**Description**

The item is a DOS binary archive of some sort. The client must read until the TCP connection closes. Beware.

### 3.6.420. IdGopherItem\_BinHex

Item is a BinHex Macintosh file.

IdGopherItem\_BinHex = '4';

### 3.6.421. IdGopherItem\_CSO

Item is a CSO phone-book server.

IdGopherItem\_CSO = '2';

### 3.6.422. IdGopherItem\_Directory

Item is a directory or Gopher menu.

IdGopherItem\_Directory = '1';

### 3.6.423. IdGopherItem\_Document

Item is a text-file.

IdGopherItem\_Document = '0';

### 3.6.424. IdGopherItem\_Error

The item is an error message.

IdGopherItem\_Error = '3';

### 3.6.425. IdGopherItem\_GIF

Item is a GIF format graphics file.

IdGopherItem\_GIF = 'g';

### 3.6.426. IdGopherItem\_HTML

The item is an HTML file.

IdGopherItem\_HTML = 'h';

### 3.6.427. IdGopherItem\_Image

The item is a graphic image file.

IdGopherItem\_Image = ':';

**Description**

The item is a graphic image file. The client will decide how to display the item.

### 3.6.428. IdGopherItem\_Image2

The item is a graphic image file.

IdGopherItem\_Image2 = 'I';

**Description**

The item is a graphic image file. IdGopherItem\_Image2 is depreciated so it should only be used in Gopher clients.

### 3.6.429. IdGopherItem\_Information

Information to be displayed to the user.

IdGopherItem\_Information = 'i';

### 3.6.430. IdGopherItem\_MIME

The item is MIME-encoded.

IdGopherItem\_MIME = 'M';

**Description**

This is a MIME encoded item or occasionally a movie (a server should use this only for MIME encoded files and a client should not always assume it is a movie.

### 3.6.431. IdGopherItem\_Movie

This is some type of movie file such as QuickTime or AVI.

IdGopherItem\_Movie = 'i';

**Description**

This is some type of movie file such as QuickTime or AVI.

### 3.6.432. IdGopherItem\_Redundant

Item is a redundant server.

IdGopherItem\_Redundant = '+';

**Description**

Item is a redundant server.

### 3.6.433. IdGopherItem\_Search

This is a gopher search item.

IdGopherItem\_Search = '7';

**Description**

This is a gopher search item.

### 3.6.434. IdGopherItem\_Sound

This is some type of sound such as a .WAV or .AU file.

IdGopherItem\_Sound = '<';

**Description**

This is some type of sound such as a .WAV or .AU file.

### 3.6.435. IdGopherItem\_Sound2

Item is some kind of sound file such as a .AU or .WAV.

IdGopherItem\_Sound2 = 'S';

**Description**

Item is some kind of sound file such as a .AU or .WAV. This was depreciated so it should only be used in Gopher clients.

### 3.6.436. IdGopherItem\_Telnet

Item points to a text-based telnet session.

```
IdGopherItem_Telnet = '8';
```

**Description**

Item points to a text-based telnet session.

### 3.6.437. IdGopherItem\_TN3270

Item points to a text-based tn3270 session.

```
IdGopherItem_TN3270 = 'T';
```

**Description**

Item points to a text-based tn3270 session.

### 3.6.438. IdGopherItem\_UUE

Item is a UNIX uuencoded file.

```
IdGopherItem_UUE = '6';
```

**Description**

Item is a UNIX uuencoded file.

### 3.6.439. IdGopherPlusAbstract

Summary of an item.

```
IdGopherPlusAbstract = '+ABSTRACT:' + EOL ⚡EOL;
```

**Description**

This is the constant for Gopher+ information which is the summary of an item. This is usually in the Gopher+ extended menu items.

### 3.6.440. IdGopherPlusAdmin

This is the constant which starts a Gopher+ Administration block.

```
IdGopherPlusAdmin = '+ADMIN:' + EOL ⚡EOL;
```

**Description**

This is the constant which starts a Gopher+ Administration block.

### 3.6.441. IdGopherPlusAsk

This constant starts a Gopher+ ASK block for a Gopher+ extended menu item.

```
IdGopherPlusAsk = '+ASK:';
```

**Description**

This constant starts a Gopher+ ASK block for a Gopher+ extended menu item.

### 3.6.442. IdGopherPlusAskFileName

This constant in an ASK block is for prompting a user with a filename in a Gopher+ ASK block.

```
IdGopherPlusAskFileName = 'AskF: ';
```

**Description**

This constant in an ASK block is for prompting a user with a filename in a Gopher+ ASK block.

### 3.6.443. IdGopherPlusAskLong

This item is a question in an ASK block which will take several lines of text as a reply.

```
IdGopherPlusAskLong = 'AskL: ';
```

**Description**



This item is a question in an ASK block which will take several lines of text as a reply.

### 3.6.444. IdGopherPlusAskPassword

This item in an ASK block prompts for a line of text which is hidden from the user such as a password prompt.

```
IdGopherPlusAskPassword = 'AskP: ' ;
```

**Description**

This item in an ASK block prompts for a line of text which is hidden from the user such as a password prompt.

### 3.6.445. IdGopherPlusChoose

This is a prompt with where a user selects only one choice from several (such as with a radio-group control).

```
IdGopherPlusChoose = 'Choose: ' ;
```

**Description**

This is a prompt with where a user selects only one choice from several (such as with a radio-group control).

### 3.6.446. IdGopherPlusChooseFile

The user should select only one file for sending to the server.

```
IdGopherPlusChooseFile = 'ChooseF: ' ;
```

**Description**

The user should select only one file for sending to the server. Some administrators may limit this to a single directory. This is only used in an ASK block.

### 3.6.447. IdGopherPlusData\_BeginSign

This Gopher+ reply means that the server will send data to the client until an end-sign defined by IdGopherPlusData\_EndSign *↪IdGopherPlusData\_EndSign*.

```
IdGopherPlusData_BeginSign = '+-1' + EOL ↪EOL ;
```

**Description**

This Gopher+ reply means that the server will send data to the client until an end-sign defined by IdGopherPlusData\_EndSign *↪IdGopherPlusData\_EndSign*.

### 3.6.448. IdGopherPlusData\_EndSign

This is the end sign used for transferring text file with Gopher.

```
IdGopherPlusData_EndSign = EOL ↪EOL + '.' + EOL ↪EOL ;
```

**Description**

This is the end sign used for transferring text file with Gopher. You should also use this in Gopher+ when you send the IdGopherPlusData\_BeginSign *↪IdGopherPlusData\_BeginSign* or the IdGopherPlusData\_ErrorBeginSign *↪IdGopherPlusData\_ErrorBeginSign* reply to a Gopher client.

### 3.6.449. IdGopherPlusData\_ErrorBeginSign

This indicates a Gopher error and that the error message is terminated with IdGopherPlusData\_EndSign *↪IdGopherPlusData\_EndSign*.

```
IdGopherPlusData_ErrorBeginSign = '--1' + EOL ↪EOL ;
```

**Description**

This indicates a Gopher error and that the error message is terminated with IdGopherPlusData\_EndSign *↪IdGopherPlusData\_EndSign*.

### 3.6.450. IdGopherPlusData\_ErrorUnknownSize

This indicates a Gopher+ error and the error reply should be read until the server closes the connection.

```
IdGopherPlusData_ErrorUnknownSize = '--2' + EOL ⚡EOL;
```

**Description**

This indicates a Gopher+ error and the error reply should be read until the server closes the connection.

### 3.6.451. IdGopherPlusData\_UnknownSize

This Gopher+ reply indicates that the Gopher+ client should simply read data until the server closes the connection.

```
IdGopherPlusData_UnknownSize = '+-2' + EOL ⚡EOL;
```

**Description**

This Gopher+ reply indicates that the Gopher+ client should simply read data until the server closes the connection.

### 3.6.452. IdGopherPlusDirectoryInformation

This Gopher+ command indicates that the client wants extended Gopher+ menu information.

```
IdGopherPlusDirectoryInformation = '$';
```

**Description**

This Gopher+ command indicates that the client wants extended Gopher+ menu including all of the blocks for all of the items in the menu.

### 3.6.453. IdGopherPlusError\_ItemMoved

This starts the reply data and means that the Gopher+ item moved someplace else.

```
IdGopherPlusError_ItemMoved = '3';
```

**Description**

This starts the reply data and means that the Gopher+ item moved someplace else. The rest of the data should be a descriptor for the new location.

### 3.6.454. IdGopherPlusError\_NotAvailable

The gopher item is not available.

```
IdGopherPlusError_NotAvailable = '1';
```

**Description**

The gopher item is not available. The rest of the reply is a text description for a user and should indicate the gopher-server administrator's E-Mail address (TIdGopherServer.AdminEmail).

### 3.6.455. IdGopherPlusError\_TryLater

This error code indicates that the Gopher client should retry the operation again.

```
IdGopherPlusError_TryLater = '2';
```

**Description**

This error code indicates that the Gopher client should retry the operation again. The rest of the line should indicate the reason such as "my work load is too high."

### 3.6.456. IdGopherPlusIndicator

This Gopher+ command indicates that the server should send a reply followed by the standard Gopher response.

```
IdGopherPlusIndicator = IdGopherItem_Redundant ⚡IdGopherItem_Redundant;
```

**Description**

This Gopher+ command indicates that the server should send a reply followed by the standard Gopher response.

### 3.6.457. IdGopherPlusInfo

This starts the information Gopher+ block for an item.

```
IdGopherPlusInfo = '+INFO: ';
```

**Description**

This starts the information Gopher+ block for an item. The rest of that block is simply the standard Gopher menu.

### 3.6.458. IdGopherPlusInformation

This Gopher+ command indicates that the client only wants information about the gopher item (the Gopher+ blocks for it).

```
IdGopherPlusInformation = '!';
```

**Description**

This Gopher+ command indicates that the client only wants information about the gopher item (the Gopher+ blocks for it) but not the gopher item itself.

### 3.6.459. IdGopherPlusSelect

In an ASK block, this indicates that the user can select more than one item from a list of choices (such as a bunch of checkboxes on a form).

```
IdGopherPlusSelect = 'Select: ';
```

**Description**

In an ASK block, this indicates that the user can select more than one item from a list of choices (such as a bunch of checkboxes on a form).

### 3.6.460. IdGopherPlusViews

This constant starts the Views list for a Gopher+ item.

```
IdGopherPlusViews = '+VIEWS:' + EOL ⚡EOL;
```

**Description**

This constant starts the Views list for a Gopher+ item.

### 3.6.461. IdHoursInDay

```
IdHoursInDay = 24;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.462. IdHoursInHalfDay

```
IdHoursInHalfDay = IdHoursInDay ⚡IdHoursInDay div 2;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.463. IdMillisecondsInDay

```
IdMillisecondsInDay = IdSecondsInDay ⚡IdSecondsInDay *  
IdMillisecondsInSecond;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.464. IdMillisecondsInHour

```
IdMillisecondsInHour = IdSecondsInHour ⚡IdSecondsInHour *  
IdMillisecondsInSecond;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.465. IdMillisecondsInMinute

```
IdMillisecondsInMinute = IdSecondsInMinute ⚡IdSecondsInMinute *  
IdMillisecondsInSecond;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.466. IdMilliSecondsInSecond

```
IdMilliSecondsInSecond = 1000;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.467. IdMillisecondsInWeek

```
IdMillisecondsInWeek = IdSecondsInWeek ⚡IdSecondsInWeek *  
IdMillisecondsInSecond;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.468. IdMinutesInHour

```
IdMinutesInHour = 60;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.469. IdMonthNames

```
IdMonthNames: array[0..IdMonthsInYear] of string = ( '',  
SLongMonthNameJan, SLongMonthNameFeb, SLongMonthNameMar,  
SLongMonthNameApr, SLongMonthNameMay, SLongMonthNameJun,  
SLongMonthNameJul, SLongMonthNameAug, SLongMonthNameSep,  
SLongMonthNameOct, SLongMonthNameNov, SLongMonthNameDec );
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.470. IdMonthShortNames

```
IdMonthShortNames: array[0..IdMonthsInYear] of string = ( '',  
SShortMonthNameJan, SShortMonthNameFeb, SShortMonthNameMar,  
SShortMonthNameApr, SShortMonthNameMay, SShortMonthNameJun,  
SShortMonthNameJul, SShortMonthNameAug, SShortMonthNameSep,  
SShortMonthNameOct, SShortMonthNameNov, SShortMonthNameDec );
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.471. IdMonthsInYear

```
IdMonthsInYear = 12;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.472. IdPORT\_AUTH

Port number for authentication services.

```
IdPORT_AUTH = 113;
```

**Description**

IdPORT\_AUTH is a constant Integer value that represents the port number used for authentication services.

3.6.473. IdPORT\_CHARGEN

Port number for the Chargen protocol.

IdPORT\_CHARGEN = 19;

**Description**

IdPORT\_CHARGEN is the constant Integer value used to represent the port number for the Chargen protocol.

3.6.474. IdPORT\_DAYTIME

Port number for the DayTime protocol.

IdPORT\_DAYTIME = 13;

**Description**

IdPORT\_DAYTIME is the constant Integer value used to represent the port number used by the DayTime protocol.  
IdPORT\_DAYTIME is used by the TIdDayTime *↯* *TIdDayTime* and TIdDayTimeServer *↯* *TIdDayTimeServer* protocol implementations.

3.6.475. IdPORT\_DICT

Port number for the DICT protocol.

IdPORT\_DICT = 2628;

**Description**

IdPORT\_DICT is the constant Integer value used to represent the port number for the DICT protocol.  
IdPORT\_DICT is used by TIdDICTServer *↯* *TIdDICTServer* protocol implementation.

3.6.476. IdPORT\_DISCARD

Port number for the Discard protocol.

IdPORT\_DISCARD = 9;

**Description**

IdPORT\_DISCARD is the constant Integer value used to represent the port number used by the Discard protocol.  
IdPORT\_DISCARD is used by the TIdDiscardServer protocol implementation.

3.6.477. IdPORT\_DOMAIN

Port number for the DNS protocol.

IdPORT\_DOMAIN = 53;

**Description**

IdPORT\_DOMAIN is the constant Integer value that represents the default port number used by the DNS protocol.  
IdPORT\_DOMAIN is used by the TIdDNSResolver *↯* *TIdDNSResolver* protocol implementation.

3.6.478. IdPORT\_ECHO

Port number for the Echo protocol.

IdPORT\_ECHO = 7;

**Description**

IdPORT\_ECHO is the constant Integer value that represents the port number used by the Echo protocol.  
IdPORT\_ECHO is used by the TIdEcho *↯* *TIdEcho* and TIdEchoServer protocol implementations.

### 3.6.479. IdPORT\_FINGER

Port number for the Finger protocol.

```
IdPORT_FINGER = 79;
```

**Description**

IdPORT\_FINGER is the constant Integer value that represents the port number used by the Finger protocol.  
IdPORT\_FINGER is used by the TIdFinger *↗TIdFinger* and TIdFingerServer *↗TIdFingerServer* protocol implementations.

### 3.6.480. IdPORT\_FTP

Port number for the FTP protocol.

```
IdPORT_FTP = 21;
```

**Description**

IdPORT\_FTP is the constant Integer value that represents the default port number used by the File Transfer Protocol.  
IdPORT\_FTP is used by the TIdFtp protocol implementation.

### 3.6.481. IdPORT\_GOPHER

Port number for the Gopher protocol.

```
IdPORT_GOPHER = 70;
```

**Description**

IdPORT\_GOPHER is the constant Integer value that represents the port number used by the Gopher protocol.  
IdPORT\_GOPHER is used by the TIdGopher *↗TIdGopher* and TIdGopherServer *↗TIdGopherServer* protocol implementations.

### 3.6.482. IdPORT\_HOSTNAME

Port number for the HostName protocol.

```
IdPORT_HOSTNAME = 101;
```

**Description**

IdPORT\_HOSTNAME is the constant Integer value that represents the port number used by the HostName protocol.  
IdPORT\_HOSTNAME is used by the TIdHostNameServer *↗TIdHostNameServer* protocol implementation.

### 3.6.483. IdPORT\_HTTP

Port number for the HTTP protocol.

```
IdPORT_HTTP = 80;
```

**Description**

IdPORT\_HTTP is the constant Integer value that represents the port number used by the Hyper Text Transfer Protocol (HTTP) protocol.  
IdPORT\_HTTP is used by the TIdHTTP *↗TIdHTTP* and TIdHTTPServer *↗TIdHTTPServer* protocol implementations.

### 3.6.484. IdPORT\_IMAP4

Port number for the IMAP protocol.

```
IdPORT_IMAP4 = 143;
```

**Description**

IdPORT\_IMAP4 is the constant Integer value that represents the default port number used by the Internet Message Access Protocol (IMAP) protocol.  
IdPORT\_IMAP4 is used by the TIdIMAP4Server *↗TIdIMAP4Server* protocol implementation.

### 3.6.485. IdPORT\_IRC

Port number for the IRC protocol.

```
IdPORT_IRC = 6667;
```

**Description**

IdPORT\_IRC is the constant Integer value that represents the default port number used by the Internet Relay Chat (IRC) protocol.  
IdPORT\_IRC is used by the TIdIRCServer *↯* *TIdIRCServer* protocol implementation.

### 3.6.486. IdPORT\_LPD

Port number for the LPD protocol.

```
IdPORT_LPD = 515;
```

**Description**

IdPORT\_LPD is the constant Integer value that represents the port number used by the Line Printer Deamon (LPD) protocol.

### 3.6.487. IdPORT\_NETSTAT

Port number for the Netstat protocol.

```
IdPORT_NETSTAT = 15;
```

**Description**

IdPORT\_NETSTAT is the constant Integer value that represents the default port number used by the NetStat protocol.

### 3.6.488. IdPORT\_NNTP

Port number for the NNTP protocol.

```
IdPORT_NNTP = 119;
```

**Description**

IdPORT\_NNTP is the constant Integer value that represents the default port number used by the Network News Transfer Protocol (NNTP) protocol.  
IdPORT\_NNTP is used by the TIdNNTP *↯* *TIdNNTP* and TIdNNTPServer *↯* *TIdNNTPServer* protocol implementations.

### 3.6.489. IdPORT\_POP2

Port number for the POP2 protocol.

```
IdPORT_POP2 = 109;
```

**Description**

IdPORT\_POP2 is the constant Integer value that represents the port number used by the Post Office Protocol 2 (POP2) protocol.

### 3.6.490. IdPORT\_POP3

Port number for the POP3 protocol.

```
IdPORT_POP3 = 110;
```

**Description**

IdPORT\_POP3 is the constant Integer value that represents the default port number used by the Post Office Protocol 3 (POP3) protocol.  
IdPORT\_POP3 is used by the TIdPOP3 *↯* *TIdPOP3* protocol implementation.

### 3.6.491. IdPORT\_QOTD

Port number for the QUOTD protocol.

```
IdPORT_QOTD = 17;
```

**Description**

IdPORT\_QOTD is the constant Integer value that represents the default port number used by

the Quote of the Day (QUOTD) protocol.  
IdPORT\_QOTD is used by the TIdQUOTD and TIdQUOTDServer protocol implementations.

3.6.492. IdPORT\_SMTP

Port number for the SMTP protocol.

IdPORT\_SMTP = 25;  
**Description**  
IdPORT\_SMTP is the constant Integer value that represents the port number used by the Simple Mail Transfer Protocol (SMTP) protocol.  
IdPORT\_SMTP is used by the TIdSMTP *↯* TIdSMTP protocol implementation.

3.6.493. IdPORT\_Sntp

Port number for the Sntp protocol.

IdPORT\_Sntp = 123;  
**Description**  
IdPORT\_Sntp is the constant Integer value that represents the port number used by the Simple Network Time Protocol (SNTP) protocol.  
IdPORT\_Sntp is used by the TIdSntp *↯* TIdSNTP protocol implementation.

3.6.494. IdPORT\_SSL

Port number for the HTTPS protocol.

IdPORT\_SSL = 443;  
**Description**  
IdPORT\_SSL is the constant Integer value that represents the default port number used by the HTTP Protocol over the Secure Socket *↯* Socket Layer (HTTPS) protocol.  
IdPORT\_SSL is used by the TIdHTTP *↯* TIdHTTP and TIdHTTPServer *↯* TIdHTTPServer protocol implementations.

3.6.495. IdPORT\_SYSTAT

Port number for the Sysstat protocol.

IdPORT\_SYSTAT = 11;  
**Description**  
IdPORT\_SYSTAT is the constant Integer value that represents the default port number used by the Sysstat (Users) protocol.

3.6.496. IdPORT\_Telnet

Default port for the Telnet protocol.

IdPORT\_TELNET = 23;  
**Description**  
IdPORT\_TELNET is the constant Integer value used to represent the default port for the Telnet protocol. IdPORT\_TELNET is used by TIdTelnet *↯* TIdTelnet and TIdTelnetServer *↯* TIdTelnetServer.

3.6.497. IdPORT\_TFTP

Port number for the TFTP protocol.

IdPORT\_TFTP = 69;  
**Description**  
IdPORT\_TFTP is the constant Integer value that represents the default port number used by the Trivial File Transfer Protocol (TFTP) protocol.  
IdPORT\_TFTP is used by the TIdTFTP and TIdTFTPServer protocol implementations.

3.6.498. IdPORT\_TIME

Port number for the Time protocol.



```
IdPORT_TIME = 37;
```

**Description**

IdPORT\_TIME is the constant Integer value that represents the default port number used by the Time protocol.  
IdPORT\_TIME is used by the TIdTime *↯TIdTime* and TIdTimeServer *↯TIdTimeServer* protocol implementations.

3.6.499. IdPORT\_WHOIS

Port number for the Whois protocol.

```
IdPORT_WHOIS = 43;
```

**Description**

IdPORT\_WHOIS is the constant Integer value that represents the default port number used by the Whois protocol.  
IdPORT\_WHOIS is used by the TIdWhois and TIdWhoisServer *↯TIdWhoisServer* protocol implementations.

3.6.500. IdSecondsInDay

```
IdSecondsInDay = IdSecondsInHour ↯IdSecondsInHour * IdHoursInDay  
↯IdHoursInDay;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.501. IdSecondsInHalfDay

```
IdSecondsInHalfDay = IdSecondsInHour ↯IdSecondsInHour *  
IdHoursInHalfDay ↯IdHoursInHalfDay;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.502. IdSecondsInHour

```
IdSecondsInHour = IdSecondsInMinute ↯IdSecondsInMinute *  
IdMinutesInHour ↯IdMinutesInHour;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.503. IdSecondsInLeapYear

```
IdSecondsInLeapYear = IdSecondsInDay ↯IdSecondsInDay *  
IdDaysInLeapYear ↯IdDaysInLeapYear;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.504. IdSecondsInMinute

```
IdSecondsInMinute = 60;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.505. IdSecondsInWeek

```
IdSecondsInWeek = IdDaysInWeek ↯IdDaysInWeek * IdSecondsInDay  
↯IdSecondsInDay;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.506. IdSecondsInYear

```
IdSecondsInYear = IdSecondsInDay ⚡IdSecondsInDay * IdDaysInYear  
⚡IdDaysInYear;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.507. IdStati

Format strings for constructing connection status text messages.

```
IdStati: array[TIdStatus] of string = ( RSStatusResolving  
⚡RSStatusResolving, RSStatusConnecting ⚡RSStatusConnecting,  
RSStatusConnected ⚡RSStatusConnected, RSStatusDisconnecting  
⚡RSStatusDisconnecting, RSStatusDisconnected ⚡RSStatusDisconnected,  
RSStatusText );
```

**Description**

IdStati is an array of strings which are used as the format specifier for constructing the status message of the TIdStatusEvent *⚡TIdStatusEvent* event. There is one array element in IdStati for each of the values in the TIdStatus *⚡TIdStatus* enumerated type.  
Do not change the order of the strings in this array or the TIdComponent.OnStatus event could return incorrect status strings.

### 3.6.508. IdTimeoutDefault

Indicates that the default time-out value should be used.

```
IdTimeoutDefault = -1;
```

**Description**

IdTimeoutDefault is a constant Integer value that indicates the default time-out value should be used in Indy components.

### 3.6.509. IdTimeoutInfinite

Indicates that an infinite time-out value should be used.

```
IdTimeoutInfinite = -2;
```

**Description**

IdTimeoutInfinite is a constant Integer value that indicates that an infinite time-out value should be used by Indy components.

### 3.6.510. IdYearsInCentury

```
IdYearsInCentury = 100;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.511. IdYearsInLeapYearCycle

```
IdYearsInLeapYearCycle = 400;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.512. IdYearsInShortLeapYearCycle

```
IdYearsInShortLeapYearCycle = 4;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.513. IMAPCommands

```
IMAPCommands: Array [1..25] Of String = ( 'CAPABILITY', 'NOOP',  
'LOGOUT', 'AUTHENTICATE', 'LOGIN', 'SELECT', 'EXAMINE', 'CREATE',  
'DELETE', 'RENAME', 'SUBSCRIBE', 'UNSUBSCRIBE', 'LIST', 'LSUB',  
'STATUS', 'APPEND', 'CHECK', 'CLOSE', 'EXPUNGE', 'SEARCH', 'FETCH',  
'STORE', 'COPY', 'UID', 'X');
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.514. IMPLINK\_HIGHEXPER

```
IMPLINK_HIGHEXPER = 158;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.515. IMPLINK\_IP

```
IMPLINK_IP = 155;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.516. IMPLINK\_LOWEXPER

```
IMPLINK_LOWEXPER = 156;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.517. INVALID\_SOCKET

```
INVALID_SOCKET = TSocket(NOT(0));
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.518. IOC\_IN

```
IOC_IN = $80000000;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.519. IOC\_INOUT

```
IOC_INOUT = (IOC_IN ≠IOC_IN or IOC_OUT);
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.520. IOC\_OUT

```
IOC_OUT = $40000000;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.521. IOC\_VOID

```
IOC_VOID = $20000000;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.522. IOCPARM\_MASK

IOCPARM\_MASK = \$7f;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.523. IP\_ADD\_MEMBERSHIP

IP\_ADD\_MEMBERSHIP = 5;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.524. IP\_DEFAULT\_MULTICAST\_LOOP

IP\_DEFAULT\_MULTICAST\_LOOP = 1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.525. IP\_DEFAULT\_MULTICAST\_TTL

IP\_DEFAULT\_MULTICAST\_TTL = 1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.526. IP\_DONTFRAGMENT

IP\_DONTFRAGMENT = 9;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.527. IP\_DROP\_MEMBERSHIP

IP\_DROP\_MEMBERSHIP = 6;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.528. IP\_MAX\_MEMBERSHIPS

IP\_MAX\_MEMBERSHIPS = 20;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.529. IP\_MULTICAST\_IF

IP\_MULTICAST\_IF = 2;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.530. IP\_MULTICAST\_LOOP

IP\_MULTICAST\_LOOP = 4;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.531. IP\_MULTICAST\_TTL

```
IP_MULTICAST_TTL = 3;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.532. IP\_OPTIONS

```
IP_OPTIONS = 1;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.533. IP\_TOS

```
IP_TOS = 8;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.534. IP\_TTL

```
IP_TTL = 7;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.535. IP\_WATCH\_ACTIVE

Default value for the TIdIPWatch.Active property.

```
IP_WATCH_ACTIVE = False;
```

Description

IP\_WATCH\_ACTIVE is the default value for the TIdIPWatch.Active property. IP\_WATCH\_ACTIVE is used in the class definition and the constructor to prevent problems with Delphi's form streaming mechanism. Changing the value of IP\_WATCH\_ACTIVE will change the default value for that property.

3.6.536. IP\_WATCH\_HIST\_ENABLED

Default value for the TIdIPWatch.HistoryEnabled property.

```
IP_WATCH_HIST_ENABLED = True;
```

Description

IP\_WATCH\_HIST\_ENABLED is the default value for the TIdIPWatch.HistoryEnabled property. IP\_WATCH\_HIST\_ENABLED is used in the class definition and the constructor to prevent problems with Delphi's form streaming mechanism. Changing the value of IP\_WATCH\_HIST\_ENABLED will change the default value for that property.

3.6.537. IP\_WATCH\_HIST\_FILENAME

Identifies the default history storage file.

```
IP_WATCH_HIST_FILENAME = 'iphist.dat';
```

Description

IP\_WATCH\_HIST\_FILENAME is a constant to define the default filename used to save and restore the history list from TIdIPWatch *↯* *TIdIPWatch*. The default value is **IPHIST.DAT**.

### 3.6.538. IP\_WATCH\_HIST\_MAX

Identifies the maximum entries in IP history.

```
IP_WATCH_HIST_MAX = 25;
```

**Description**

IP\_WATCH\_HIST\_MAX is a constant used to define the default value for the maximum number of entries allowed in TIdIPWatch ~~↗~~*TIdIPWatch*. The default value is **25**.

### 3.6.539. IP\_WATCH\_INTERVAL

Default value for the TIdIPWatch.WatchInterval property.

```
IP_WATCH_INTERVAL = 1000;
```

**Description**

IP\_WATCH\_INTERVAL is the default value for the TIdIPWatch.WatchInterval property. IP\_WATCH\_INTERVAL is used in the class definition and the constructor to prevent problems with Delphi's form streaming mechanism. Changing the value of IP\_WATCH\_INTERVAL will change the default value for that property.

### 3.6.540. IPPORT\_RESERVED

```
IPPORT_RESERVED = 1024;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.541. IPPROTO\_GGP

```
IPPROTO_GGP = 3;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.542. IPPROTO\_ICMP

```
IPPROTO_ICMP = 1;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.543. IPPROTO\_IDP

```
IPPROTO_IDP = 22;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.544. IPPROTO\_IGMP

```
IPPROTO_IGMP = 2;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.545. IPPROTO\_IP

```
IPPROTO_IP = 0;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.546. IPPROTO\_MAX

```
IPPROTO_MAX = 256;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.547. IPPROTO\_ND

IPPROTO\_ND = 77;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.548. IPPROTO\_PUP

IPPROTO\_PUP = 12;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.549. IPPROTO\_RAW

IPPROTO\_RAW = 255;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.550. IPPROTO\_TCP

IPPROTO\_TCP = 6;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.551. IPPROTO\_UDP

IPPROTO\_UDP = 17;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.552. kana\_tbl

Half-width Katakana characters.

kana\_tbl: **array**[\$A1..#\$DF] **of** Word = (  
\$2123,\$2156,\$2157,\$2122,\$2126,\$2572,\$2521,\$2523,\$2525,\$2527,  
\$2529,\$2563,\$2565,\$2567,\$2543,\$213C,\$2522,\$2524,\$2526,\$2528,  
\$252A,\$252B,\$252D,\$252F,\$2531,\$2533,\$2535,\$2537,\$2539,\$253B,  
\$253D,\$253F,\$2541,\$2544,\$2546,\$2548,\$254A,\$254B,\$254C,\$254D,  
\$254E,\$254F,\$2552,\$2555,\$2558,\$255B,\$255E,\$255F,\$2560,\$2561,  
\$2562,\$2564,\$2566,\$2568,\$2569,\$256A,\$256B,\$256C,\$256D,\$256F,  
\$2573,\$212B,\$212C);

**Description**

kana\_tbl is a Word array that represents the half-width Katakana characters in the range #A1 through #DF. kana\_tbl is used in ISO-2022-JP character set encoding.

### 3.6.553. KnownCommands

KnownCommands: **Array** [1..26] **of string** = ('ARTICLE', 'BODY', 'HEAD',  
'STAT', 'GROUP', 'LIST', 'HELP', 'IHAVE', 'LAST', 'NEWGROUPS',  
'NEWNEWS', 'NEXT', 'POST', 'QUIT', 'SLAVE', 'AUTHINFO', 'XOVER',  
'XHDR', 'DATE', 'LISTGROUP', 'MODE', 'TAKETHIS', 'CHECK', 'XTHREAD',  
'XGTITLE', 'XPAT' );

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.554. LF

Represents the Line Feed character.

LF = #10;

**Description**

LF is a constant value that represents the Line Feed character used by many Internet protocols.

### 3.6.555. MAX\_PACKET\_SIZE

Specifies the maximum packet size for an ICMP response message.

MAX\_PACKET\_SIZE = 1024;

**Description**

MAX\_PACKET\_SIZE is a constant value that specifies the maximum packet size for an ICMP response message, like the TCharBuf *↯* TCharBuf receive buffer used by TIdICMPClient.

### 3.6.556. MAXGETHOSTSTRUCT

MAXGETHOSTSTRUCT = 1024;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.557. MaxMIMEBinToASCIIType

MaxMIMEBinToASCIIType = 2;

**Description**

MaxMIMEBinToASCIIType represents the number of entries to allocate for MIME encoding types.

### 3.6.558. MaxMIMECompressType

MaxMIMECompressType = 0;

**Description**

MaxMIMECompressType represents the number of entries to allocate for MIME Compression type constants.

### 3.6.559. MaxMIMEEncType

MaxMIMEEncType = MaxMIMEBinToASCIIType *↯* MaxMIMEBinToASCIIType +  
MaxMIMEMessageDigestType *↯* MaxMIMEMessageDigestType + 1 +  
MaxMIMECompressType *↯* MaxMIMECompressType + 1;

**Description**

MaxMIMEEncType represents the number of MIME encoding types to allocate for encoding type constants.

### 3.6.560. MaxMIMEMessageDigestType

MaxMIMEMessageDigestType = 3;

**Description**

MaxMIMEMessageDigestType represents the number of entried to allocate for Message Digest constants.

### 3.6.561. MaxMIMESubTypes

MaxMIMESubTypes = 1;

**Description**

MaxMIMESubTypes represents the number of entries to allocate for MIME subtype constants.



### 3.6.562. MaxMIMEType

MaxMIMEType = 6;

**Description**

MaxMIMEType represents the number of entries to allocate for constants constructed from valid MIME media type and subtype combinations.

### 3.6.563. maxPriv

Maximum user permissions for UU- or XX- Encode/Decode.

maxPriv = 799;

**Description**

maxPriv represents the maximum user permissions allowable for a UU- or XX- encode/decode operation in the UNIX environment.

### 3.6.564. MaxWord

Specifies the maximum value for a Word data type.

MaxWord = High(Word);

**Description**

MaxWord specifies the maximum value for a Word data type.

### 3.6.565. MIME7Bit

MIME type for 7-bit text.

MIME7Bit = '7bit';

**Description**

MIME7Bit is a constant value that represents the MIME type for 7-bit text.

### 3.6.566. MIMEEncBase64

MIMEEncBase64 = 'base64';

**Description**

MIMEEncBase64 represents the MIME BASE64 encoding type.

### 3.6.567. MIMEEncNISTSHA

MIMEEncNISTSHA = MIMEXVal &MIMEXVal + 'nist-sha';

**Description**

MIMEEncNISTSHA represents the MIME subtype for NIST Secure Hash Algorithm (SHA).

### 3.6.568. MIMEEncRLECompress

MIMEEncRLECompress = MIMEXVal &MIMEXVal + 'rle-compress';

**Description**

MIMEEncRLECompress represents the MIME subtype for Run-Length Encoded (RLE) compression.

### 3.6.569. MIMEEncRSAMD2

MIMEEncRSAMD2 = MIMEXVal &MIMEXVal + 'rsa-md2';

**Description**

MIMEEncRSAMD2 represents the MIME subtype for RSA Message Digest 2 (MD2).

### 3.6.570. MIMEEncRSAMD4

MIMEEncRSAMD4 = MIMEXVal &MIMEXVal + 'rsa-md4';

**Description**

MIMEEncRSAMD4 represents the MIME subtype for RSA Message Digest 4 (MD4).

### 3.6.571. MIMEEncRSAMD5

MIMEEncRSAMD5 = MIMEXVal *⚡MIMEXVal* + 'rsa-md5';

**Description**

MIMEEncRSAMD5 represents the MIME subtype for RSA Message Digest 5 (MD5).

### 3.6.572. MIMEEncUUEncode

MIMEEncUUEncode = MIMEXVal *⚡MIMEXVal* + 'uu';

**Description**

MIMEEncUUEncode represents the MIME type for UU-Encoded values.

### 3.6.573. MIMEEncXXEncode

MIMEEncXXEncode = MIMEXVal *⚡MIMEXVal* + 'xx';

**Description**

MIMEEncXXEncode represents the MIME type for XX-Encoded values.

### 3.6.574. MIMEFullApplicationOctetStream

MIMEFullApplicationOctetStream = MIMETYPEapplication  
*⚡MIMETYPEapplication* + MIMESubOctetStream *⚡MIMESubOctetStream*;

**Description**

MIMEFullApplicationOctetStream represents the constant MIME media type and subtype values for applications data in 8-bit format.

### 3.6.575. MIMEGenericText

MIME type for generic text.

MIMEGenericText = 'text/';

**Description**

MIMEGenericText is the constant value used to represent the partial MIME type for generic text.

### 3.6.576. MIMESplit

MIMESplit = '/';

**Description**

MIMESplit represents the media type and subtype seperator character.

### 3.6.577. MIMESubMacBinHex40

MIMESubMacBinHex40 = 'mac-binhex40';

**Description**

MIMESubMacBinHex40 represents the MIME subtype for Mac BinHex files.

### 3.6.578. MIMESubOctetStream

MIMESubOctetStream = 'octet-stream';

**Description**

MIMESubOctetStream represents the MIME subtype for octet-stream (8-bit) content.

### 3.6.579. MIMETYPEapplication

MIMETYPEapplication = 'application' + MIMESplit *⚡MIMESplit*;

**Description**

MIMETYPEapplication represents the MIME type for application-specific files.

### 3.6.580. MIMETYPEaudio

MIMETYPEaudio = 'audio' + MIMESplit *⚡MIMESplit*;

**Description**

MIMETYPEaudio represents the MIME type for Audio files.

### 3.6.581. MIMETypedImage

```
MIMETypedImage = 'image' + MIMESplit ⚡MIMESplit;
```

**Description**  
MIMETypedImage represents the MIME type for image files.

### 3.6.582. MIMETYPEMessage

```
MIMETYPEMessage = 'message' + MIMESplit ⚡MIMESplit;
```

**Description**  
MIMETYPEMessage represents the MIME type for Internet Message files.

### 3.6.583. MIMETYPEMultipart

```
MIMETYPEMultipart = 'multipart' + MIMESplit ⚡MIMESplit;
```

**Description**  
MIMETYPEMultipart represents the MIME type for multipart files.

### 3.6.584. MIMETYPEText

```
MIMETYPEText = 'text' + MIMESplit ⚡MIMESplit;
```

**Description**  
MIMETYPEText represents the MIME type for tetxt files.

### 3.6.585. MIMETYPEVideo

```
MIMETYPEVideo = 'video' + MIMESplit ⚡MIMESplit;
```

**Description**  
MIMETYPEVideo represents the MIME type for Video files.

### 3.6.586. MIMEXVal

```
MIMEXVal = 'x-';
```

**Description**  
MIMEXVal represents the MIMEType for MIME type extensions.

### 3.6.587. minPriv

Minimum user permissions for UU- or XX- Encode/Decode.

```
minPriv = 600;
```

**Description**  
minPriv represents the minimum user permissions needed to perform a UU- or XX- encode/decode operation in the UNIX environment.

### 3.6.588. MSG\_DONTROUTE

```
MSG_DONTROUTE = $4;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.589. MSG\_MAXIOVLEN

```
MSG_MAXIOVLEN = 16;
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.590. MSG\_OOB

```
MSG_OOB = $1;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.591. MSG\_PARTIAL

```
MSG_PARTIAL = $8000;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.592. MSG\_PEEK

```
MSG_PEEK = $2;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.593. MultiPartAlternativeBoundary

Identifies the boundary marker for the alternate form of the message body.

```
MultiPartAlternativeBoundary =
'=_NextPart_2altrfkindysadvnqw3nerasdf';
```

Description

MultiPartAlternativeBoundary is the constant value used to represent the boundary marker for MIME messages with the Content-Type: multipart/alternative. MultiPartAlternativeBoundary identifies the end of the designated content range. MultiPartAlternativeBoundary is used by TIdMessage *✎TIdMessage* in the SendBody method to generate the contents of a message, and in SendHeader for messages with a Content-ID

header.

3.6.594. MultiPartBoundary

Identifies the boundary marker for multipart MIME messages.

```
MultiPartBoundary = '=_NextPart_2rfkindysadvnqw3nerasdf';
```

Description

MultiPartBoundary is the constant value used to represent the boundary marker for MIME messages. MultiPartBoundary is used to identify both the beginning and the end of the designated content range. MultiPartBoundary is used by TIdMessage *✎TIdMessage* in the SendHeader and SendBody methods to generate the boundary markers for attachments or plain text.

3.6.595. MultiPartRelatedBoundary

Identifies the boundary marker for MIME messages with inline content.

```
MultiPartRelatedBoundary = '=_NextPart_2relrfksadvnqindyw3nerasdf';
```

Description

MultiPartRelatedBoundary is a constant value used to indicate the boundary marker for multipart MIME messages with a content range identifies by the Content-ID header. MultiPartRelatedBoundary is used in TIdMessage *✎TIdMessage* by the SendBody and SendHeader methods to mark the start of the content range.

3.6.596. NO\_ADDRESS

```
NO_ADDRESS = WSANO_ADDRESS ✎WSANO_ADDRESS;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.597. NO\_DATA

```
NO_DATA = WSANO_DATA  WSANO_DATA;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.598. NO\_RECOVERY

```
NO_RECOVERY = WSANO_RECOVERY  WSANO_RECOVERY;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.599. NTPMaxInt

NTP conversion value.

```
NTPMaxInt = 4294967297.0;
```

Description

NTPMaxInt is a constant value used in translating NTP seconds and fractional seconds to/from a native Delphi TDateTime value.

3.6.600. NumberOfClientsType

Number of client connections for the Tunnel component.

```
NumberOfClientsType = 7;
```

Description

NumberOfClientsType is a constant value that represents the statistical category for the number of client connections for the Tunnel component.

3.6.601. NumberOfConnectionsType

Number of thread or service connections for the Tunnel component.

```
NumberOfConnectionsType = 1;
```

Description

NumberOfConnectionsType is a constant value that represents the number of thread or service connections for the Tunnel component.

3.6.602. NumberOfPacketsType

Number of packets handled by the Tunnel component.

```
NumberOfPacketsType = 2;
```

Description

NumberOfPacketsType is a constant value that represents the number of packets handled by the Tunnel component.

3.6.603. NumberOfServicesType

Number of distinct service connections for he Tunnel component.

```
NumberOfServicesType = 9;
```

Description

NumberOfServicesType is a constant value that represents the number of distinct service connections for the Tunnel component.

### 3.6.604. NumberOfSlavesType

Number of slave connections for the Tunnel component.

NumberOfSlavesType = 8;

**Description**

NumberOfSlavesType is a constant value that represents the number of slave connections for the Tunnel component.

### 3.6.605. PF\_APPLETALK

PF\_APPLETALK = AF\_APPLETALK *⚡AF\_APPLETALK*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.606. PF\_BAN

PF\_BAN = AF\_BAN *⚡AF\_BAN*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.607. PF\_CCITT

PF\_CCITT = AF\_CCITT *⚡AF\_CCITT*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.608. PF\_CHAOS

PF\_CHAOS = AF\_CHAOS *⚡AF\_CHAOS*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.609. PF\_DATAKIT

PF\_DATAKIT = AF\_DATAKIT *⚡AF\_DATAKIT*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.610. PF\_DECnet

PF\_DECnet = AF\_DECnet *⚡AF\_DECnet*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.611. PF\_DLI

PF\_DLI = AF\_DLI *⚡AF\_DLI*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.612. PF\_ECMA

PF\_ECMA = AF\_ECMA *⚡AF\_ECMA*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.613. PF\_FIREFOX

```
PF_FIREFOX = AF_FIREFOX  ⚡AF_FIREFOX;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.614. PF\_HYLINK

```
PF_HYLINK = AF_HYLINK  ⚡AF_HYLINK;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.615. PF\_IMPLINK

```
PF_IMPLINK = AF_IMPLINK  ⚡AF_IMPLINK;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.616. PF\_INET

```
PF_INET = AF_INET  ⚡AF_INET;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.617. PF\_IPX

```
PF_IPX = AF_IPX  ⚡AF_IPX;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.618. PF\_ISO

```
PF_ISO = AF_ISO  ⚡AF_ISO;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.619. PF\_LAT

```
PF_LAT = AF_LAT  ⚡AF_LAT;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.620. PF\_MAX

```
PF_MAX = AF_MAX  ⚡AF_MAX;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.621. PF\_NS

```
PF_NS = AF_NS  ⚡AF_NS;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.622. PF\_OSI

PF\_OSI = AF\_OSI *⚡AF\_OSI*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.623. PF\_PUP

PF\_PUP = AF\_PUP *⚡AF\_PUP*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.624. PF\_SNA

PF\_SNA = AF\_SNA *⚡AF\_SNA*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.625. PF\_UNIX

PF\_UNIX = AF\_UNIX *⚡AF\_UNIX*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.626. PF\_UNKNOWN1

PF\_UNKNOWN1 = AF\_UNKNOWN1 *⚡AF\_UNKNOWN1*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.627. PF\_UNSPEC

PF\_UNSPEC = AF\_UNSPEC *⚡AF\_UNSPEC*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.628. PF\_VOICEVIEW

PF\_VOICEVIEW = AF\_VOICEVIEW *⚡AF\_VOICEVIEW*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.629. RSAAboutBoxCompName

RSAAboutBoxCompName = 'Internet Direct (Indy)';

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.630. RSAAboutBoxCopyright

RSAAboutBoxCopyright = 'Copyright © 1993 - 2001'#13#10 + 'Kudzu (Chad Z. Hower)'#13#10 + 'and the'#13#10 + 'Indy Pit Crew';



**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.631. RSAAboutBoxIndyWebsite

```
RSAAboutBoxIndyWebsite = 'http://www.nevrona.com/indy/';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.632. RSAAboutBoxPleaseVisit

```
RSAAboutBoxPleaseVisit = 'For the latest updates and information  
please visit:';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.633. RSAAboutBoxVersion

```
RSAAboutBoxVersion = 'Version %s';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.634. RSAAboutCreditsCoCoordinator

```
RSAAboutCreditsCoCoordinator = 'Project Co-Coordinator';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.635. RSAAboutCreditsCoordinator

```
RSAAboutCreditsCoordinator = 'Project Coordinator';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.636. RSAAboutFormCaption

```
RSAAboutFormCaption = 'About';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.637. RSAAboutMenuItemName

```
RSAAboutMenuItemName = 'About Internet &Direct (Indy) %s...';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.638. RSAcceptWaitCannotBeModifiedWhileServerIsActive

```
RSAcceptWaitCannotBeModifiedWhileServerIsActive = 'AcceptWait property  
cannot be modified while server is active.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.639. RSAAlreadyConnected

```
RSAlreadyConnected = 'Already connected.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.640. RSByteIndexOutOfBounds

```
RSByteIndexOutOfBounds = 'Byte index out of range.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.641. RSCannotAllocateSocket

```
RSCannotAllocateSocket = 'Cannot allocate socket.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.642. RSCannotChangeDebugTargetAtWhileActive

```
RSCannotChangeDebugTargetAtWhileActive = 'Cannot change target while active.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.643. RSCMDNotRecognized

```
RSCMDNotRecognized = 'command not recognized';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.644. RSCodeNoError

```
RSCodeNoError = 'RCode NO Error';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.645. RSCodeQueryFormat

```
RSCodeQueryFormat = 'DNS Server Reports Query Format Error';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.646. RSCodeQueryName

```
RSCodeQueryName = 'DNS Server Reports Query Name Error';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.647. RSCodeQueryNotImplemented

```
RSCodeQueryNotImplemented = 'DNS Server Reports Query Not Implemented Error';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.648. RSCodeQueryQueryRefused

```
RSCodeQueryQueryRefused = 'DNS Server Reports Query Refused Error';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.649. RSCodeQueryServer

```
RSCodeQueryServer = 'DNS Server Reports Query Server Error';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.650. RSCodeQueryUnknownError

```
RSCodeQueryUnknownError = 'Server Returned Unknown Error';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.651. RSCoderNoTableEntryNotFound

```
RSCoderNoTableEntryNotFound = 'Coding table entry not found.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.652. RSConnectionClosedGracefully

```
RSConnectionClosedGracefully = 'Connection Closed Gracefully.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.653. RSCorruptServicesFile

```
RSCorruptServicesFile = '%s is corrupt.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.654. RSCouldNotBindSocket

```
RSCouldNotBindSocket = 'Could not bind socket. Address and port are already in use.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.655. RSCouldNotLoad

```
RSCouldNotLoad = '%s could not be loaded.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.656. RSDestinationFileAlreadyExists

```
RSDestinationFileAlreadyExists = 'Destination file already exists.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.657. RSDNSMailAObsolete

```
RSDNSMailAObsolete = 'MailA is an Obsolete Command. USE MX.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.658. RSDNSMailBNotImplemented

```
RSDNSMailBNotImplemented = '-Err 501 MailB is not implemented';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.659. RSDNSMDISObsolete

```
RSDNSMDISObsolete = 'MD is an Obsolete Command. Use MX.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.660. RSDNSMFIsObsolete

```
RSDNSMFIsObsolete = 'MF is an Obsolete Command. USE MX.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.661. RSFailedTimeZoneInfo

```
RSFailedTimeZoneInfo = 'Failed attempting to retrieve time zone information.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.662. RSFTPUnknownHost

```
RSFTPUnknownHost = 'Unknown';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.663. RSGopherNotGopherPlus

```
RSGopherNotGopherPlus = '%s is not a Gopher+ server';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.664. RSGopherServerNoProgramCode

```
RSGopherServerNoProgramCode = 'Error: No program code to return request!';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.665. RSHTTPAccepted

```
RSHTTPAccepted = 'Accepted';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.666. RSHTTPBadGateway

```
RSHTTPBadGateway = 'Bad Gateway';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.667. RSHTTPBadRequest

```
RSHTTPBadRequest = 'Bad Request';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.668. RSHTTPCannotSwitchSessionStateWhenActive

```
RSHTTPCannotSwitchSessionStateWhenActive = 'Cannot change session state when the server is active.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.669. RSHTTPChunkStarted

```
RSHTTPChunkStarted = 'Chunk Started';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.670. RSHTTPConflict

```
RSHTTPConflict = 'Conflict';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.671. RSHTTPContinue

```
RSHTTPContinue = 'Continue';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.672. RSHTTPCreated

```
RSHTTPCreated = 'Created';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.673. RSHTTPErrorParsingCommand

```
RSHTTPErrorParsingCommand = 'Error in parsing command.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.674. RSHTTFForbidden

RSHTTFForbidden = 'Forbidden';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.675. RSHTTPGatewayTimeout

RSHTTPGatewayTimeout = 'Gateway timeout';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.676. RSHTTPGone

RSHTTPGone = 'Gone';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.677. RSHTTPHeaderAlreadyWritten

RSHTTPHeaderAlreadyWritten = 'Header has already been written.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.678. RSHTTPHTTPVersionNotSupported

RSHTTPHTTPVersionNotSupported = 'HTTP version not supported';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.679. RSHTTPInternalServerError

RSHTTPInternalServerError = 'Internal Server Error';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.680. RSHTTPLengthRequired

RSHTTPLengthRequired = 'Length Required';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.681. RSHTTPMethodNotAllowed

RSHTTPMethodNotAllowed = 'Method not allowed';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.682. RSHTTPMovedPermanently

RSHTTPMovedPermanently = 'Moved Permanently';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.683. RSHTTPMovedTemporarily

RSHTTPMovedTemporarily = 'Moved Temporarily';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.684. RSHTTPNoContent

RSHTTPNoContent = 'No Content';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.685. RSHTTPNonAuthoritativeInformation

RSHTTPNonAuthoritativeInformation = 'Non-authoritative Information';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.686. RSHTTPNotAcceptable

RSHTTPNotAcceptable = 'Not Acceptable';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.687. RSHTTPNotFound

RSHTTPNotFound = 'Not Found';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.688. RSHTTPNotImplemented

RSHTTPNotImplemented = 'Not Implemented';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.689. RSHTTPNotModified

RSHTTPNotModified = 'Not Modified';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.690. RSHTTPOK

RSHTTPOK = 'OK';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.691. RSHTTTPartialContent

RSHTTTPartialContent = 'Partial Content';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.692. RSHTTTPreconditionFailed

RSHTTTPreconditionFailed = 'Precondition Failed';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.693. RSHTTTProxyAuthenticationRequired

RSHTTTProxyAuthenticationRequired = 'Proxy Authentication Required';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.694. RSHTTTPRequestEntityToLong

RSHTTTPRequestEntityToLong = 'Request Entity To Long';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.695. RSHTTTPRequestTimeout

RSHTTTPRequestTimeout = 'Request Timeout';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.696. RSHTTTPRequestURITooLong

RSHTTTPRequestURITooLong = 'Request-URI Too Long. 256 Chars max';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.697. RSHTTTPResetContent

RSHTTTPResetContent = 'Reset Content';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.698. RSHTTTPSeeOther

RSHTTTPSeeOther = 'See Other';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.699. RSHTTTPServiceUnavailable

RSHTTTPServiceUnavailable = 'Service Unavailable';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.700. RSHTTTPSwitchingProtocols

RSHTTTPSwitchingProtocols = 'Switching protocols';

Description



The text for this constant has been generated automatically. This means that it is not documented.

3.6.701. RSHTTPUnauthorized

```
RSHTTPUnauthorized = 'Unauthorized';
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.702. RSHTTPUnknownResponseCode

```
RSHTTPUnknownResponseCode = 'Unknown Response Code';
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.703. RSHTTPUnsupportedAuthorisationScheme

```
RSHTTPUnsupportedAuthorisationScheme = 'Unsupported authorization scheme.';
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.704. RSHTTPUnsupportedMediaType

```
RSHTTPUnsupportedMediaType = 'Unsupported Media Type';
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.705. RSHTTPUseProxy

```
RSHTTPUseProxy = 'Use Proxy';
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.706. RSICMPNonEchoResponse

```
RSICMPNonEchoResponse = 'Non-echo type response received';
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.707. RSICMPNotEnoughBytes

```
RSICMPNotEnoughBytes = 'Not enough bytes received';
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.708. RSICMPReceiveError0

```
RSICMPReceiveError0 = 'ICMP Receive Error = 0.';
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.709. RSICMPWrongDestination

```
RSICMPWrongDestination = 'Received someone else's packet';
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.710. RSIdNoDataToRead

RSIdNoDataToRead = 'No data to read.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.711. RSInterceptPropInvalid

RSInterceptPropInvalid = 'Intercept value is not valid';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.712. RSInterceptProplsNil

RSInterceptPropIsNil = 'InterceptEnabled cannot be set to true when Intercept is nil.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.713. RSInvalidServiceName

RSInvalidServiceName = '%s is not a valid service.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.714. RSLPDAbortJob

RSLPDAbortJob = 'abort job';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.715. RSLPDClosingConnection

RSLPDClosingConnection = 'closing connection';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.716. RSLPDConnectTo

RSLPDConnectTo = 'connected with %s';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.717. RSLPDControlFileSaved

RSLPDControlFileSaved = 'Control file save to %s';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.718. RSLPDDataFileSaved

RSLPDDataFileSaved = 'Data file saved to %s';

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.719. RSLPDDirectoryDoesNotExist

```
RSLPDDirectoryDoesNotExist = 'Directory %s does not exist';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.720. RSLPDNoQueuesDefined

```
RSLPDNoQueuesDefined = 'Error: no queues defined';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.721. RSLPDQueueStatus

```
RSLPDQueueStatus = 'Queue %s status: %s';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.722. RSLPDReceiveControlFile

```
RSLPDReceiveControlFile = 'Receive control file';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.723. RSLPDReceiveDataFile

```
RSLPDReceiveDataFile = 'Receive data file';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.724. RSLPDServerActive

```
RSLPDServerActive = 'Server status: active';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.725. RSLPDServerStartTitle

```
RSLPDServerStartTitle = 'Winshoes LPD Server %s ';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.726. RSLPDUnknownQueue

```
RSLPDUnknownQueue = 'Unknown queue %s';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.727. RSMsgClientEncodingAttachment

```
RSMsgClientEncodingAttachment = 'Encoding attachment';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.728. RMsgClientEncodingText

RMsgClientEncodingText = 'Encoding text';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.729. RMsgCmpEdtrBodyText

RMsgCmpEdtrBodyText = 'Body Text Editor';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.730. RMsgCmpEdtrExtraHead

RMsgCmpEdtrExtraHead = 'Extra Headers Text Editor';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.731. RMsgCmpEdtrNew

RMsgCmpEdtrNew = '&New Message Part...';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.732. RNETCALCInvalidNetworkMask

RNETCALCInvalidNetworkMask = 'Invalid network mask.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.733. RNETCALCInvalidValueLength

RNETCALCInvalidValueLength = 'Invalid value length: Should be 32.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.734. RNETCALConfirmLongIPList

RNETCALConfirmLongIPList = 'There is too many IP addresses in the specified range (%d) to be displayed at design time.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.735. RNETCALInvalidIPString

RNETCALInvalidIPString = 'The string %s does not translate into a valid IP.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.736. RSNNTPCConnectionRefused

```
RSNNTPCConnectionRefused = 'Connection explicitly refused by NNTP
server.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.737. RSNNTPNoOnNewGroupsList

```
RSNNTPNoOnNewGroupsList = 'No OnNewGroupsList event has been
defined.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.738. RSNNTPNoOnNewNewsList

```
RSNNTPNoOnNewNewsList = 'No OnNewNewsList event has been defined.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.739. RSNNTPNoOnNewsgroupList

```
RSNNTPNoOnNewsgroupList = 'No OnNewsgroupList event has been
defined.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.740. RSNNTPServerGoodBye

```
RSNNTPServerGoodBye = 'Goodbye';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.741. RSNNTPServerNotRecognized

```
RSNNTPServerNotRecognized = 'command not recognized (%s)';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.742. RSNNTPStringListNotInitialized

```
RSNNTPStringListNotInitialized = 'Stringlist not initialized!';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.743. RSNoBindingsSpecified

```
RSNoBindingsSpecified = 'No bindings specified.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.744. RSNoExecuteSpecified

```
RSNoExecuteSpecified = 'No execute handler found.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.745. RSNotAllBytesSent

RSNotAllBytesSent = 'Not all bytes sent.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.746. RSNotEnoughDataInBuffer

RSNotEnoughDataInBuffer = 'Not enough data in buffer.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.747. RSObjectTypeNotSupported

RSObjectTypeNotSupported = 'Object type not supported.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.748. RSONExecuteNotAssigned

RSONExecuteNotAssigned = 'OnExecute not assigned.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.749. RSOnlyOneAntiFreeze

RSOnlyOneAntiFreeze = 'Only one TIdAntiFreeze can exist per application.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.750. RSOSSLCertificateLookup

RSOSSLCertificateLookup = 'SSL certificate request error.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.751. RSOSSLConnectionDropped

RSOSSLConnectionDropped = 'SSL connection has dropped.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.752. RSOSSLCouldNotLoadSSLLibrary

RSOSSLCouldNotLoadSSLLibrary = 'Could not load SSL library.';

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.753. RSOSslInternal

```
RSOSslInternal = 'SSL library internal error.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.754. RSOSslModeNotSet

```
RSOSslModeNotSet = 'Mode has not been set.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.755. RSOSslStatusString

```
RSOSslStatusString = 'SSL status: "%s"';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.756. RSPackageSizeTooBig

```
RSPackageSizeTooBig = 'Package Size Too Big.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.757. RSPOP3FieldNotSpecified

```
RSPOP3FieldNotSpecified = ' not specified';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.758. RSQueryInvalidHeaderID

```
RSQueryInvalidHeaderID = 'Invalid Header Id %d';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.759. RSQueryInvalidPacketSize

```
RSQueryInvalidPacketSize = 'Invalid Packet Size %d';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.760. RSQueryInvalidQueryCount

```
RSQueryInvalidQueryCount = 'Invaild Query Count %d';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.761. RSQueryLessThanFour

```
RSQueryLessThanFour = 'Received Packet is too small. Less than 4 bytes %d';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.762. RSQueryLessThanTwelve

```
RSQueryLessThanTwelve = 'Received Packet is too small. Less than 12
bytes %d';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.763. RSQueryPackReceivedTooSmall

```
RSQueryPackReceivedTooSmall = 'Received Packet is too small. %d';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.764. RSRawReceiveError0

```
RSRawReceiveError0 = 'Raw Receive Error = 0.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.765. RSSetSizeExceeded

```
RSSetSizeExceeded = 'Set Size Exceeded.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.766. RSSocksAuthError

```
RSSocksAuthError = 'Authentication error to socks server.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.767. RSSocksAuthMethodError

```
RSSocksAuthMethodError = 'Invalid socks authentication method.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.768. RSSocksRequestFailed

```
RSSocksRequestFailed = 'Request rejected or failed.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.769. RSSocksRequestIdentFailed

```
RSSocksRequestIdentFailed = 'Request rejected because the client
program and identd report different user-ids.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.



### 3.6.770. RSSocksRequestServerFailed

```
RSSocksRequestServerFailed = 'Request rejected because SOCKS server cannot connect.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.771. RSSocksServerAddressError

```
RSSocksServerAddressError = 'Address type not supported.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.772. RSSocksServerCommandError

```
RSSocksServerCommandError = 'Command not supported.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.773. RSSocksServerConnectionRefusedError

```
RSSocksServerConnectionRefusedError = 'Connection refused.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.774. RSSocksServerGeneralError

```
RSSocksServerGeneralError = 'General SOCKS server failure.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.775. RSSocksServerHostUnreachableError

```
RSSocksServerHostUnreachableError = 'Host unreachable.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.776. RSSocksServerNetUnreachableError

```
RSSocksServerNetUnreachableError = 'Network unreachable.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.777. RSSocksServerPermissionError

```
RSSocksServerPermissionError = 'Connection not allowed by ruleset.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.778. RSSocksServerRespondError

```
RSSocksServerRespondError = 'Socks server did not respond.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.779. RSSocksServerTTLExpiredError

`RSSocksServerTTLExpiredError = 'TTL expired.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.780. RSSocksUnknownError

`RSSocksUnknownError = 'Unknown socks error.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.781. RSSSLAcceptError

`RSSSLAcceptError = 'Error accepting connection with SSL.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.782. RSSSLConnectError

`RSSSLConnectError = 'Error connecting with SSL.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.783. RSSSLCreatingContextError

`RSSSLCreatingContextError = 'Error creating SSL context.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.784. RSSSLDataBindingError

`RSSSLDataBindingError = 'Error binding data to SSL socket.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.785. RSSSLGetMethodError

`RSSSLGetMethodError = 'Error geting SSL method.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.786. RSSSSLLoadingCertError

`RSSSSLLoadingCertError = 'Could not load certificate.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.787. RSSSSLLoadingKeyError

`RSSSSLLoadingKeyError = 'Could not load key, check password.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.788. RSSSLLoadingRootCertError

`RSSSLLoadingRootCertError = 'Could not load root certificate.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.789. RSSSLSettingChiperError

`RSSSLSettingChiperError = 'SetCipher failed.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.790. RSStackEACCES

`RSStackEACCES = 'Access denied.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.791. RSStackEADDRINUSE

`RSStackEADDRINUSE = 'Address already in use.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.792. RSStackEADDRNOTAVAIL

`RSStackEADDRNOTAVAIL = 'Cannot assign requested address.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.793. RSStackEAFNOSUPPORT

`RSStackEAFNOSUPPORT = 'Address family not supported by protocol family.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.794. RSStackEALREADY

`RSStackEALREADY = 'Operation already in progress.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.795. RSStackEBADF

`RSStackEBADF = 'Bad file number.';`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.796. RSStackECONNABORTED

```
RSStackECONNABORTED = 'Software caused connection abort.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.797. RSStackECONNREFUSED

```
RSStackECONNREFUSED = 'Connection refused.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.798. RSStackECONNRESET

```
RSStackECONNRESET = 'Connection reset by peer.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.799. RSStackEDESTADDRREQ

```
RSStackEDESTADDRREQ = 'Destination address required.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.800. RSStackEDQUOT

```
RSStackEDQUOT = 'Disk Quota Exceeded.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.801. RSStackEFAULT

```
RSStackEFAULT = 'Bad address.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.802. RSStackEHOSTDOWN

```
RSStackEHOSTDOWN = 'Host is down.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.803. RSStackEHOSTUNREACH

```
RSStackEHOSTUNREACH = 'No route to host.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.804. RSStackEINPROGRESS

```
RSStackEINPROGRESS = 'Operation now in progress.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.805. RSStackEINTR

```
RSStackEINTR = 'Interrupted system call.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.806. RSStackEINVAL

```
RSStackEINVAL = 'Invalid argument.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.807. RSStackEISCONN

```
RSStackEISCONN = 'Socket is already connected.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.808. RSStackELOOP

```
RSStackELOOP = 'Too many levels of symbolic links.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.809. RSStackEMFILE

```
RSStackEMFILE = 'Too many open files.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.810. RSStackEMSGSIZE

```
RSStackEMSGSIZE = 'Message too long.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.811. RSStackENAMETOOLONG

```
RSStackENAMETOOLONG = 'File name too long.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.812. RSStackENETDOWN

```
RSStackENETDOWN = 'Network is down.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.813. RSStackENETRESET

```
RSStackENETRESET = 'Net dropped connection or reset.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.814. RSStackENETUNREACH

```
RSStackENETUNREACH = 'Network is unreachable.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.815. RSStackENOBUFS

```
RSStackENOBUFS = 'No buffer space available.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.816. RSStackENOPROTOOPT

```
RSStackENOPROTOOPT = 'Bad protocol option.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.817. RSStackENOTCONN

```
RSStackENOTCONN = 'Socket is not connected.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.818. RSStackENOTEMPTY

```
RSStackENOTEMPTY = 'Directory not empty';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.819. RSStackENOTSOCK

```
RSStackENOTSOCK = 'Socket operation on non-socket.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.820. RSStackEOPNOTSUPP

```
RSStackEOPNOTSUPP = 'Operation not supported on socket.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.821. RSStackEPFNOSUPPORT

```
RSStackEPFNOSUPPORT = 'Protocol family not supported.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.822. RSStackEPROCLIM

```
RSStackEPROCLIM = 'Too many processes.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.823. RSStackEPROTONOSUPPORT

```
RSStackEPROTONOSUPPORT = 'Protocol not supported.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.824. RSStackEPROTOTYPE

```
RSStackEPROTOTYPE = 'Protocol wrong type for socket.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.825. RSStackEREMOTE

```
RSStackEREMOTE = 'Too many levels of remote in path.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.826. RSStackError

```
RSStackError = 'Socket Error # %d' + #13#10 + '%s';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.827. RSStackESHUTDOWN

```
RSStackESHUTDOWN = 'Cannot send or receive after socket is closed.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.828. RSStackESOCKTNOSUPPORT

```
RSStackESOCKTNOSUPPORT = 'Socket type not supported.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.829. RSStackESTALE

```
RSStackESTALE = 'Stale NFS file handle.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.830. RSStackETIMEDOUT

```
RSStackETIMEDOUT = 'Connection timed out.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.831. RSStackETOOMANYREFS

```
RSStackETOOMANYREFS = 'Too many references, cannot splice.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.832. RSStackEUSERS

```
RSStackEUSERS = 'Too many users.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.833. RSStackEWOULDBLOCK

```
RSStackEWOULDBLOCK = 'Operation would block. ';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.834. RSStackHOST\_NOT\_FOUND

```
RSStackHOST_NOT_FOUND = 'Host not found.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.835. RSStackNO\_DATA

```
RSStackNO_DATA = 'Valid name, no data record (check DNS setup).';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.836. RSStackNO\_RECOVERY

```
RSStackNO_RECOVERY = 'Non-recoverable errors: FORMERR, REFUSED, NOTIMP.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.837. RSStackNOTINITIALISED

```
RSStackNOTINITIALISED = 'Winsock not loaded yet.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.838. RSStackSYSNOTREADY

```
RSStackSYSNOTREADY = 'Network subsystem is unavailable.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.839. RSStackTRY\_AGAIN

```
RSStackTRY_AGAIN = 'Non-authoritative response (try again or check DNS setup).';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.840. RSStackVERNOTSUPPORTED

```
RSStackVERNOTSUPPORTED = 'WINSOCK DLL Version out of range.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.



### 3.6.841. RSStatusConnected

```
RSStatusConnected = 'Connected.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.842. RSStatusConnecting

```
RSStatusConnecting = 'Connecting to %s.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.843. RSStatusDisconnected

```
RSStatusDisconnected = 'Not connected.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.844. RSStatusDisconnecting

```
RSStatusDisconnecting = 'Disconnecting from %s.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.845. RSStatusResolving

```
RSStatusResolving = 'Resolving hostname %s.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.846. RSStatusText

```
RSStatusText = '%s';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.847. RSTELNETCLIConnectError

```
RSTELNETCLIConnectError = 'server not responding';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.848. RSTELNETCLIReadError

```
RSTELNETCLIReadError = 'Server did not respond.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.849. RSTELNETSRVInvalidLogin

```
RSTELNETSRVInvalidLogin = 'Invalid Login.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.850. RSTELNETSRVMaxloginAttempt

```
RSTELNETSRVMaxloginAttempt = 'Allowed login attempts exceeded, good
bye.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.851. RSTELNETSRVNoAuthHandler

```
RSTELNETSRVNoAuthHandler = 'No authentication handler has been
specified.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.852. RSTELNETSRVOnDataAvailableIsNil

```
RSTELNETSRVOnDataAvailableIsNil = 'OnDataAvailable event is nil.';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.853. RSTELNETSRVPasswordPrompt

```
RSTELNETSRVPasswordPrompt = 'Password: ';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.854. RSTELNETSRVUsernamePrompt

```
RSTELNETSRVUsernamePrompt = 'Username: ';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.855. RSTELNETSRVWelcomeString

```
RSTELNETSRVWelcomeString = 'Indy Telnet Server';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.856. RSTFTPAccessDenied

```
RSTFTPAccessDenied = 'Access to %s denied';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.857. RSTFTPDiskFull

```
RSTFTPDiskFull = 'Unable to complete write request, progress halted at
%d bytes';
```

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.858. RSTFTPFileNotFound

```
RSTFTPFileNotFound = 'Unable to open %s';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.859. RSTFTPUnexpectedOp

```
RSTFTPUnexpectedOp = 'Unexpected operation from %s:%d';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.860. RSTFTPUnsupportedTrxMode

```
RSTFTPUnsupportedTrxMode = 'Unsupported transfer mode: "%s"';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.861. RSThreadClassNotSpecified

```
RSThreadClassNotSpecified = 'Thread Class Not Specified.';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.862. RSTIdMessagePartCreate

```
RSTIdMessagePartCreate = 'TidMessagePart can not be created. Use descendant classes. ';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.863. RSTIdTextInvalidCount

```
RSTIdTextInvalidCount = 'Invalid Text count. TidText must be greater than 1';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.864. RSTimeOut

```
RSTimeOut = 'Timeout';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.865. RSTunnelConnectMsg

```
RSTunnelConnectMsg = 'Connecting';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.866. RSTunnelConnectToMasterFailed

```
RSTunnelConnectToMasterFailed = 'Cannt connect to the Master server';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.867. RSTunnelCRCFailed

```
RSTunnelCRCFailed = 'CRC Failed';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.868. RSTunnelDisconnectMsg

```
RSTunnelDisconnectMsg = 'Disconnect';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.869. RSTunnelDontAllowConnections

```
RSTunnelDontAllowConnections = 'Do not allow connctions now';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.870. RSTunnelGetByteRange

```
RSTunnelGetByteRange = 'Call to %s.GetByte [property Bytes] with index  
<> [0..%d]';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.871. RSTunnelMessageCustomInterpretError

```
RSTunnelMessageCustomInterpretError = 'Custom message interpretation  
failed';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.872. RSTunnelMessageHandlingError

```
RSTunnelMessageHandlingError = 'Message handling failed';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.873. RSTunnelMessageInterpretError

```
RSTunnelMessageInterpretError = 'Interpretation of message failed';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.874. RSTunnelMessageTypeError

```
RSTunnelMessageTypeError = 'Message type recognition error';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.875. RSTunnelTransformError

RSTunnelTransformError = 'Transform failed';

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.876. RSTunnelTransformErrorBS

RSTunnelTransformErrorBS = 'Error in transformation before send';

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.877. RSUDPReceiveError0

RSUDPReceiveError0 = 'UDP Receive Error = 0.';

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.878. RSWinsockInitializationError

RSWinsockInitializationError = 'Winsock Initialization Error.';

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.879. RSWSockStack

RSWSockStack = 'Winsock stack';

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.880. sBlockSize

Value used in TFTP Blocksize Option negotiations.

sBlockSize = 'blksize'#0;

**Description**

sBlockSize is a constant String value used in TFTP Blocksize Option negotiations.

### 3.6.881. sj1\_tbl

ISO-2022-JP character sets encodings.

sj1\_tbl: **array**[#128..#255] **of** Byte = (  
\$00,\$21,\$23,\$25,\$27,\$29,\$2B,\$2D,\$2F,\$31,\$33,\$35,\$37,\$39,\$3B,\$3D,  
\$3F,\$41,\$43,\$45,\$47,\$49,\$4B,\$4D,\$4F,\$51,\$53,\$55,\$57,\$59,\$5B,\$5D,  
\$00,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,  
\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,  
\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,  
\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,\$01,  
\$5F,\$61,\$63,\$65,\$67,\$69,\$6B,\$6D,\$6F,\$71,\$73,\$75,\$77,\$79,\$7B,\$7D,  
\$02,\$02,\$02,\$02,\$02,\$02,\$02,\$02,\$02,\$02,\$02,\$02,\$02,\$02,\$00,\$00,\$00);

**Description**

sj1\_tbl is a Byte array that represent trhe single characters #128 through #255 used in ISO-2022-JP character sets encodings.

### 3.6.882. sj2\_tbl

Double-byte alphabet used in ISO-2022-JP character encoding

sj2\_tbl: **array**[Char] **of** Word = (  
\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,

\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,  
\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,  
\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,  
\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,  
\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,  
\$0000,\$0000,\$0000,\$0000,\$0021,\$0022,\$0023,\$0024,\$0025,\$0026,  
\$0027,\$0028,\$0029,\$002A,\$002B,\$002C,\$002D,\$002E,\$002F,\$0030,  
\$0031,\$0032,\$0033,\$0034,\$0035,\$0036,\$0037,\$0038,\$0039,\$003A,  
\$003B,\$003C,\$003D,\$003E,\$003F,\$0040,\$0041,\$0042,\$0043,\$0044,  
\$0045,\$0046,\$0047,\$0048,\$0049,\$004A,\$004B,\$004C,\$004D,\$004E,  
\$004F,\$0050,\$0051,\$0052,\$0053,\$0054,\$0055,\$0056,\$0057,\$0058,  
\$0059,\$005A,\$005B,\$005C,\$005D,\$005E,\$005F,\$0000,\$0060,\$0061,  
\$0062,\$0063,\$0064,\$0065,\$0066,\$0067,\$0068,\$0069,\$006A,\$006B,  
\$006C,\$006D,\$006E,\$006F,\$0070,\$0071,\$0072,\$0073,\$0074,\$0075,  
\$0076,\$0077,\$0078,\$0079,\$007A,\$007B,\$007C,\$007D,\$007E,\$0121,  
\$0122,\$0123,\$0124,\$0125,\$0126,\$0127,\$0128,\$0129,\$012A,\$012B,  
\$012C,\$012D,\$012E,\$012F,\$0130,\$0131,\$0132,\$0133,\$0134,\$0135,  
\$0136,\$0137,\$0138,\$0139,\$013A,\$013B,\$013C,\$013D,\$013E,\$013F,  
\$0140,\$0141,\$0142,\$0143,\$0144,\$0145,\$0146,\$0147,\$0148,\$0149,  
\$014A,\$014B,\$014C,\$014D,\$014E,\$014F,\$0150,\$0151,\$0152,\$0153,  
\$0154,\$0155,\$0156,\$0157,\$0158,\$0159,\$015A,\$015B,\$015C,\$015D,  
\$015E,\$015F,\$0160,\$0161,\$0162,\$0163,\$0164,\$0165,\$0166,\$0167,  
\$0168,\$0169,\$016A,\$016B,\$016C,\$016D,\$016E,\$016F,\$0170,\$0171,  
\$0172,\$0173,\$0174,\$0175,\$0176,\$0177,\$0178,\$0179,\$017A,\$017B,  
\$017C,\$017D,\$017E,\$0000,\$0000,\$0000);

**Description**  
sj2\_tbl is an Array or Word that provides Char-indexed access to the double-byte alphabet used in ISO-2022-JP character encoding.

3.6.883. SO\_ACCEPTCONN

SO\_ACCEPTCONN = \$0002;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.884. SO\_BROADCAST

SO\_BROADCAST = \$0020;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.885. SO\_CONNDATA

SO\_CONNDATA = \$7000;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.886. SO\_CONNDATALEN

SO\_CONNDATALEN = \$7004;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.887. SO\_CONNECT\_TIME

SO\_CONNECT\_TIME = \$700C;

**Description**  
The text for this constant has been generated automatically. This means that it is not documented.

3.6.888. SO\_CONNOPT

SO\_CONNOPT = \$7001;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.889. SO\_CONNOPTLEN

SO\_CONNOPTLEN = \$7005;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.890. SO\_DEBUG

SO\_DEBUG = \$0001;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.891. SO\_DISCDATA

SO\_DISCDATA = \$7002;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.892. SO\_DISCDATALEN

SO\_DISCDATALEN = \$7006;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.893. SO\_DISCOPT

SO\_DISCOPT = \$7003;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.894. SO\_DISCOPTLEN

SO\_DISCOPTLEN = \$7007;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.895. SO\_DONTLINGER

SO\_DONTLINGER = \$ff7f;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.896. SO\_DONTROUTE

SO\_DONTROUTE = \$0010;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.897. SO\_ERROR

SO\_ERROR = \$1007;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.898. SO\_KEEPLIVE

SO\_KEEPLIVE = \$0008;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.899. SO\_LINGER

SO\_LINGER = \$0080;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.900. SO\_MAXDG

SO\_MAXDG = \$7009;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.901. SO\_MAXPATHDG

SO\_MAXPATHDG = \$700A;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.902. SO\_OOINLINE

SO\_OOINLINE = \$0100;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.903. SO\_OPENTYPE

SO\_OPENTYPE = \$7008;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.904. SO\_RCVBUF

SO\_RCVBUF = \$1002;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.905. SO\_RCVLOWAT

SO\_RCVLOWAT = \$1004;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.906. SO\_RCVTIMEO

SO\_RCVTIMEO = \$1006;

Description



The text for this constant has been generated automatically. This means that it is not documented.

3.6.907. SO\_REUSEADDR

SO\_REUSEADDR = \$0004;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.908. SO\_SNDBUF

SO\_SNDBUF = \$1001;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.909. SO\_SNDLOWAT

SO\_SNDLOWAT = \$1003;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.910. SO\_SNDTIMEO

SO\_SNDTIMEO = \$1005;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.911. SO\_SYNCHRONOUS\_ALERT

SO\_SYNCHRONOUS\_ALERT = \$10;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.912. SO\_SYNCHRONOUS\_NONALERT

SO\_SYNCHRONOUS\_NONALERT = \$20;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.913. SO\_TYPE

SO\_TYPE = \$1008;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.914. SO\_UPDATE\_ACCEPT\_CONTEXT

SO\_UPDATE\_ACCEPT\_CONTEXT = \$700B;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.915. SO\_USELOOPBACK

SO\_USELOOPBACK = \$0040;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.916. SOCK\_DGRAM

SOCK\_DGRAM = 2;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.917. SOCK\_RAW

SOCK\_RAW = 3;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.918. SOCK\_RDM

SOCK\_RDM = 4;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.919. SOCK\_SEQPACKET

SOCK\_SEQPACKET = 5;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.920. SOCK\_STREAM

SOCK\_STREAM = 1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.921. SOCKET\_ERROR

SOCKET\_ERROR = -1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.922. SOL\_SOCKET

SOL\_SOCKET = \$ffff;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.923. SOMAXCONN

SOMAXCONN = 5;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.924. TAB

Represents the Tab character.

TAB = #9;

**Description**

TAB is a constant value that represents the Tab character.

3.6.925. TCP\_BSDURGENT

TCP\_BSDURGENT = \$7000;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.926. TCP\_NODELAY

TCP\_NODELAY = \$0001;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.927. TF\_DISCONNECT

TF\_DISCONNECT = \$01;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.928. TF\_REUSE\_SOCKET

TF\_REUSE\_SOCKET = \$02;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.929. TF\_WRITE\_BEHIND

TF\_WRITE\_BEHIND = \$04;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.930. TFTP\_ACK

TrivialFTP Acknowledgment Op Code.

TFTP\_ACK = 4;

**Description**

TFTP\_ACK represents the TrivialFTP Acknowledgment Op Code.

3.6.931. TFTP\_DATA

TrivialFTP Data Op Code.

TFTP\_DATA = 3;

**Description**

TFTP\_DATA represents the TrivialFTP Data Op Code.

3.6.932. TFTP\_ERROR

TrivialFTP Error Op Code.

TFTP\_ERROR = 5;

**Description**

TFTP\_ERROR represents the TrivialFTP Error Op Code.

### 3.6.933. TFTP\_OACK

TrivialFTP Option Acknowledgement Op Code.

TFTP\_OACK = 6;

**Description**

TFTP\_OACK represents the TrivialFTP Option Acknowledgement Op Code.

### 3.6.934. TFTP\_RRQ

TrivialFTP Read Request Op Code.

TFTP\_RRQ = 1;

**Description**

TFTP\_RRQ represents the TrivialFTP Read Request Op Code.

### 3.6.935. TFTP\_WRQ

TrivialFTP Write Request Op Code.

TFTP\_WRQ = 2;

**Description**

TFTP\_WRQ represents the TrivialFTP Write Request Op Code.

### 3.6.936. tmConnect

message type for a Tunnel connection request.

tmConnect = 3;

**Description**

tmConnect is a constant value that represents the message type for a Tunnel connection request.

### 3.6.937. tmCustom

Message type for user-defined Tunnel message.

tmCustom = 99;

**Description**

tmCustom is a constant value that represents the message type for user-defined Tunnel message.

### 3.6.938. tmData

Message type for a Tunnel message containing data.

tmData = 1;

**Description**

tmData is a constant value that represents the message type for a Tunnel message containing data.

### 3.6.939. tmDisconnect

Message type for a Tunnel disconnection request.

tmDisconnect = 2;

**Description**

tmDisconnect is a constant value that represents the message type for a Tunnel disconnection request.

### 3.6.940. tmError

Message type for Tunnel error conditions.

tmError = 0;

**Description**

tmError is a constant value that represents the message type for a Tunnel where data transformation has failed, or the tunnel has been disconnected by the server.

3.6.941. TNC\_AO

Represents the TELNET Abort Output command.

TNC\_AO = #245;

**Description**

TNC\_AO is a constant value that represents the TELNET Abort Output command (Decimal 245).

3.6.942. TNC\_AYT

Represents the TELNET command Are You There.

TNC\_AYT = #246;

**Description**

TNC\_AYT is a constant value that represents the TELNET command AYT (Are You There) (Decimal 246).

3.6.943. TNC\_BREAK

Represents the TELNET Break command.

TNC\_BREAK = #243;

**Description**

TNC\_BREAK is a constant value that represents the TELNET BRK (Break) command (Decimal 243).

3.6.944. TNC\_DATA\_MARK

Represents the TELNET Data Mark in a Synch stream.

TNC\_DATA\_MARK = #242;

**Description**

TNC\_DATA\_MARK is a constant value that represents the TELNET Data Mark (DM) command (Decimal 242) in a Synch stream.

3.6.945. TNC\_DO

Represents the TELNET command to perform or confirm an option.

TNC\_DO = #253;

**Description**

TNC\_DO is a constant value that represents the TELNET command to perform or confirm an option on the peer connection (Decimal 253).

3.6.946. TNC\_DONT

Represents the TELNET command to stop or confirm stopping an option.

TNC\_DONT = #254;

**Description**

TNC\_DONT is a constant value that represents the TELNET command to stop or confirm stopping an option (Decimal 254).

3.6.947. TNC\_EC

Represents the TELNET Erase Character command.

TNC\_EC = #247;

**Description**

TNC\_EC is a constant value that represents the TELNET Erase Character command (Decimal 247).

**3.6.948. TNC\_EL**

Represents the TELNET Erase Line command.

TNC\_EL = #248;

**Description**

TNC\_EL is a constant value that represents the TELNET Erase Line command (Decimal 248).

**3.6.949. TNC\_EOR**

Represents the TELNET End-of-Record command.

TNC\_EOR = #239;

**Description**

TNC\_EOR is a constant value that represents the TELNET End-of-Record command (Decimal 239).

**3.6.950. TNC\_GA**

Represents the TELNET Go Ahead command.

TNC\_GA = #249;

**Description**

TNC\_GA is a constant value that represents the TELNET Go Ahead command (Decimal 249).

**3.6.951. TNC\_IAC**

Represents the TELNET Interpret As Command directive.

TNC\_IAC = #255;

**Description**

TNC\_IAC is a constant value that represents the TELNET Interpret As Command directive (Decimal 255).

**3.6.952. TNC\_IP**

Represents the TELNET Interrupt Process command.

TNC\_IP = #244;

**Description**

TNC\_IP is a constant value that represents the TELNET Interrupt Process command (Decimal 244).

**3.6.953. TNC\_NOP**

Represents the TELNET No Operation command.

TNC\_NOP = #241;

**Description**

TNC\_NOP is a constant value that represents the TELNET No Operation command (Decimal 241).

**3.6.954. TNC\_SB**

Represents the TELNET Subnegotiation Begin command.

TNC\_SB = #250;

**Description**

TNC\_SB is a constant value that represents the TELNET Subnegotiation Begin command (Decimal 250).

3.6.955. TNC\_SE

Represents the TELNET Subnegotiation End command.

TNC\_SE = #240;

**Description**

TNC\_SE is a constant value that represents the TELNET Subnegotiation End command (Decimal 240).

3.6.956. TNC\_WILL

Represents the TELNET command to begin or confirm an option.

TNC\_WILL = #251;

**Description**

TNC\_WILL is a constant value that represents the TELNET command to begin or confirm an option (Decimal 251).

3.6.957. TNC\_WONT

Represents the TELNET command to refuse or cease performing an option.

TNC\_WONT = #252;

**Description**

TNC\_WONT is a constant value that represents the TELNET command to refuse or cease performing an option (Decimal 252).

3.6.958. TNO\_3270REGIME

Represents the TELNET 3270 Regime option.

TNO\_3270REGIME = #29;

**Description**

TNO\_3270REGIME is a constant value that represents the TELNET 3270 Regime option (Decimal 29).

3.6.959. TNO\_AMSN

Represents the TELNET Approximate Message Size Negotiation option.

TNO\_AMSN = #4;

**Description**

TNO\_AMSN is a constant value that represents the TELNET Approximate Message Size Negotiation option (Decimal 4).

3.6.960. TNO\_AUTH

Represents the TELNET Authenticate option.

TNO\_AUTH = #37;

**Description**

TNO\_AUTH is a constant value that represents the TELNET Authenticate option (Decimal 37).

3.6.961. TNO\_BINARY

Represents the TELNET Binary Transmission option.

TNO\_BINARY = #0;

**Description**

TNO\_BINARY is a constant value that represents the TELNET Binary Transmission option (Decimal 0).

3.6.962. TNO\_BYTE\_MACRO

Represents the TELNET Byte Macro option.

TNO\_BYTE\_MACRO = #19;

Description

TNO\_BYTE\_MACRO is a constant value that represents the TELNET Byte Macro option (Decimal 19).

3.6.963. TNO\_DET

Represents the TELNET Data Entry Terminal option.

TNO\_DET = #20;

Description

TNO\_DET is a constant value that represents the TELNET Data Entry Terminal option (Decimal 20).

3.6.964. TNO\_EA

Represents the TELNET Extended ASCII option.

TNO\_EA = #17;

Description

TNO\_EA is a constant value that represents the TELNET Extended ASCII option (Decimal 17).

3.6.965. TNO\_ECHO

Represents the TELNET Echo option.

TNO\_ECHO = #1;

Description

TNO\_ECHO is a constant value that represents the TELNET Echo option (Decimal 1).

3.6.966. TNO\_ENCRYPT

Represents the TELNET Encryption option.

TNO\_ENCRYPT = #38;

Description

TNO\_ENCRYPT is a constant value that represents the TELNET Encryption option (Decimal 38).

3.6.967. TNO\_EOL

Represents the TELNET Extended-Options-List option.

TNO\_EOL = #255;

Description

TNO\_EOL is a constant value that represents the TELNET Extended-Options-List option (Decimal 255).

3.6.968. TNO\_EOR

Represents the TELNET End-of-Record option.

TNO\_EOR = #25;

Description



TNO\_EOR is a constant value that represents the TELNET End-of-Record option (Decimal 25).

3.6.969. TNO\_LINEMODE

Represents the TELNET Line Mode option.

TNO\_LINEMODE = #34;

Description

TNO\_LINEMODE is a constant value that represents the TELNET Line Mode option (Decimal 34).

3.6.970. TNO\_LOGOUT

Represents the TELNET Logout option.

TNO\_LOGOUT = #18;

Description

TNO\_LOGOUT is a constant value that represents the TELNET Logout option (Decimal 18).

3.6.971. TNO\_NAWS

Represents the TELNET NVT Approximate Window Size option.

TNO\_NAWS = #31;

Description

TNO\_NAWS is a constant value that represents the TELNET NVT Approximate Window Size option (Decimal 31).

3.6.972. TNO\_OCRD

Represents the TELNET Output Carriage-Return Disposition option.

TNO\_OCRD = #10;

Description

TNO\_OCRD is a constant value that represents the TELNET Output Carriage-Return Disposition option (Decimal 10).

3.6.973. TNO\_OFD

Represents the TELNET option Output FormFeed Disposition.

TNO\_OFD = #13;

Description

TNO\_OFD is a constant value that represents the TELNET option Output FormFeed Disposition (Decimal 13).

3.6.974. TNO\_OHTD

Represents the TELNET Output Horizontal Tab Disposition option.

TNO\_OHTD = #12;

Description

TNO\_OHTD is a constant value that represents the TELNET Output Horizontal Tab Disposition option (Decimal 12).

3.6.975. TNO\_OHTS

Represents the TELNET Output Horizontal Tab Stops option.

TNO\_OHTS = #11;

**Description**

TNO\_OHTS is a constant value that represents the TELNET Output Horizontal Tab Stops option (Decimal 11).

3.6.976. TNO\_OLD

Represents the TELNET Output Linefeed Disposition option.

TNO\_OLD = #16;

**Description**

TNO\_OLD is a constant value that represents the TELNET Output Linefeed Disposition option (Decimal 16).

3.6.977. TNO\_OLW

Represents the TELNET Output Line Width option.

TNO\_OLW = #8;

**Description**

TNO\_OLW is a constant value that represents the TELNET Output Line Width option (Decimal 8).

3.6.978. TNO\_OM

Represents the TELNET Output Marking option.

TNO\_OM = #27;

**Description**

TNO\_OM is a constant value that represents the TELNET Output Marking option (Decimal 27).

3.6.979. TNO\_OPS

Represents the TELNET Output Page Size option.

TNO\_OPS = #9;

**Description**

TNO\_OPS is a constant value that represents the TELNET Output Page Size option (Decimal 9).

3.6.980. TNO\_OVT

Represents the TELNET Output Vertical Tabstops option.

TNO\_OVT = #14;

**Description**

TNO\_OVT is a constant value that represents the TELNET Output Vertical Tabstops option (Decimal 14).

3.6.981. TNO\_OVTD

Represents the TELNET Output Vertical Tab Disposition option.

TNO\_OVTD = #15;

**Description**

TNO\_OVTD is a constant value that represents the TELNET Output Vertical Tab Disposition option (Decimal 15).

3.6.982. TNO\_RCTE

Represents the TELNET Remote Controlled Transmit and Echo option.

TNO\_RCTE = #7;

**Description**

TNO\_RCTE is a constant value that represents the TELNET Remote Controlled Transmit and Echo option (Decimal 7).

3.6.983. TNO\_RECONNECT

Represents the TELNET Reconnection option.

TNO\_RECONNECT = #2;

**Description**

TNO\_RECONNECT is a constant value that represents the TELNET Reconnection option (Decimal 2).

3.6.984. TNO\_RFLOW

Represents the TELNET Remote Flow Control option.

TNO\_RFLOW = #33;

**Description**

TNO\_RFLOW is a constant value that represents the TELNET Remote Flow Control option (Decimal 33).

3.6.985. TNO\_SGA

Represents the TELNET Suppress Go Ahead option.

TNO\_SGA = #3;

**Description**

TNO\_SGA is a constant value that represents the TELNET Suppress Go Ahead option (Decimal 3).

3.6.986. TNO\_SL

Represents the TELNET Send *Send* Location option.

TNO\_SL = #23;

**Description**

TNO\_SL is a constant value that represents the TELNET Send *Send* Location option (Decimal 23).

3.6.987. TNO\_STATUS

Represents the TELNET Status option.

TNO\_STATUS = #5;

**Description**

TNO\_STATUS is a constant value that represents the TELNET Status option (Decimal 5).

3.6.988. TNO\_SUPDUP

Represents the TELNET SUPDUP Terminal Service option.

TNO\_SUPDUP = #21;

**Description**

TNO\_SUPDUP is a constant value that represents the TELNET SUPDUP Terminal Service option (Decimal 21).

3.6.989. TNO\_SUPDUP\_OUTPUT

Represents the TELNET SUPDUP Terminal Output option.

TNO\_SUPDUP\_OUTPUT = #22;

**Description**

TNO\_SUPDUP\_OUTPUT is a constant value that represents the TELNET SUPDUP Terminal Output option. (Decimal 22).

3.6.990. TNO\_TACACS\_ID

Represents the TELNET TACACS User Identification option.

TNO\_TACACS\_ID = #26;  
**Description**  
TNO\_TACACS\_ID is a constant value that represents the TELNET TACACS User Identification option (Decimal 26).

3.6.991. TNO\_TERM\_SPEED

Represents the TELNET Terminal Speed option.

TNO\_TERM\_SPEED = #32;  
**Description**  
TNO\_TERM\_SPEED is a constant value that represents the TELNET Terminal Speed option (Decimal 32).

3.6.992. TNO\_TERMTYPE

Represents the TELNET Terminal Type option.

TNO\_TERMTYPE = #24;  
**Description**  
TNO\_TERMTYPE is a constant value that represents the TELNET Terminal Type option (Decimal 24).

3.6.993. TNO\_TIMING\_MARK

Represents the TELNET Timing Mark option.

TNO\_TIMING\_MARK = #6;  
**Description**  
TNO\_TIMING\_MARK is a constant value that represents the TELNET Timing Mark option (Decimal 6).

3.6.994. TNO\_TLN

Represents the TELNET Terminal Location Number option.

TNO\_TLN = #28;  
**Description**  
TNO\_TLN is a constant value that represents the TELNET Terminal Location Number option (Decimal 28).

3.6.995. TNO\_X3PAD

Represents the TELNET X.3 PAD option.

TNO\_X3PAD = #30;  
**Description**  
TNO\_X3PAD is a constant value that represents the TELNET X.3 PAD option (Decimal 30).

3.6.996. TNO\_XDISPLOC

Represents the TELNET X Display Location option.

TNO\_XDISPLOC = #35;  
**Description**

TNO\_XDISPLOC is a constant value that represents the TELNET X Display Location option (Decimal 35).

3.6.997. TNOS\_NAME

Represents a TELNET Suboption Name.

TNOS\_NAME = #3;

Description

TNOS\_NAME is a constant value that represents the TELNET Suboption Name (Decimal 3).

3.6.998. TNOS\_REPLY

Represents a TELNET Suboption reply.

TNOS\_REPLY = #2;

Description

TNOS\_REPLY is a constant value that represents a TELNET Suboption reply (Decimal 2).

3.6.999. TNOS\_TERM\_IS

Represents the TELNET Temrinal Identification Suboption.

TNOS\_TERM\_IS = #0;

Description

TNOS\_TERM\_IS is a constant value that represents the TELNET Temrinal Identification Suboption (Decimal 1).

3.6.1000.TNOS\_TERMTYPE\_SEND

Represents the TELNET Terminal Type Send *Send* Suboption.

TNOS\_TERMTYPE\_SEND = #1;

Description

TNOS\_TERMTYPE\_SEND is a constant value that represents the TELNET Terminal Type Send *Send* Suboption (Decimal 1).

3.6.1001TRY\_AGAIN

TRY\_AGAIN = WSATRY\_AGAIN *WSATRY\_AGAIN*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1002TZ\_ADT

TZ\_ADT = -3;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1003TZ\_AHST

TZ\_AHST = -10;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1004TZ\_AST

TZ\_AST = -4;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1005TZ\_AT

TZ\_AT = -2;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1006TZ\_BST

TZ\_BST = TZ\_WAT *↗TZ\_WAT*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1007TZ\_BT

TZ\_BT = 3;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1008TZ\_CAT

TZ\_CAT = TZ\_AHST *↗TZ\_AHST*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1009TZ\_CCT

TZ\_CCT = 8;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1010TZ\_CDT

TZ\_CDT = TZ\_EST *↗TZ\_EST*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1011TZ\_CET

TZ\_CET = 1;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1012TZ\_CST

TZ\_CST = -6;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1013TZ\_EADT

TZ\_EADT = 11;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1014TZ\_EAST

TZ\_EAST = TZ\_AHST *↗TZ\_AHST*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1015TZ\_EDT

TZ\_EDT = TZ\_AST *↗TZ\_AST*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1016TZ\_EET

TZ\_EET = 2;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1017TZ\_EST

TZ\_EST = -5;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1018TZ\_FST

TZ\_FST = TZ\_EET *↗TZ\_EET*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1019TZ\_FWT

TZ\_FWT = TZ\_CET *↗TZ\_CET*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1020TZ\_GMT

TZ\_GMT = 0;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1021TZ\_GST

TZ\_GST = 10;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1022TZ\_HDT

TZ\_HDT = TZ\_YST *↗TZ\_YST*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1023TZ\_HST

TZ\_HST = TZ\_AHST *↗TZ\_AHST*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1024TZ\_IDLE

TZ\_IDLE = 12;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1025TZ\_IDLW

TZ\_IDLW = -12;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1026TZ\_JST

TZ\_JST = 9;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1027TZ\_MDT

TZ\_MDT = TZ\_CST *↗TZ\_CST*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1028TZ\_MEST

TZ\_MEST = TZ\_EET *↗TZ\_EET*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1029TZ\_MESZ

TZ\_MESZ = TZ\_EET *↗TZ\_EET*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1030TZ\_MET

TZ\_MET = TZ\_CET *↗TZ\_CET*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1031TZ\_MEWT

TZ\_MEWT = TZ\_CET *↗TZ\_CET*;

Description

The text for this constant has been generated automatically. This means that it is not documented.



3.6.1032TZ\_MST

TZ\_MST = -7;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1033TZ\_NT

TZ\_NT = -11;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1034TZ\_NZDT

TZ\_NZDT = 13;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1035TZ\_NZST

TZ\_NZST = TZ\_IDLE *⚡TZ\_IDLE*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1036TZ\_NZT

TZ\_NZT = TZ\_IDLE *⚡TZ\_IDLE*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1037TZ\_PDT

TZ\_PDT = TZ\_MST *⚡TZ\_MST*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1038TZ\_PST

TZ\_PST = -8;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1039TZ\_SST

TZ\_SST = TZ\_EET *⚡TZ\_EET*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1040TZ\_SWT

TZ\_SWT = TZ\_CET *⚡TZ\_CET*;

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1041TZ\_UT

`TZ_UT = TZ_GMT TZ_GMT ;`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1042TZ\_UTC

`TZ_UTC = TZ_GMT TZ_GMT ;`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1043TZ\_WADT

`TZ_WADT = TZ_CCT TZ_CCT ;`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1044TZ\_WAST

`TZ_WAST = 7 ;`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1045TZ\_WAT

`TZ_WAT = -1 ;`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1046TZ\_WET

`TZ_WET = TZ_GMT TZ_GMT ;`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1047TZ\_YDT

`TZ_YDT = TZ_PST TZ_PST ;`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1048TZ\_YST

`TZ_YST = -9 ;`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1049TZ\_ZP4

`TZ_ZP4 = 4 ;`

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1050TZ\_ZP5

TZ\_ZP5 = 5 ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1051TZ\_ZP6

TZ\_ZP6 = 6 ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1052TZM\_A

TZM\_A = TZ\_WAT ↗TZ\_WAT ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1053TZM\_Alpha

TZM\_Alpha = TZM\_A ↗TZM\_A ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1054TZM\_B

TZM\_B = TZ\_AT ↗TZ\_AT ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1055TZM\_Bravo

TZM\_Bravo = TZM\_B ↗TZM\_B ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1056TZM\_C

TZM\_C = TZ\_ADT ↗TZ\_ADT ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1057TZM\_Charlie

TZM\_Charlie = TZM\_C ↗TZM\_C ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1058TZM\_D

TZM\_D = TZ\_AST ↗TZ\_AST ;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1059TZM\_Delta

```
TZM_Delta = TZM_D  ⚡TZM_D;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1060TZM\_E

```
TZM_E = TZ_EST  ⚡TZ_EST;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1061TZM\_Echo

```
TZM_Echo = TZM_E  ⚡TZM_E;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1062TZM\_F

```
TZM_F = TZ_CST  ⚡TZ_CST;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1063TZM\_Foxtrot

```
TZM_Foxtrot = TZM_F  ⚡TZM_F;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1064TZM\_G

```
TZM_G = TZ_MST  ⚡TZ_MST;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1065TZM\_Golf

```
TZM_Golf = TZM_G  ⚡TZM_G;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1066TZM\_H

```
TZM_H = TZ_PST  ⚡TZ_PST;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1067TZM\_Hotel

```
TZM_Hotel = TZM_H  ⚡TZM_H;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1068TZM\_J

TZM\_J = TZ\_YST *↗TZ\_YST;*

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1069TZM\_Juliet

TZM\_Juliet = TZM\_J *↗TZM\_J;*

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1070TZM\_K

TZM\_K = TZ\_AHST *↗TZ\_AHST;*

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1071TZM\_Kilo

TZM\_Kilo = TZM\_K *↗TZM\_K;*

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1072TZM\_L

TZM\_L = TZ\_NT *↗TZ\_NT;*

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1073TZM\_Lima

TZM\_Lima = TZM\_L *↗TZM\_L;*

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1074TZM\_M

TZM\_M = TZ\_IDLW *↗TZ\_IDLW;*

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1075TZM\_Mike

TZM\_Mike = TZM\_M *↗TZM\_M;*

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1076TZM\_N

TZM\_N = TZ\_CET *↗TZ\_CET;*

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1077TzM\_November

```
TzM_November = TzM_N ⚡TzM_N;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1078TzM\_O

```
TzM_O = Tz_EET ⚡Tz_EET;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1079TzM\_Oscar

```
TzM_Oscar = TzM_O ⚡TzM_O;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1080TzM\_P

```
TzM_P = Tz_BT ⚡Tz_BT;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1081TzM\_Papa

```
TzM_Papa = TzM_P ⚡TzM_P;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1082TzM\_Q

```
TzM_Q = Tz_ZP4 ⚡Tz_ZP4;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1083TzM\_Quebec

```
TzM_Quebec = TzM_Q ⚡TzM_Q;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1084TzM\_R

```
TzM_R = Tz_ZP5 ⚡Tz_ZP5;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1085TzM\_Romeo

```
TzM_Romeo = TzM_R ⚡TzM_R;
```

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1086TZM\_S

TZM\_S = TZ\_ZP6 *↗TZ\_ZP6;*

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1087TZM\_Sierra

TZM\_Sierra = TZM\_S *↗TZM\_S;*

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1088TZM\_T

TZM\_T = TZ\_WAST *↗TZ\_WAST;*

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1089TZM\_Tango

TZM\_Tango = TZM\_T *↗TZM\_T;*

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1090TZM\_U

TZM\_U = TZ\_CCT *↗TZ\_CCT;*

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1091TZM\_Uniform

TZM\_Uniform = TZM\_U *↗TZM\_U;*

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1092TZM\_V

TZM\_V = TZ\_JST *↗TZ\_JST;*

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1093TZM\_Victor

TZM\_Victor = TZM\_V *↗TZM\_V;*

Description

The text for this constant has been generated automatically. This means that it is not documented.

3.6.1094TZM\_W

TZM\_W = TZ\_GST *↗TZ\_GST;*

Description

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1095TzM\_Whiskey

TzM\_Whiskey = TzM\_W *↗TzM\_W*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1096TzM\_X

TzM\_X = Tz\_NT *↗Tz\_NT*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1097TzM\_XRay

TzM\_XRay = TzM\_X *↗TzM\_X*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1098TzM\_Y

TzM\_Y = Tz\_IDLE *↗Tz\_IDLE*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1099TzM\_Yankee

TzM\_Yankee = TzM\_Y *↗TzM\_Y*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1100TzM\_Z

TzM\_Z = Tz\_GMT *↗Tz\_GMT*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1101TzM\_Zulu

TzM\_Zulu = TzM\_Z *↗TzM\_Z*;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1102UUBegin

UUEncoding begin table marker.

UUBegin = 'BEGIN ';

**Description**

UUBegin is a constant value that represents the UUEncoding begin table marker.

### 3.6.1103UUBEGINFound

State value for location of the UUEncode header.

UUBEGINFound = 5;

**Description**

UUBEGINFound is a constant value that represents the TIdUUDecoder *↗TIdUUDecoder* state set when the beginning of the UUEncode header has been located.



### 3.6.1104UUCodeTable

Represents the character encoding table used for UUEncoding operations.

```
UUCodeTable: string  =  '`!"#$%&'()*+,-
./0123456789:;<=>?@ABCDEFGHIJKLMN
OPQRSTUVWXYZ[ ]^_';
```

**Description**

UUCodeTable is a String constant that represents the character encoding table used for UUEncoding operations.

### 3.6.1105UUDataStarted

Represents the coder state for encoding data operations.

```
UUDataStarted = 4;
```

**Description**

UUDataStarted is a constant value that represents the coder state message used when data encoding operations have started for the coder.

### 3.6.1106UUEnd

UUEncoding end table marker.

```
UUEnd = 'END';
```

**Description**

UUEnd ia constant value that represents the UUEncoding end table marker.

### 3.6.1107UUENDFound

State value for completion of the UUDecode operation.

```
UUENDFound = 10;
```

**Description**

UUENDFound is a constant value that represents the TldUUDecoder *↯TldUUDecoder* state value set when the end of the UUEncode data has been encountered.

### 3.6.1108UUErrIncompletePrivilege

Error message for invalid UNIX privileges in the UUEncode header.

```
UUErrIncompletePrivilege = 'Not enough chars for three-digit
Privilege';
```

**Description**

UUErrIncompletePrivilege is a constant value that represents the error message added to the TldUUDecoder *↯TldUUDecoder* error message list when the UNIX privilege in the UUEncode header does not contain the required three character UNIX user permissions.

### 3.6.1109UUErrIncompletePrivilege2

Error message for invalid UNIX privileges in the UUEncode header.

```
UUErrIncompletePrivilege2 = 'Too many chars for three-digit
Privilege';
```

**Description**

UUErrIncompletePrivilege2 is a constant value that represents the message added to the TldUUDecoder *↯TldUUDecoder* error list when the UNIX privilege in the UUEncode header contains more than the three characters allowing in a UNIX user permissions.

### 3.6.1110UUErrorDataEndWithoutEND

Error message for indicating completion before END in UUEncoded data.

```
UUErrorDataEndWithoutEND = ' Data ended without an END statment';
```

**Description**

UUErrorDataEndWithoutEND is a constant value that represents the message added to the TIdUUDecoder *⚡TIdUUDecoder* message list when the UUDecode operation has been completed prior to the END directive in the encoded packet.

### 3.6.1111UUErrorNoBEGINafterTABLE

Error message for missing BEGIN for UUEncoded data.

```
UUErrorNoBEGINafterTABLE = 'No BEGIN statement followed a TABLE';
```

**Description**

UUErrorNoBEGINafterTABLE is a constant value that represents the message added to the TIdUUDecoder *⚡TIdUUDecoder* message list when the UUDecode operation has processed the UUEncode table but did not encounter the BEGIN directive for UUEncoded data.

### 3.6.1112UUErrorPivilageNotNumeric

```
UUErrorPivilageNotNumeric = 'Privilege chars not numeric';
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1113UUErrTableNotAtEnd

Error message for an error reading the UUEncode table.

```
UUErrTableNotAtEnd = UUTable ⚡UUTable = ' not at end of line';
```

**Description**

UUErrTableNotAtEnd is a constant value that represents the error messages added to the TIdUUDecoder *⚡TIdUUDecoder* message list when an end of the UUEncode Alphabet table is expected, but not encountered.

### 3.6.1114UUInitialLength

Size of the decoder internal buffer.

```
UUInitialLength = 8;
```

**Description**

UUInitialLength is a constant value that represents the initial length of the TIdUUDecoder *⚡TIdUUDecoder* internal buffer for decoding operations.

### 3.6.1115UULastCharFound

State value for completion of encoded data.

```
UULastCharFound = 9;
```

**Description**

UULastCharFound is a constant value that represents when state value set when TIdUUDecoder *⚡TIdUUDecoder* has read the last byte of data in the UUEncoded packet.

### 3.6.1116UUPrivilegeFound

UUDecoder state after locating UNIX privilege.

```
UUPrivilegeFound = 6;
```

**Description**

UUPrivilegeFound is a constant value that represents the TIdUUDecoder *⚡TIdUUDecoder* state value set when the UNIX privilege is located in the UUEncode header.

### 3.6.1117UUStarted

UUStarted = 0;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1118UUTable

UUEncoding table marker.

UUTable = 'TABLE';

**Description**

UUTable is a constant value that represents the UUEncoding table marker.

### 3.6.1119UUTableBeenRead

State value for location of the UUEncode alphabet in the header.

UUTableBeenRead = 3;

**Description**

UUTableBeenRead is a constant value that represents the TIdUUDecoder *↗TIdUUDecoder* state set when the UUEncode alphabet table has been read from the UUEncode header.

### 3.6.1120UUTableBegun

State value for beginning of a table in the UUEncode header.

UUTableBegun = 1;

**Description**

UUTableBegun is a constant value that represents the TIdUUDecoder *↗TIdUUDecoder* state value set when the beginning of a table in the UUEncode header has been encountered.

### 3.6.1121UUTableOneLine

State value for reading a UUEncode table line.

UUTableOneLine = 2;

**Description**

UUTableOneLine is a constant value that represents the TIdUUDecoder *↗TIdUUDecoder* state value set when a single line from the UUEncode alphabet table has been read.

### 3.6.1122vkana\_tbl

Half-width Voiced Katakana characters.

vkana\_tbl: **array**[\$A1..#\$DF] **of** Word = (  
\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,  
\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$2574,\$0000,  
\$0000,\$252C,\$252E,\$2530,\$2532,\$2534,\$2536,\$2538,\$253A,\$253C,  
\$253E,\$2540,\$2542,\$2545,\$2547,\$2549,\$0000,\$0000,\$0000,\$0000,  
\$0000,\$2550,\$2553,\$2556,\$2559,\$255C,\$0000,\$0000,\$0000,\$0000,  
\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,\$0000,  
\$0000,\$0000,\$0000);

**Description**

vkana\_tbl is a Word array that represents the voiced half-width Katakana characters in the range #A1 through #DF. vkana\_tbl is used in ISO-2022-JP character set encoding.

### 3.6.1123WSABASEERR

WSABASEERR = 10000;

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1124WSAEACCES

WSAEACCES = (WSABASEERR+13);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1125WSAEADDRINUSE

WSAEADDRINUSE = (WSABASEERR+48);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1126WSAEADDRNOTAVAIL

WSAEADDRNOTAVAIL = (WSABASEERR+49);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1127WSAEAFNOSUPPORT

WSAEAFNOSUPPORT = (WSABASEERR+47);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1128WSAEALREADY

WSAEALREADY = (WSABASEERR+37);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1129WSAEBADF

WSAEBADF = (WSABASEERR+9);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1130WSAECONNABORTED

WSAECONNABORTED = (WSABASEERR+53);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1131WSAECONNREFUSED

WSAECONNREFUSED = (WSABASEERR+61);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1132WSAECONNRESET

WSAECONNRESET = (WSABASEERR+54);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1133WSAEDESTADDRREQ

WSAEDESTADDRREQ = (WSABASEERR+39);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1134WSAEDISCON

WSAEDISCON = (WSABASEERR+101);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1135WSAEDQUOT

WSAEDQUOT = (WSABASEERR+69);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1136WSAEFAULT

WSAEFAULT = (WSABASEERR+14);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1137WSAEHOSTDOWN

WSAEHOSTDOWN = (WSABASEERR+64);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1138WSAEHOSTUNREACH

WSAEHOSTUNREACH = (WSABASEERR+65);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1139WSAEINPROGRESS

WSAEINPROGRESS = (WSABASEERR+36);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1140WSAEINTR

WSAEINTR = (WSABASEERR+4);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1141WSAEINVAL

WSAEINVAL = (WSABASEERR+22);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1142WSAEISCONN

WSAEISCONN = (WSABASEERR+56);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1143WSAELOOP

WSAELOOP = (WSABASEERR+62);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1144WSAEMFILE

WSAEMFILE = (WSABASEERR+24);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1145WSAEMSGSIZE

WSAEMSGSIZE = (WSABASEERR+40);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1146WSAENAMETOOLONG

WSAENAMETOOLONG = (WSABASEERR+63);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1147WSAENETDOWN

WSAENETDOWN = (WSABASEERR+50);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1148WSAENETRESET

WSAENETRESET = (WSABASEERR+52);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1149WSAENETUNREACH

WSAENETUNREACH = (WSABASEERR+51);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1150WSAENOBUFS

WSAENOBUFS = (WSABASEERR+55);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1151WSAENOPROTOOPT

WSAENOPROTOOPT = (WSABASEERR+42);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1152WSAENOTCONN

WSAENOTCONN = (WSABASEERR+57);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1153WSAENOTEMPTY

WSAENOTEMPTY = (WSABASEERR+66);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1154WSAENOTSOCK

WSAENOTSOCK = (WSABASEERR+38);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1155WSAEOPNOTSUPP

WSAEOPNOTSUPP = (WSABASEERR+45);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1156WSAEPFNOSUPPORT

WSAEPFNOSUPPORT = (WSABASEERR+46);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1157WSAEPROCLIM

WSAEPROCLIM = (WSABASEERR+67);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1158WSAEPROTONOSUPPORT

WSAEPROTONOSUPPORT = (WSABASEERR+43);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1159WSAEPROTOTYPE

WSAEPROTOTYPE = (WSABASEERR+41);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1160WSAEREMOTE

WSAEREMOTE = (WSABASEERR+71);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1161WSAESHUTDOWN

WSAESHUTDOWN = (WSABASEERR+58);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1162WSAESOCKTNOSUPPORT

WSAESOCKTNOSUPPORT = (WSABASEERR+44);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1163WSAESTALE

WSAESTALE = (WSABASEERR+70);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1164WSAETIMEDOUT

WSAETIMEDOUT = (WSABASEERR+60);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1165WSAETOOMANYREFS

WSAETOOMANYREFS = (WSABASEERR+59);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1166WSAEUSERS

WSAEUSERS = (WSABASEERR+68);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1167WSAEWOULDBLOCK

WSAEWOULDBLOCK = (WSABASEERR+35);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1168WSAHOST\_NOT\_FOUND

WSAHOST\_NOT\_FOUND = (WSABASEERR+1001);

**Description**

The text for this constant has been generated automatically. This means that it is not documented.



### 3.6.1169WSANO\_ADDRESS

```
WSANO_ADDRESS = WSANO_DATA  WSANO_DATA;
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1170WSANO\_DATA

```
WSANO_DATA = (WSABASEERR+1004);
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1171WSANO\_RECOVERY

```
WSANO_RECOVERY = (WSABASEERR+1003);
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1172WSANOTINITIALISED

```
WSANOTINITIALISED = (WSABASEERR+93);
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1173WSASYSNOTREADY

```
WSASYSNOTREADY = (WSABASEERR+91);
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1174WSATRY\_AGAIN

```
WSATRY_AGAIN = (WSABASEERR+1002);
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1175WSAVERNOTSUPPORTED

```
WSAVERNOTSUPPORTED = (WSABASEERR+92);
```

**Description**

The text for this constant has been generated automatically. This means that it is not documented.

### 3.6.1176wsErr

Represents the POP3 error code.

```
wsErr = 0;
```

**Description**

wsErr is a constant value used in POP3 development to test for failure using the method TldTCPConnection.SendCmd. wsErr is used to compensate for the lack of numeric response codes in the POP3 protocol.

### 3.6.1177wsOk

Represents the POP3 success code.

```
wsOk = 1;
```

**Description**

wsOk is constant value used in POP3 development to test for success when using the method TIdTCPConnection.SendCmd. wsOk is used to compensate for the lack of numeric response codes in the POP3 protocol.

3.6.1178XXCodeTable

Represents the character encoding table used for XXEncoding operations.

```
XXCodeTable: string = '+-
0123456789ABCDEFGHIJKLMN
PQRSTUVWXYZabcdefghijklmnopqrstuvwxyz';
```

**Description**  
XXCodeTable is a String constant that represents the character encoding table used for XXEncoding operations.

3.7. Units

3.7.1. IdAntiFreeze.pas

Unit: IdAntiFreeze.pas \*  
  
This file contains Classes, Types, Procedures, and Functions needed to define and implement the Indy AntiFreeze GUI-integration class.

3.7.2. IdAntiFreezeBase.pas

Unit: IdAntiFreezeBase.pas  
  
This file contains Classes, Types, Procedures, and Functions needed to define and implement the ancestor for the Indy GUI-integration class.

3.7.3. IdBaseComponent.pas

Unit: IdBaseComponent.pas  
  
This file contains Classes, Types, Procedures, and Functions needed to implement the ancestor class for all Indy components.

3.7.4. IdChargenServer.pas

Unit: IdChargenServer.pas  
  
This file contains Classes, Types, Procedures, and Functions needed to define and implement a Chargen server as defined in the Internet Standards document: RFC 864.

3.7.5. IdCoder.pas

Unit: IdCoder.pas  
  
This file contains Classes, Types, Procedures, and Functions needed to define and implement the ancestor for Indy coder classes.

3.7.6. IdCoder3To4.pas

Unit: IdCoder3To4.pas  
  
This file contains Classes, Types, Procedures, and Functions needed to define and implement Indy coder classes that support common character table-based encoding schemes.

3.7.7. **IdCoderIMF.pas**

Unit: IdCoderIMF.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement Internet Message Format coders.

3.7.8. **IdCoderMessageDigest.pas**

Unit: IdCoderMessageDigest.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement Message Digest-based coders.

3.7.9. **IdCoderText.pas**

Unit: IdCoderText.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement text-based coders.

3.7.10. **IdCompilerDefines.inc**

The text for this file has been generated automatically. This means that it is not documented.

3.7.11. **IdComponent.pas**

Unit: IdComponent.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement the Indy component class.

3.7.12. **IdDateTimeStamp.pas**

Unit: IdDateTimeStamp.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement storage for date and time values using the various formats required by some Internet Protocols.

3.7.13. **IdDayTime.pas**

Unit: IdDayTime.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a DayTime protocol (RFC 867) client.

3.7.14. **IdDayTimeServer.pas**

Unit: IdDayTimeServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a DayTime protocol server (RFC 867).

3.7.15. **IdDICTServer.pas**

Unit: IdDICTServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and

implement a Dictionary Protocol server (RFC 2229).

3.7.16. IdDiscardServer.pas

Unit: IdDiscardServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Discard Protocol server (RFC 863).

3.7.17. IdDNSResolver.pas

Unit: IdDNSResolver.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a UDP-based a resolver for DNS protocol queries (RFC 1034).

3.7.18. IdEcho.pas

Unit: IdEcho.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an Echo client (RFC 862).

3.7.19. IdEchoServer.pas

Unit: IdEchoServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an Echo Protocol server (RFC 862).

3.7.20. IdEmailAddress.pas

Unit: IdEmailAddress.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement classes to support RFC 822 email address items and list.

3.7.21. IdException.pas

Unit: IdException.pas

This file contains Classes, Types, Procedures, and Functions needed to define Indy exception classes.

3.7.22. IdFinger.pas

Unit: IdFinger.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Finger protocol client (RFC 1288).

3.7.23. IdFingerServer.pas

Unit: IdFingerServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Finger protocol server (RFC 1288).

3.7.24. IdFTP.pas

Unit: IdFTP.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a File Transfer Protocol (FTP) client (RFC 959).

3.7.25. IdGlobal.pas

Unit: IdGlobal.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement common routines and classes used in the Indy component suite.

3.7.26. IdGopher.pas

Unit: IdGopher.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Gopher client (RFC 1436).

3.7.27. IdGopherConsts.pas

Unit: IdGopherConsts.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a common Gopher protocol constants.

3.7.28. IdGopherServer.pas

Unit: IdGopherServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Gopher server (RFC 1436).

3.7.29. IdHeaderCoder.pas

Unit: IdHeaderCoder.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement support routines for working with encoded values in headers for an internet messages.

3.7.30. IdHeaderList.pas

Unit: IdHeaderList.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement collections of header names and values used in many protocols such as Gopher+, HTTP, NNTP, POP3, and SMTP.

3.7.31. IdHostnameServer.pas

Unit: IdHostnameServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a HostName server (RFC 953).

3.7.32. **IdHTTP.pas**

Unit: IdHTTP.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an HTTP client (RFC 1945, 2616, 2660).

3.7.33. **IdHTTPServer.pas**

Unit: IdHTTPServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an HTTP server (RFC 1945, 2616, 2660).

3.7.34. **IdIcmpClient.pas**

Unit: IdIcmpClient.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an ICMP client.

3.7.35. **IdIMAP4Server.pas**

Unit: IdIMAP4Server.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an IMAP4 server.

3.7.36. **IdIntercept.pas**

Unit: IdIntercept.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a framework that performs state change and data transformation tasks.

3.7.37. **IdIPWatch.pas**

Unit: IdIPWatch.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a class that determines the Online status, and IP history for a local computer.

3.7.38. **IdIrcServer.pas**

Unit: IdIrcServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an IRC server.

3.7.39. **IdLogBase.pas**

Unit: IdLogBase.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement the ancestor class for the Indy logging framework.

3.7.40. **IdLogDebug.pas**

Unit: IdLogDebug.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an Indy logging component.

3.7.41. **IdMappedPortTCP.pas**

Unit: IdMappedPortTCP.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Port-based proxy component.

3.7.42. **IdMessage.pas**

Unit: IdMessage.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement classes to support Internet messages and attachments.

3.7.43. **IdMessageClient.pas**

Unit: IdMessageClient.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an Internet Message client.

3.7.44. **IdMIMETypes.pas**

Unit: IdMIMETypes.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement support for MIME types.

3.7.45. **IdNetworkCalculator.pas**

Unit: IdNetworkCalculator.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an calculator for IP addresses and network masks.

3.7.46. **IdNNTP.pas**

Unit: IdNNTP.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an NNTP client.

3.7.47. **IdNNTPServer.pas**

Unit: IdNNTPServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an NNTP server.

3.7.48. IdPOP3.pas

Unit: IdPOP3.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a POP3 client.

3.7.49. IdQotd.pas

Unit: IdQotd.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a QOTD client.

3.7.50. IdQotdServer.pas

Unit: IdQotdServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a QOTD server.

3.7.51. IdRawBase.pas

Unit: IdRawBase.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement support for raw packet formats.

3.7.52. IdRawClient.pas

Unit: IdRawClient.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a client using raw sockets.

3.7.53. IdRawFunctions.pas

Unit: IdRawFunctions.pas

This file contains Classes, Types, Procedures, and Functions needed to support raw socket-based connections.

3.7.54. IdRawHeaders.pas

The text for this file has been generated automatically. This means that it is not documented.

3.7.55. IdResourceStrings.pas

The text for this file has been generated automatically. This means that it is not documented.

3.7.56. IdSimpleServer.pas

Unit: IdSimpleServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a simple TCP server.



3.7.57. IdSMTP.pas

Unit: IdSMTP.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an SMTP client.

3.7.58. IdSNTP.pas

Unit: IdSNTP.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement an SNTP client.

3.7.59. IdSocketHandle.pas

Unit: IdSocketHandle.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement bindings and binding collections.

3.7.60. IdSocks.pas

Unit: IdSocks.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement SOCKS proxy support.

3.7.61. IdSSLIntercept.pas

Unit: IdSSLIntercept.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement Secure Socket *↗*Socket Layer connection intercept support.

3.7.62. IdSSLOpenSSL.pas

Unit: IdSSLOpenSSL.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement the Open SSL implementation of the Secure Socket *↗*Socket Layer.

3.7.63. IdStack.pas

Unit: IdStack.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement support for the Internet protocol stack.

3.7.64. IdStackConsts.pas

The text for this file has been generated automatically. This means that it is not documented.

3.7.65. IdStackWinsock.pas

Unit: IdStackWinsock.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement support for the Winsock protocol stack.

3.7.66. **IdTCPClient.pas**

**Unit: IdTCPClient.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a TCP client.

3.7.67. **IdTCPConnection.pas**

**Unit: IdTCPConnection.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a TCP connection component.

3.7.68. **IdTCPServer.pas**

**Unit: IdTCPServer.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a TCP server.

3.7.69. **IdTelnet.pas**

**Unit: IdTelnet.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Teelnet client.

3.7.70. **IdTelnetServer.pas**

**Unit: IdTelnetServer.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Telnet server.

3.7.71. **IdThread.pas**

**Unit: IdThread.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement the Indy thread mamangement framework.

3.7.72. **IdThreadMgr.pas**

**Unit: IdThreadMgr.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement the default Indy thread manager.

3.7.73. **IdThreadMgrDefault.pas**

**Unit: IdThreadMgrDefault.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement the default Indy thread manager.

3.7.74. **IdThreadMgrPool.pas**

Unit: IdThreadMgrPool.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a pooled thread manager in Indy.

3.7.75. **IdTime.pas**

Unit: IdTime.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Time protocol client.

3.7.76. **IdTimeServer.pas**

Unit: IdTimeServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Time protocol server.

3.7.77. **IdTrivialFTP.pas**

Unit: IdTrivialFTP.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Trivial FTP client.

3.7.78. **IdTrivialFTPBase.pas**

Unit: IdTrivialFTPBase.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement the ancestor classes for Trival FTP support.

3.7.79. **IdTrivialFTPServer.pas**

Unit: IdTrivialFTPServer.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Trivial FTP server.

3.7.80. **IdTunnelCommon.pas**

Unit: IdTunnelCommon.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement common support for IP tunnels.

3.7.81. **IdTunnelMaster.pas**

Unit: IdTunnelMaster.pas

This file contains Classes, Types, Procedures, and Functions needed to define and implement the master controller for IP tunnel connections.

3.7.82. **IdTunnelSlave.pas**

**Unit: IdTunnelSlave.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement the slave controller for IP tunnel connections.

3.7.83. **IdUDPBase.pas**

**Unit: IdUDPBase.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement the ancestor classes for UDP connections.

3.7.84. **IdUDPClient.pas**

**Unit: IdUDPClient.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a UDP client.

3.7.85. **IdUDPServer.pas**

**Unit: IdUDPServer.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a UDP server.

3.7.86. **IdURI.pas**

**Unit: IdURI.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Universal Resource Identifier class.

3.7.87. **IdVCard.pas**

**Unit: IdVCard.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement VCard electronic business cards.

3.7.88. **IdWhois.pas**

**Unit: IdWhois.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Whois protocol client.

3.7.89. **IdWholsServer.pas**

**Unit: IdWholsServer.pas**

This file contains Classes, Types, Procedures, and Functions needed to define and implement a Whols protocol server.

### 3.7.90. IdWinsock.pas

The text for this file has been generated automatically. This means that it is not documented.

# Index

—

\_\_WSAFDIsSet 280  
\_TRANSMIT\_FILE\_BUFFERS 124

## A

Accept 280  
AcceptEx 280  
AF\_APPLETALK 295  
AF\_BAN 295  
AF\_CCITT 296  
AF\_CHAOS 296  
AF\_DATAKIT 296  
AF\_DECnet 296  
AF\_DLI 296  
AF\_ECMA 297  
AF\_FIREFOX 297  
AF\_HYLINK 297  
AF\_IMPLINK 297  
AF\_INET 298  
AF\_IPX 298  
AF\_ISO 298  
AF\_LAT 298  
AF\_MAX 298  
AF\_NETBIOS 299  
AF\_NS 299  
AF\_OSI 299  
AF\_PUP 299  
AF\_SNA 300  
AF\_UNIX 300  
AF\_UNKNOWN1 300  
AF\_UNSPEC 300  
AF\_VOICEVIEW 300  
AnsiSameText 162

## B

BACKSPACE 301  
base64\_tbi 301

Base64CodeTable 301  
Base64Encode 162  
Bind 280  
BUFFERLEN 302  
BytesReadType 302

## C

cA 303  
cAABit 303  
cAAMask 303  
CardinalRec 124  
cAXFR 303  
cCH 304  
cCS 304  
CHAR0 304  
CHAR32 305  
cHINFO 305  
cHS 305  
CldCoder 210  
cIN 305  
CloseSocket 281  
cMAILA 306  
cMAILB 306  
cMB 306  
cMD 306  
cMF 307  
cMG 307  
cMINFO 307  
cMR 308  
cMX 308  
CN\_CODED\_DATA 308  
CN\_CODING\_ENDED 308  
CN\_CODING\_STARTED 309  
CN\_DATA\_END\_FOUND 309  
CN\_DATA\_START\_FOUND 309  
CN\_IMF\_BODY\_PART\_END 310  
CN\_IMF\_BODY\_START 310  
CN\_IMF\_CODER\_START 310  
CN\_IMF\_DATA\_END 311  
CN\_IMF\_END\_MULTIPART 311  
CN\_IMF\_HEAD\_VALUE 311  
CN\_IMF\_NEW\_FILENAME 311  
CN\_IMF\_NEW\_MULTIPART 312  
CN\_NEW\_FILENAME 312  
CN\_UU\_BEGIN\_FOUND 312

CN\_UU\_CODER\_START 313  
CN\_UU\_END\_FOUND 313  
CN\_UU\_LAST\_CHAR\_FOUND 313  
CN\_UU\_NEW\_FILENAME 313  
CN\_UU\_PRIVILEGE\_ERROR 314  
CN\_UU\_PRIVILEGE\_FOUND 314  
CN\_UU\_TABLE\_BEGIN\_ABORT 314  
CN\_UU\_TABLE\_CHANGED 315  
CN\_UU\_TABLE\_FOUND 315  
cName 315  
cNS 315  
cNULL 316  
CoderCollective 281  
CommaSeperatedToStringList 163  
CompressedBytesType 316  
CompressionRatioType 316  
Connect 281  
ConstBoundary 317  
ConstContentDisposition 317  
ConstContentMD5 317  
ConstContentTransferEncoding 317  
ConstContentType 318  
ConstFileName 318  
ConstIMFBoundaryEnd 318  
ConstIMFMessageStart 319  
ConstIMFStart 319  
ConstName 319  
cOpCodeBits 319  
cOpCodeMask 320  
cOPCodeStrs 320  
CopyFileTo 163  
CP\_FALLBACK 320  
CP\_IMF 321  
CP\_STANDARD 321  
cPTR 321  
cQClassStr 321  
cQRBit 322  
cQRMask 322  
CR 322  
cRABit 323  
cRAMask 323  
cRCodeBits 323  
cRCodeFormatErr 323  
cRCodeMask 324  
cRCodeNameErr 324  
cRCodeNoError 324  
cRCodeNotImplemented 325

cRCodeRefused 325  
cRCodeServerErr 325  
cRCodeStrs 325  
cRDBit 326  
cRDMask 326  
cReslQuery 326  
cResQuery 327  
cResStatus 327  
CSET 210  
cSOA 327  
csSPECIALS 328  
cStar 328  
CT\_Creation 328  
CT\_Realisation 328  
cTCBit 329  
cTCMask 329  
CTL3To4 329  
cTXT 330  
CurrentProcessId 164  
cWKS 330

## D

DateTimeToGmtOffSetStr 164  
DateTimeToInternetStr 165  
DebugOutput 166  
Decode2022JP 166  
DecodeAddress 167  
DecodeAddresses 167  
DecodeHeader 168  
DEF\_PACKET\_SIZE 330

## E

EADDRINUSE 330  
EADDRNOTAVAIL 331  
EAFNOSUPPORT 331  
EALREADY 331  
ECONNABORTED 331  
ECONNREFUSED 331  
ECONNRESET 332  
EDESTADDRREQ 332  
EDQUOT 332  
EHOSTDOWN 332  
EHOSTUNREACH 333

EldAcceptWaitCannotBeModifiedWhileServerIsActive 8  
EldAlreadyConnected 8  
EldCanNotChangeTarget 8  
EldCanNotCreateMessagePart 9  
EldClosedSocket 9  
EldConnClosedGracefully 9  
EldCorruptServicesFile 10  
EldCouldNotBindSocket 10  
EldDnsResolverError 10  
EldEldTunnelConnectToMasterFailed 11  
EldException 11  
EldFailedToRetreiveTimeZoneInfo 11  
EldFTPFileAlreadyExists 12  
EldHTTPCannotSwitchSessionStateWhenActive 12  
EldHTTPHeaderAlreadyWritten 13  
EldHTTPServerError 13  
EldHTTPUnsupportedAuthorisationScheme 13  
EldIcmpException 14  
EldInterceptPropInvalid 14  
EldInterceptPropsNil 14  
EldInvalidServiceName 15  
EldInvalidSocket 15  
EldLoginException 15  
EldMaxLoginAttempt 16  
EldMessageException 16  
EldMoreThanOneTldAntiFreeze 16  
EldNNTPConnectionRefused 17  
EldNNTPException 17  
EldNNTPNoOnNewGroupsList 17  
EldNNTPNoOnNewNewsList 18  
EldNNTPNoOnNewsgroupList 18  
EldNNTPStringListNotInitialized 18  
EldNoBindingsSpecified 19  
EldNoDataToRead 19  
EldNoExecuteSpecified 19  
EldNoOnAuthentication 20  
EldNotAllBytesSent 20  
EldNotEnoughDataInBuffer 20  
EldObjectTypeNotSupported 21  
EldOpenSSLError 21  
EldOpenSSLLoadError 21  
EldOSSLAcceptError 22  
EldOSSLConnectError 22  
EldOSSLCouldNotLoadSSLLibrary 22  
EldOSSLCreatingContextError 23  
EldOSSLDataBindingError 23  
EldOSSLGetMethodError 23

EldOSSLLoadingCertError 24  
EldOSSLLoadingKeyError 24  
EldOSSLLoadingRootCertError 24  
EldOSSLMoDeNotSet 25  
EldOSSLSettingCipherError 25  
EldPackageSizeTooBig 25  
EldProtocolReplyError 26  
EldResponseError 26  
EldSetSizeExceeded 26  
EldSilentException 26  
EldSocketError 27  
EldSocksAuthError 27  
EldSocksAuthMethodError 28  
EldSocksError 28  
EldSocksRequestFailed 29  
EldSocksRequestIdentFailed 29  
EldSocksRequestServerFailed 29  
EldSocksServerAddressError 30  
EldSocksServerCommandError 30  
EldSocksServerConnectionRefusedError 30  
EldSocksServerGeneralError 31  
EldSocksServerHostUnreachableError 31  
EldSocksServerNetUnreachableError 31  
EldSocksServerPermissionError 32  
EldSocksServerRespondError 32  
EldSocksServerTTLExpiredError 32  
EldSocksUnknownError 33  
EldStackError 33  
EldStackInitializationFailed 34  
EldStackSizeExceeded 34  
EldTableNotFound 34  
EldTCPConnectionError 34  
EldTCPServerError 35  
EldTelnetClientConnectError 35  
EldTelnetError 36  
EldTelnetServerException 36  
EldTelnetServerOnDataAvailablesNil 36  
EldTextInvalidCount 36  
EldTFTPAccessViolation 37  
EldTFTPAllocationExceeded 37  
EldTFTPException 37  
EldTFTPFileAlreadyExists 38  
EldTFTPFileNotFound 38  
EldTFTPIllegalOperation 38  
EldTFTPNoSuchUser 38  
EldTFTPOptionNegotiationFailed 39  
EldTFTPUnknownTransferID 39

EldThreadClassNotSpecified 39  
EldThreadMgrError 40  
EldTunnelConnectToMasterFailed 40  
EldTunnelCRCFailed 40  
EldTunnelCustomMessageInterpretationFailure 41  
EldTunnelDontAllowConnections 41  
EldTunnelException 41  
EldTunnelInterpretationOfMessageFailed 42  
EldTunnelMessageHandlingFailed 42  
EldTunnelMessageTypeRecognitionError 42  
EldTunnelTransformError 43  
EldTunnelTransformErrorBeforeSend 43  
EldUDPException 43  
EldUDPReceiveErrorZeroBytes 44  
EldUDPServerException 44  
EINPROGRESS 333  
EISCONN 333  
ELOOP 333  
EMSGSIZE 333  
ENAMETOOLONG 334  
Encode2022JP 168  
EncodeAddress 169  
EncodeAddressItem 169  
EncodeHeader 170  
ENETDOWN 334  
ENETRESET 334  
ENETUNREACH 334  
ENOBUFS 335  
ENOPROTOOPT 335  
ENOTCONN 335  
ENOTEMPTY 335  
ENOTSOCK 335  
EOL 336  
EOPNOTSUPP 336  
EPFNOSUPPORT 336  
EPROCLIM 336  
EPROTONOSUPPORT 337  
EPROTOTYPE 337  
EREMOTE 337  
ErrAccessViolation 337  
ErrAllocationExceeded 338  
ErrFileAlreadyExists 338  
ErrFileNotFound 338  
ErrIllegalOperation 338  
ErrNoSuchUser 339  
ErrOptionNegotiationFailed 339  
ErrUndefined 339

ErrUnknownTransferID 340  
ESHUTDOWN 340  
ESOCKTNOSUPPORT 340  
ESTALE 340  
ETIMEDOUT 341  
ETOOMANYREFS 341  
EUSERS 341  
EWOULDBLOCK 341

F

FD\_ACCEPT 341  
FD\_CLOSE 342  
FD\_CLR 171  
FD\_ISSET 171  
FD\_OOB 342  
FD\_READ 342  
FD\_SET 172  
FD\_SETSIZE 343  
FD\_WRITE 343  
FD\_ZERO 172  
Fetch 172  
FileSizeByName 173  
FIOASYNC 343  
FIONBIO 343  
FIONREAD 344  
FreeAndNil 173

G

GAntiFreeze 281  
GContentType 344  
GetAcceptExSockaddrs 282  
GetHostByAddr 282  
GetHostByName 282  
GetHostByName 283  
GetMimeTypeFromFile 174  
GetPeerName 283  
GetProtoByname 283  
GetProtoByNumber 283  
GetQClassStr 174  
GetQTypeStr 175  
GetServByName 283  
GetServByPort 284  
GetSockName 284

GetSockOpt 284  
GetSystemLocale 175  
GetTickCount 176  
GFContentLength 344  
GFMage 345  
GFRequestedBlockSize 345  
GFTTL 345  
GLoginAttempts 345  
GmtOffsetStrToDateTime 177  
GMTToLocalDateTime 177  
GOffsetFromUTC 284  
GPathSep 346  
GReceiveTimeout 346  
GRecvBufferSizeDefault 346  
GResponseNo 347  
GSendBufferSizeDefault 347  
GServeFileProc 285  
GServerSoftware 347  
gsldProductName 348  
gsldVersion 348  
GStack 285  
GSystemLocale 285  
GTimeZoneBias 286  
GTransferMode 348

H

HalfCodeTable 348  
hdrsize 349  
HiLoBytes 125  
HiLoWords 125  
HOST\_NOT\_FOUND 349  
hostent 125  
Htonl 286  
Htons 286

I

ICMP\_MIN 349  
Id\_ARP\_HSIZE 350  
Id\_ARPHRD\_ETHER 350  
Id\_ARPOP\_INVREPLY 350  
Id\_ARPOP\_INVREQUEST 350  
Id\_ARPOP\_REPLY 350  
Id\_ARPOP\_REQUEST 351

Id\_ARPOP\_REVREPLY 351  
Id\_ARPOP\_REVREQUEST 351  
ID\_Default\_TIdAntiFreezeBase\_Active 351  
ID\_Default\_TIdAntiFreezeBase\_ApplicationHasPriority 352  
ID\_Default\_TIdAntiFreezeBase\_IdleTimeOut 352  
ID\_Default\_TIdAntiFreezeBase\_OnlyWhenIdle 352  
Id\_DNS\_HSIZE 353  
Id\_ETHER\_HSIZE 353  
Id\_ETHER\_ADDR\_LEN 353  
Id\_ETHERTYPE\_ARP 353  
Id\_ETHERTYPE\_IP 353  
Id\_ETHERTYPE\_LOOPBACK 354  
Id\_ETHERTYPE\_PUP 354  
Id\_ETHERTYPE\_REVARP 354  
Id\_ETHERTYPE\_VLAN 354  
Id\_ICMP\_ECHO 355  
Id\_ICMP\_ECHO\_HSIZE 355  
Id\_ICMP\_ECHOREPLY 355  
Id\_ICMP\_HSIZE 355  
Id\_ICMP\_IREQ 355  
Id\_ICMP\_IREQREPLY 356  
Id\_ICMP\_MASK\_HSIZE 356  
Id\_ICMP\_MASKREPLY 356  
Id\_ICMP\_MASKREQ 356  
Id\_ICMP\_PARAMPROB 357  
Id\_ICMP\_PARAMPROB\_OPTABSENT 357  
Id\_ICMP\_REDIRECT 357  
Id\_ICMP\_REDIRECT\_HOST 357  
Id\_ICMP\_REDIRECT\_HSIZE 357  
Id\_ICMP\_REDIRECT\_NET 358  
Id\_ICMP\_REDIRECT\_TOSHOST 358  
Id\_ICMP\_REDIRECT\_TOSNET 358  
Id\_ICMP\_ROUTERADVERT 358  
Id\_ICMP\_ROUTERSOLICIT 359  
Id\_ICMP\_SOURCEQUENCH 359  
Id\_ICMP\_TIMEXCEED\_HSIZE 359  
Id\_ICMP\_TIMXCEED 359  
Id\_ICMP\_TIMXCEED\_INTRANS 359  
Id\_ICMP\_TIMXCEED\_REASS 360  
Id\_ICMP\_TS\_HSIZE 360  
Id\_ICMP\_TSTAMP 360  
Id\_ICMP\_TSTAMPREPLY 360  
Id\_ICMP\_UNREACH 361  
Id\_ICMP\_UNREACH\_FILTER\_PROHIB 361  
Id\_ICMP\_UNREACH\_HOST 361  
Id\_ICMP\_UNREACH\_HOST\_PRECEDENCE 361  
Id\_ICMP\_UNREACH\_HOST\_PROHIB 361

Id\_ICMP\_UNREACH\_HOST\_UNKNOWN 362  
Id\_ICMP\_UNREACH\_HSIZE 362  
Id\_ICMP\_UNREACH\_ISOLATED 362  
Id\_ICMP\_UNREACH\_NEEDFRAG 362  
Id\_ICMP\_UNREACH\_NET 363  
Id\_ICMP\_UNREACH\_NET\_PROHIB 363  
Id\_ICMP\_UNREACH\_NET\_UNKNOWN 363  
Id\_ICMP\_UNREACH\_PORT 363  
Id\_ICMP\_UNREACH\_PRECEDENCE\_CUTOFF 363  
Id\_ICMP\_UNREACH\_PROTOCOL 364  
Id\_ICMP\_UNREACH\_SRCFAIL 364  
Id\_ICMP\_UNREACH\_TOSHOST 364  
Id\_ICMP\_UNREACH\_TOSNET 364  
Id\_IGMP\_HSIZE 365  
Id\_IGMP\_LEAVE\_GROUP 365  
Id\_IGMP\_MEMBERSHIP\_QUERY 365  
Id\_IGMP\_V1\_MEMBERSHIP\_REPORT 365  
Id\_IGMP\_V2\_MEMBERSHIP\_REPORT 365  
Id\_INADDR\_ANY 366  
Id\_INADDR\_NONE 366  
Id\_INVALID\_SOCKET 366  
Id\_IP\_DF 366  
Id\_IP\_HSIZE 367  
Id\_IP\_MAXPACKET 367  
Id\_IP\_MF 367  
Id\_IP\_OFFMASK 367  
Id\_IP\_RF 367  
Id\_IP\_TTL 368  
Id\_IPPROTO\_ICMP 368  
Id\_IPPROTO\_IGMP 368  
Id\_IPPROTO\_IP 368  
Id\_IPPROTO\_MAX 369  
Id\_IPPROTO\_RAW 369  
Id\_IPPROTO\_TCP 369  
Id\_IPPROTO\_UDP 369  
ID\_LOGBASE\_Active 369  
ID\_LOGBASE\_LogTime 370  
ID\_MAPPED\_PORT\_TCP\_PORT 370  
Id\_MAX\_IPOPTLEN 370  
ID\_MSG\_NODECODE 371  
ID\_MSG\_PRIORITY 371  
ID\_MSG\_USENOWFORDATE 371  
ID\_NC\_MASK\_LENGTH 372  
ID\_NETWORKCLASS 372  
Id\_PF\_INET 372  
Id\_RIP\_HSIZE 373  
Id\_RIPCMD\_MAX 373

Id\_RIPCMD\_POLL 373  
Id\_RIPCMD\_POLLENTRY 373  
Id\_RIPCMD\_REQUEST 373  
Id\_RIPCMD\_RESPONSE 374  
Id\_RIPCMD\_TRACEOFF 374  
Id\_RIPCMD\_TRACEON 374  
Id\_RIPVER\_0 374  
Id\_RIPVER\_1 375  
Id\_RIPVER\_2 375  
Id\_SD\_Both 375  
Id\_SD\_Recv 375  
Id\_SD\_Send 375  
ID\_SIMPLE\_SERVER\_BOUND\_PORT 376  
Id\_SO\_BROADCAST 376  
Id\_SO\_DEBUG 376  
Id\_SO\_DONTROUTE 376  
Id\_SO\_False 286  
Id\_SO\_KEEPAIVE 377  
Id\_SO\_LINGER 377  
Id\_SO\_OOBLINE 377  
Id\_SO\_RCVBUF 377  
Id\_SO\_RCVTIMEO 377  
Id\_SO\_REUSEADDR 378  
Id\_SO\_SNDBUF 378  
Id\_SO\_SNDTIMEO 378  
Id\_SO\_True 287  
Id\_SOCKET\_DGRAM 378  
Id\_SOCKET\_RAW 379  
Id\_SOCKET\_STREAM 379  
Id\_SOCKET\_ERROR 379  
ID\_SOCKS\_AUTH 379  
ID\_SOCKS\_PORT 380  
ID\_SOCKS\_VER 380  
Id\_SOL\_SOCKET 380  
Id\_TCP\_ACK 380  
Id\_TCP\_FIN 381  
Id\_TCP\_HSIZE 381  
Id\_TCP\_NODELAY 381  
Id\_TCP\_PUSH 381  
Id\_TCP\_RST 381  
Id\_TCP\_SYN 382  
Id\_TCP\_URG 382  
Id\_TId\_HTTPAutoStartSession 382  
Id\_TId\_HTTPServer\_ParseParams 382  
Id\_TId\_HTTPServer\_SessionState 383  
Id\_TId\_HTTPSessionTimeOut 383  
Id\_TIdFinger\_VerboseOutput 383



Id\_TIdFTP\_Passive 384  
Id\_TIdFTP\_TransferType 384  
Id\_TIdGopherServer\_TruncateLength 384  
Id\_TIdGopherServer\_TruncateUserFriendly 385  
Id\_TIdHTTP\_HandleRedirects 385  
Id\_TIdHTTP\_ProtocolVersion 385  
Id\_TIdHTTP\_RedirectMax 386  
Id\_TIDICMP\_ReceiveTimeout 386  
ID\_TIDLOGDEBUG\_TARGET 386  
Id\_TIdRawBase\_BufferSize 387  
Id\_TIdRawBase\_Port 387  
ID\_TIDSMTP\_AUTH\_TYPE 387  
ID\_UDP\_BUFFERSIZE 387  
Id\_UDP\_HSIZE 388  
Id\_WSAEACCES 388  
Id\_WSAEADDRINUSE 388  
Id\_WSAEADDRNOTAVAIL 388  
Id\_WSAEAFNOSUPPORT 389  
Id\_WSAEALREADY 389  
Id\_WSAEBADF 389  
Id\_WSAECONNABORTED 389  
Id\_WSAECONNREFUSED 389  
Id\_WSAECONNRESET 390  
Id\_WSAEDESTADDRREQ 390  
Id\_WSAEFAULT 390  
Id\_WSAEHOSTDOWN 390  
Id\_WSAEHOSTUNREACH 391  
Id\_WSAEINPROGRESS 391  
Id\_WSAEINTR 391  
Id\_WSAEINVAL 391  
Id\_WSAEISCONN 391  
Id\_WSAELOOP 392  
Id\_WSAEMFILE 392  
Id\_WSAEMSGSIZE 392  
Id\_WSAENAMETOOLONG 392  
Id\_WSAENETDOWN 393  
Id\_WSAENETRESET 393  
Id\_WSAENETUNREACH 393  
Id\_WSAENOBUFS 393  
Id\_WSAENOPROTOOPT 393  
Id\_WSAENOTCONN 394  
Id\_WSAENOTEMPTY 394  
Id\_WSAENOTSOCK 394  
Id\_WSAEOPNOTSUPP 394  
Id\_WSAEPFNOSUPPORT 395  
Id\_WSAEPROTONOSUPPORT 395  
Id\_WSAEPROTOTYPE 395

Id\_WSAESHUTDOWN 395  
Id\_WSAESOCKNOSUPPORT 395  
Id\_WSAETIMEDOUT 396  
Id\_WSAETOOMANYREFS 396  
Id\_WSAEWOULDBLOCK 396  
IdAntiFreeze.pas 585  
IdAntiFreezeBase.pas 585  
IdBaseComponent.pas 586  
IdBeatsInDay 396  
IdChargenServer.pas 586  
IdCoder.pas 586  
IdCoder3To4.pas 586  
IdCoderIMF.pas 587  
IdCoderMessageDigest.pas 587  
IdCoderText.pas 587  
IdCompilerDefines.inc 587  
IdComponent.pas 587  
IdDateTimeStamp.pas 588  
IdDayNames 397  
IdDayShortNames 397  
IdDaysInCentury 397  
IdDaysInFourYears 397  
IdDaysInLeapCentury 398  
IdDaysInLeapYear 398  
IdDaysInLeapYearCycle 398  
IdDaysInMonth 398  
IdDaysInShortLeapYearCycle 399  
IdDaysInShortNonLeapYearCycle 399  
IdDaysInWeek 399  
IdDaysInYear 399  
IdDayTime.pas 588  
IdDayTimeServer.pas 588  
IdDICTServer.pas 588  
IdDiscardServer.pas 589  
IdDNSResolver.pas 589  
IdDNSResolver\_ReceiveTimeout 400  
IdEcho.pas 589  
IdEchoServer.pas 589  
iDEFAULTPACKETSIZE 400  
iDEFAULTREPLYBUFSIZE 400  
IdEmailAddress.pas 590  
IdException.pas 590  
IdFinger.pas 590  
IdFingerServer.pas 590  
IdFTP.pas 591  
IdGlobal.pas 591  
IdGopher.pas 591

IdGopherConsts.pas 591  
IdGopherItem\_Binary 401  
IdGopherItem\_BinDOS 401  
IdGopherItem\_BinHex 401  
IdGopherItem\_CSO 401  
IdGopherItem\_Directory 401  
IdGopherItem\_Document 402  
IdGopherItem\_Error 402  
IdGopherItem\_GIF 402  
IdGopherItem\_HTML 402  
IdGopherItem\_Image 402  
IdGopherItem\_Image2 403  
IdGopherItem\_Information 403  
IdGopherItem\_MIME 403  
IdGopherItem\_Movie 403  
IdGopherItem\_Redundant 404  
IdGopherItem\_Search 404  
IdGopherItem\_Sound 404  
IdGopherItem\_Sound2 404  
IdGopherItem\_Telnet 405  
IdGopherItem\_TN3270 405  
IdGopherItem\_UUE 405  
IdGopherPlusAbstract 405  
IdGopherPlusAdmin 406  
IdGopherPlusAsk 406  
IdGopherPlusAskFileName 406  
IdGopherPlusAskLong 406  
IdGopherPlusAskPassword 407  
IdGopherPlusChoose 407  
IdGopherPlusChooseFile 407  
IdGopherPlusData\_BeginSign 408  
IdGopherPlusData\_EndSign 408  
IdGopherPlusData\_ErrorBeginSign 408  
IdGopherPlusData\_ErrorUnknownSize 409  
IdGopherPlusData\_UnknownSize 409  
IdGopherPlusDirectoryInformation 409  
IdGopherPlusError\_ItemMoved 409  
IdGopherPlusError\_NotAvailable 410  
IdGopherPlusError\_TryLater 410  
IdGopherPlusIndicator 410  
IdGopherPlusInfo 411  
IdGopherPlusInformation 411  
IdGopherPlusSelect 411  
IdGopherServer.pas 592  
IdHeaderCoder.pas 592  
IdHeaderList.pas 592  
IdHostnameServer.pas 592

IdHoursInDay 412  
IdHoursInHalfDay 412  
IdHTTP.pas 593  
IdHTTPServer.pas 593  
IdIcmpClient.pas 593  
IdIMAP4Server.pas 593  
IdIntercept.pas 594  
IdIPWatch.pas 594  
IdIrcServer.pas 594  
IdLogBase.pas 594  
IdLogDebug.pas 595  
IdMappedPortTCP.pas 595  
IdMessage.pas 595  
IdMessageClient.pas 595  
IdMillisecondsInDay 412  
IdMillisecondsInHour 412  
IdMillisecondsInMinute 413  
IdMilliSecondsInSecond 413  
IdMillisecondsInWeek 413  
IdMIMETypes.pas 596  
IdMinutesInHour 413  
IdMonthNames 414  
IdMonthShortNames 414  
IdMonthsInYear 414  
IdNetworkCalculator.pas 596  
IdNNTP.pas 596  
IdNNTPServer.pas 596  
IdPOP3.pas 597  
IdPORT\_AUTH 414  
IdPORT\_CHARGEN 415  
IdPORT\_DAYTIME 415  
IdPORT\_DICT 415  
IdPORT\_DISCARD 416  
IdPORT\_DOMAIN 416  
IdPORT\_FINGER 417  
IdPORT\_FTP 417  
IdPORT\_GOPHER 417  
IdPORT\_HOSTNAME 418  
IdPORT\_HTTP 418  
IdPORT\_IMAP4 418  
IdPORT\_IRC 419  
IdPORT\_LPD 419  
IdPORT\_NETSTAT 419  
IdPORT\_NNTP 419  
IdPORT\_POP2 420  
IdPORT\_POP3 420  
IdPORT\_QOTD 420

IdPORT_SMTP 421	IdStati 425
IdPORT_SNTP 421	IdTCPClient.pas 601
IdPORT_SSL 421	IdTCPConnection.pas 601
IdPORT_SYSTAT 422	IdTCPServer.pas 601
IdPORT_TELNET 422	IdTelnet.pas 601
IdPORT_TFTP 422	IdTelnetServer.pas 602
IdPORT_TIME 422	IdThread.pas 602
IdPORT_WHOIS 423	IdThreadMgr.pas 602
IdPorts 178	IdThreadMgrDefault.pas 602
IdQotd.pas 597	IdThreadMgrPool.pas 603
IdQotdServer.pas 597	IdTime.pas 603
IdRawBase.pas 597	IdTimeoutDefault 425
IdRawBuildArp 178	IdTimeoutInfinite 426
IdRawBuildDns 179	IdTimeServer.pas 603
IdRawBuildEthernet 179	IdTrivialFTP.pas 603
IdRawBuildIcmpEcho 179	IdTrivialFTPBase.pas 604
IdRawBuildIcmpMask 179	IdTrivialFTPServer.pas 604
IdRawBuildIcmpRedirect 180	IdTunnelCommon.pas 604
IdRawBuildIcmpTimeExceed 180	IdTunnelMaster.pas 604
IdRawBuildIcmpTimestamp 180	IdTunnelSlave.pas 605
IdRawBuildIcmpUnreach 180	IdUDPBase.pas 605
IdRawBuildIcmp 181	IdUDPClient.pas 605
IdRawBuildIp 181	IdUDPServer.pas 605
IdRawBuildRip 181	IdURI.pas 606
IdRawBuildTcp 181	IdVCard.pas 606
IdRawBuildUdp 182	IdWhois.pas 606
IdRawClient.pas 598	IdWhoisServer.pas 606
IdRawFunctions.pas 598	IdWinsock.pas 607
IdRawHeaders.pas 598	IdYearsInCentury 426
IdResourceStrings.pas 598	IdYearsInLeapYearCycle 426
IdSecondsInDay 423	IdYearsInShortLeapYearCycle 426
IdSecondsInHalfDay 423	IMAPCommands 427
IdSecondsInHour 424	IMPLINK_HIGHEXPER 427
IdSecondsInLeapYear 424	IMPLINK_IP 427
IdSecondsInMinute 424	IMPLINK_LOWEXPER 427
IdSecondsInWeek 424	in_addr 126
IdSecondsInYear 425	IncludeTrailingBackSlash 182
IdSimpleServer.pas 598	IncQWord 182
IdSMTP.pas 599	IndyPos 287
IdSNTP.pas 599	Inet_Addr 287
IdSocketHandle.pas 599	Inet_Ntoa 287
IdSocks.pas 599	InfoCallback 183
IdSSLIntercept.pas 600	InitializeMime 183
IdSSLOpenSSL.pas 600	InMainThread 184
IdStack.pas 600	Internet Direct Credits 2
IdStackConsts.pas 600	Internet Direct Sponsors 4
IdStackWinsock.pas 600	Introduction to Internet Direct 2

IntToBin 184  
INVALID\_SOCKET 428  
IOC\_IN 428  
IOC\_INOUT 428  
IOC\_OUT 428  
IOC\_VOID 428  
IOCPARM\_MASK 429  
loctlSocket 288  
IP\_ADD\_MEMBERSHIP 429  
IP\_DEFAULT\_MULTICAST\_LOOP 429  
IP\_DEFAULT\_MULTICAST\_TTL 429  
IP\_DONTFRAGMENT 430  
IP\_DROP\_MEMBERSHIP 430  
IP\_MAX\_MEMBERSHIPS 430  
IP\_MULTICAST\_IF 430  
IP\_MULTICAST\_LOOP 430  
IP\_MULTICAST\_TTL 431  
IP\_OPTIONS 431  
IP\_TOS 431  
IP\_TTL 431  
IP\_WATCH\_ACTIVE 432  
IP\_WATCH\_HIST\_ENABLED 432  
IP\_WATCH\_HIST\_FILENAME 432  
IP\_WATCH\_HIST\_MAX 433  
IP\_WATCH\_INTERVAL 433  
IPPORT\_RESERVED 433  
IPPROTO\_GGP 433  
IPPROTO\_ICMP 434  
IPPROTO\_IDP 434  
IPPROTO\_IGMP 434  
IPPROTO\_IP 434  
IPPROTO\_MAX 434  
IPPROTO\_ND 435  
IPPROTO\_PUP 435  
IPPROTO\_RAW 435  
IPPROTO\_TCP 435  
IPPROTO\_UDP 436  
IsCurrentThread 185  
IsNumeric 185

## K

kana\_tbl 436

**L**

- LF 437
- linger 126
- LoadWinsock 186
- LogicalAnd 186

## M

- MakeAckPkt 187
- MakeTempFilename 187
- Max 188
- MAX\_PACKET\_SIZE 437
- MAXGETHOSTSTRUCT 437
- MaxMIMEBinToASCIIType 437
- MaxMIMECompressType 438
- MaxMIMEEncType 438
- MaxMIMEMessageDigestType 438
- MaxMIMESubTypes 438
- MaxMIMETYPE 439
- maxPriv 439
- MaxWord 439
- MClientThread 44
- MIME7Bit 439
- MIMEEncBase64 440
- MIMEEncNISTSHA 440
- MIMEEncRLECompress 440
- MIMEEncRSAMD2 440
- MIMEEncRSAMD4 440
- MIMEEncRSAMD5 441
- MIMEEncUUEncode 441
- MIMEEncXXEncode 441
- MIMEFullApplicationOctetStream 441
- MIMEGenericText 441
- MIMEMediaType 288
- MIMESplit 442
- MIMESubMacBinHex40 442
- MIMESubOctetStream 442
- MIMETYPEApplication 442
- MIMETYPEAudio 442
- MIMETYPEImage 443
- MIMETYPEMessage 443
- MIMETYPEMultipart 443
- MIMETYPEText 443
- MIMETYPEVideo 443

MIMEXVal 444  
Min 188  
minPriv 444  
MSG\_DONTROUTE 444  
MSG\_MAXIOVLEN 444  
MSG\_OOB 445  
MSG\_PARTIAL 445  
MSG\_PEEK 445  
MultiPartAlternativeBoundary 445  
MultiPartBoundary 446  
MultiPartRelatedBoundary 446

## N

netent 127  
NO\_ADDRESS 446  
NO\_DATA 447  
NO\_RECOVERY 447  
nownCommands 436  
Ntohl 288  
Ntohs 289  
NTPMaxInt 447  
NumberOfConnectionsType 448  
NumberOfPacketsType 448  
NumberOfServicesType 448  
NumberOfSlavesType 449

## O

OffsetFromUTC 189

## P

ParseNewsGroup 189  
ParseURI 190  
ParseXOVER 191  
PasswordCallback 192  
PF\_APPLETALK 449  
PF\_BAN 449  
PF\_CCITT 449  
PF\_CHAOS 450  
PF\_DATAKIT 450  
PF\_DECnet 450  
PF\_DLI 450

PF\_ECMA 450  
PF\_FIREFOX 451  
PF\_HYLINK 451  
PF\_IMPLINK 451  
PF\_INET 451  
PF\_IPX 452  
PF\_ISO 452  
PF\_LAT 452  
PF\_MAX 452  
PF\_NS 452  
PF\_OSI 453  
PF\_PUP 453  
PF\_SNA 453  
PF\_UNIX 453  
PF\_UNKNOWN1 454  
PF\_UNSPEC 454  
PF\_VOICEVIEW 454  
PFDSet 211  
PHostEnt 211  
PIdArpHdr 211  
PIdBase64Decoder 211  
PIdBase64Encoder 211  
PIdCoder 212  
PIdCoderItem 212  
PIdDnsHdr 212  
PIdEthernetHdr 212  
PIdIcmpEcho 213  
PIdIcmpFrag 213  
PIdIcmpHdr 213  
PIdIcmpTs 213  
PIdIgmpHdr 213  
PIdInAddr 214  
PIdIpHdr 214  
PIdRipHdr 214  
PIdTcpHdr 214  
PIdUdpHdr 214  
PIdUUDecoder 215  
PIdUUEncoder 215  
PIdXXDecoder 215  
PIdXXEncoder 215  
PIMFCoderUsage 215  
PInAddr 216  
PLinger 216  
PNetEnt 216  
PosInStrArray 193  
PProtoEnt 216  
protoent 127

PServEnt 216  
PSOCKADDR 217  
PSockAddrIn 217  
PSockProto 217  
PTimeVal 217  
PTransmitFileBuffers 217

## R

Recv 289  
RecvFrom 289  
RegisterCoderClass 194  
ReturnMIMEType 195  
RightStr 195  
ROL 196  
ROR 196  
RPos 197  
RSAAboutBoxCompName 454  
RSAAboutBoxCopyright 454  
RSAAboutBoxIndyWebsite 455  
RSAAboutBoxPleaseVisit 455  
RSAAboutBoxVersion 455  
RSAAboutCreditsCoCoordinator 455  
RSAAboutCreditsCoordinator 456  
RSAAboutFormCaption 456  
RSAAboutMenuItemName 456  
RSAcceptWaitCannotBeModifiedWhileServerIsActive 456  
RSAlreadyConnected 457  
RSByteIndexOutOfBounds 457  
RSCannotAllocateSocket 457  
RSCannotChangeDebugTargetAtWhileActive 457  
RSCMDNotRecognized 458  
RSCodeNoError 458  
RSCodeQueryFormat 458  
RSCodeQueryName 458  
RSCodeQueryNotImplemented 458  
RSCodeQueryQueryRefused 459  
RSCodeQueryServer 459  
RSCodeQueryUnknownError 459  
RSCoderNoTableEntryNotFound 459  
RSConnectionClosedGracefully 460  
RSCorruptServicesFile 460  
RSCouldNotBindSocket 460  
RSCouldNotLoad 460  
RSDestinationFileAlreadyExists 461  
RSDNSMailAObsolete 461

RSDNSMailBNotImplemented 461  
RSDNSMDISObsolete 461  
RSDNSMFIsObsolete 461  
RSFailedTimeZoneInfo 462  
RSFTPUnknownHost 462  
RSGopherNotGopherPlus 462  
RSGopherServerNoProgramCode 462  
RSHTTPAccepted 463  
RSHTTTPBadGateway 463  
RSHTTTPBadRequest 463  
RSHTTTPCannotSwitchSessionStateWhenActive 463  
RSHTTTPChunkStarted 464  
RSHTTTPConflict 464  
RSHTTTPContinue 464  
RSHTTTPCreated 464  
RSHTTTPErrorParsingCommand 464  
RSHTTTPForbidden 465  
RSHTTTPGatewayTimeout 465  
RSHTTTPGone 465  
RSHTTTPHeaderAlreadyWritten 465  
RSHTTTPHTTPVersionNotSupported 466  
RSHTTTPInternalServerError 466  
RSHTTTPLengthRequired 466  
RSHTTTPMethodNotAllowed 466  
RSHTTTPMovedPermanently 466  
RSHTTTPMovedTemporarily 467  
RSHTTTPNoContent 467  
RSHTTTPNonAuthoritativeInformation 467  
RSHTTTPNotAcceptable 467  
RSHTTTPNotFound 468  
RSHTTTPNotImplemented 468  
RSHTTTPNotModified 468  
RSHTTTPOK 468  
RSHTTTPPartialContent 468  
RSHTTTPPreconditionFailed 469  
RSHTTTPProxyAuthenticationRequired 469  
RSHTTTPRequestEntityTooLong 469  
RSHTTTPRequestTimeout 469  
RSHTTTPRequestURITooLong 470  
RSHTTTPResetContent 470  
RSHTTTPSeeOther 470  
RSHTTTPServiceUnavailable 470  
RSHTTTPSwitchingProtocols 470  
RSHTTTPUnauthorized 471  
RSHTTTPUnknownResponseCode 471  
RSHTTTPUnsupportedAuthorisationScheme 471  
RSHTTTPUnsupportedMediaType 471

RSHTTPUseProxy 472  
RSICMPNonEchoResponse 472  
RSICMPNotEnoughBytes 472  
RSICMPReceiveError0 472  
RSICMPWrongDestination 472  
RSIdNoDataToRead 473  
RSInterceptPropInvalid 473  
RSInterceptPropsNil 473  
RSInvalidServiceName 473  
RSLPDAbortJob 474  
RSLPDClosingConnection 474  
RSLPDConnectTo 474  
RSLPDControlFileSaved 474  
RSLPDDataFileSaved 474  
RSLPDDirectoryDoesNotExist 475  
RSLPDNoQueuesDefined 475  
RSLPDQueueStatus 475  
RSLPDReceiveControlFile 475  
RSLPDReceiveDataFile 476  
RSLPDServerActive 476  
RSLPDServerStartTitle 476  
RSLPDUnknownQueue 476  
RSMsgClientEncodingAttachment 476  
RSMsgClientEncodingText 477  
RSMsgCmpEdtrBodyText 477  
RSMsgCmpEdtrExtraHead 477  
RSMsgCmpEdtrNew 477  
RSNETCALCInvalidNetworkMask 478  
RSNETCALCInvalidValueLength 478  
RSNETCALCConfirmLongIPList 478  
RSNETCALInvalidIPString 478  
RSNNTTPConnectionRefused 479  
RSNNTPNoOnNewGroupsList 479  
RSNNTPNoOnNewNewsList 479  
RSNNTPNoOnNewsgroupList 479  
RSNNTPServerGoodBye 480  
RSNNTPServerNotRecognized 480  
RSNNTPStringListNotInitialized 480  
RSNoBindingsSpecified 480  
RSNoExecuteSpecified 480  
RSNotAllBytesSent 481  
RSNotEnoughDataInBuffer 481  
RSTypeNotSupported 481  
RSOnExecuteNotAssigned 481  
RSOnlyOneAntiFreeze 482  
RSOSSLCertificateLookup 482  
RSOSSLConnectionDropped 482  
RSOSSLCouldNotLoadSSLLibrary 482  
RSOSSLInternal 483  
RSOSSLModeNotSet 483  
RSOSSLStatusString 483  
RSPackageSizeTooBig 483  
RSPOP3FieldNotSpecified 483  
RSQueryInvalidHeaderID 484  
RSQueryInvalidPacketSize 484  
RSQueryInvalidQueryCount 484  
RSQueryLessThanFour 484  
RSQueryLessThanTwelve 485  
RSQueryPackReceivedTooSmall 485  
RSRawReceiveError0 485  
RSSetSizeExceeded 485  
RSSocksAuthError 486  
RSSocksAuthMethodError 486  
RSSocksRequestFailed 486  
RSSocksRequestIdFailed 486  
RSSocksRequestServerFailed 487  
RSSocksServerAddressError 487  
RSSocksServerCommandError 487  
RSSocksServerConnectionRefusedError 487  
RSSocksServerGeneralError 488  
RSSocksServerHostUnreachableError 488  
RSSocksServerNetUnreachableError 488  
RSSocksServerPermissionError 488  
RSSocksServerRespondError 488  
RSSocksServerTTLExpiredError 489  
RSSocksUnknownError 489  
RSSSLAcceptError 489  
RSSSLConnectError 489  
RSSSLCreatingContextError 490  
RSSSLDataBindingError 490  
RSSSLGetMethodError 490  
RSSSLLoadingCertError 490  
RSSSLLoadingKeyError 490  
RSSSLLoadingRootCertError 491  
RSSSLSettingChiperError 491  
RSStackEACCES 491  
RSStackEADDRINUSE 491  
RSStackEADDRNOTAVAIL 492  
RSStackEAFNOSUPPORT 492  
RSStackEALREADY 492  
RSStackEBADF 492  
RSStackECONNABORTED 493  
RSStackECONNREFUSED 493  
RSStackECONNRESET 493

RSStackEDESTADDRREQ 493  
RSStackEDQUOT 493  
RSStackEFAULT 494  
RSStackEHOSTDOWN 494  
RSStackEHOSTUNREACH 494  
RSStackEINPROGRESS 494  
RSStackEINTR 495  
RSStackEINVAL 495  
RSStackEISCONN 495  
RSStackELOOP 495  
RSStackEMFILE 495  
RSStackEMSGSIZE 496  
RSStackENAMETOOLONG 496  
RSStackENETDOWN 496  
RSStackENETRESET 496  
RSStackENETUNREACH 497  
RSStackENOBUFS 497  
RSStackENOPROTOPT 497  
RSStackENOTCONN 497  
RSStackENOTEMPTY 497  
RSStackENOTSOCK 498  
RSStackEOPNOTSUPP 498  
RSStackEPROFNOSUPPORT 498  
RSStackEPROCLIM 498  
RSStackEPROTONOSUPPORT 499  
RSStackEPROTOTYPE 499  
RSStackEREMOTE 499  
RSStackError 499  
RSStackESHUTDOWN 499  
RSStackESOCKTNOSUPPORT 500  
RSStackESTALE 500  
RSStackETIMEDOUT 500  
RSStackETOOMANYREFS 500  
RSStackEUSERS 501  
RSStackEWOULDBLOCK 501  
RSStackHOST\_NOT\_FOUND 501  
RSStackNO\_DATA 501  
RSStackNO\_RECOVERY 501  
RSStackNOTINITIALISED 502  
RSStackSYSNOTREADY 502  
RSStackTRY\_AGAIN 502  
RSStackVERNONSUPPORTED 502  
RSStatusConnected 503  
RSStatusConnecting 503  
RSStatusDisconnected 503  
RSStatusDisconnecting 503  
RSStatusResolving 503  
RSStatusText 504  
RSTELNETCLIConnectError 504  
RSTELNETCLIReadError 504  
RSTELNETSRVInvalidLogin 504  
RSTELNETSRVMaxloginAttempt 505  
RSTELNETSRVNoAuthHandler 505  
RSTELNETSRVOnDataAvailableIsNil 505  
RSTELNETSRVPasswordPrompt 505  
RSTELNETSRVUsernamePrompt 506  
RSTELNETSRVWelcomeString 506  
RSTFTPAccessDenied 506  
RSTFTPDiskFull 506  
RSTFTPFileNotFound 507  
RSTFTPUnexpectedOp 507  
RSTFTPUnsupportedTrxMode 507  
RSThreadClassNotSpecified 507  
RSTIdMessagePartCreate 507  
RSTIdTextInvalidCount 508  
RSTimeOut 508  
RSTunnelConnectMsg 508  
RSTunnelConnectToMasterFailed 508  
RSTunnelCRCFailed 509  
RSTunnelDisconnectMsg 509  
RSTunnelDontAllowConnections 509  
RSTunnelGetByteRange 509  
RSTunnelMessageCustomInterpretError 510  
RSTunnelMessageHandlingError 510  
RSTunnelMessageInterpretError 510  
RSTunnelMessageTypeError 510  
RSTunnelTransformError 511  
RSTunnelTransformErrorBS 511  
RSUDPReceiveError0 511  
RSWinsockInitializationError 511  
RSWSockStack 511  
sBlockSize 512  
Select 289  
Send 289  
SendError 198-200  
SendTo 290  
servent 127  
SetLocalTime 201  
SetSockOpt 290  
SetThreadPriority 202

ShutDown 290  
sj1\_tbl 512  
sj2\_tbl 512  
Sleep 202  
SO\_ACCEPTCONN 513  
SO\_BROADCAST 514  
SO\_CONNDATA 514  
SO\_CONNDATALEN 514  
SO\_CONNECT\_TIME 514  
SO\_CONNOPT 514  
SO\_CONNOPTLEN 515  
SO\_DEBUG 515  
SO\_DISCDATA 515  
SO\_DISCDATALEN 515  
SO\_DISCOPT 516  
SO\_DISCOPTLEN 516  
SO\_DONTLINGER 516  
SO\_DONTROUTE 516  
SO\_ERROR 516  
SO\_KEEPALIVE 517  
SO\_LINGER 517  
SO\_MAXDG 517  
SO\_MAXPATHDG 517  
SO\_OOBINLINE 518  
SO\_OPENTYPE 518  
SO\_RCVBUF 518  
SO\_RCVLOWAT 518  
SO\_RCVTIMEO 518  
SO\_REUSEADDR 519  
SO\_SNDBUF 519  
SO\_SNDLOWAT 519  
SO\_SNDTIMEO 519  
SO\_SYNCHRONOUS\_ALERT 520  
SO\_SYNCHRONOUS\_NONALERT 520  
SO\_TYPE 520  
SO\_UPDATE\_ACCEPT\_CONTEXT 520  
SO\_USELOOPBACK 520  
SOCK\_DGRAM 521  
SOCK\_RAW 521  
SOCK\_RDM 521  
SOCK\_SEQPACKET 521  
SOCK\_STREAM 522  
sockaddr\_in 128  
Socket 290  
SOCKET\_ERROR 522  
sockproto 128  
SOL\_SOCKET 522

SOMAXCONN 522  
StrInternetToDateTime 203  
StrToCard 203  
StrToDay 204  
StrToMonth 204  
StrToWorld 205  
SunB 128  
SunW 129

T

T\_\_WSAFDIsSetProc 218  
T128BitRecord 218  
T160BitRecord 218  
T16x4LongWordRecord 218  
T384BitRecord 218  
T4x4LongWordRecord 219  
T4x4x4LongWordRecord 219  
T64BitRecord 219  
TAB 522  
TAcceptExProc 219  
TAcceptProc 220  
TAccessFileEvent 220  
TAREcord 45  
TAuthenticationEvent 220  
TAuthenticationType 221  
TBeforeClientConnectEvent 221  
TBindProc 222  
TByteArray 129  
TCallbackEvent 222  
TCharBuf 222  
TCharSet 223  
TClassIdException 223  
TClientData 45  
TClientEvent 223  
TClosesocketProc 224  
TCommandEvent 224  
TConnectionResult 225  
TConnectProc 225  
TCP\_BSDURGENT 523  
TCP\_NODELAY 523  
TDataEvent 225  
TDays 226  
TDoByIDEvent 226  
TDoByNoEvent 226  
Technical Support 5

TEventNewNewsList 227  
TEventNewsgroupList 227  
TEventStreaming 228  
TEVP\_MD 130  
TF\_DISCONNECT 523  
TF\_REUSE\_SOCKET 523  
TF\_WRITE\_BEHIND 524  
TFDSet 130  
TFTP\_ACK 524  
TFTP\_DATA 524  
TFTP\_ERROR 524  
TFTP\_OACK 525  
TFTP\_RRQ 525  
TFTP\_WRQ 525  
TGetAcceptExSockaddrsProc 228  
TGetEvent 228  
TGetHostByAddrProc 229  
TGetHostByNameProc 229  
TGetHostNameProc 229  
TGetPeerNameProc 229  
TGetProtoByNameProc 230  
TgetProtoByNumberProc 230  
TGetServByNameProc 230  
TGetServByPortProc 230  
TGetSockNameProc 230  
TGetSockOptProc 231  
TGroupEvent 231  
THInfo 130  
THInfoRecord 45  
THostEnt 231  
THostNameGetEvent 231  
THostNameOneParmEvent 232  
THtonIProc 232  
THtonsProc 233  
TICMPDataBuffer 233  
TId3To4Coder 46  
TIdAntiFreeze 46  
TIdAntiFreezeBase 47  
TIdArpHdr 131  
TIdASCIICoder 47  
TIdAttachment 47  
TIdBase64Decoder 48  
TIdBase64Encoder 48  
TIdBaseComponent 51  
TIdBuffer 51  
TIdCardAddressAttributes 233  
TIdCardAddressItem 52

TIdCardinalBytes 131  
TIdCardPhoneNumber 52  
TIdChargenServer 52  
TIdCoder 52  
TIdCoderCollection 53  
TIdCoderCRC16 53  
TIdCoderItem 54  
TIdCoderMD2 54  
TIdCoderMD4 55  
TIdCoderMD5 55  
TIdComponent 57  
TIdConnectionIntercept 57  
TIdConnectionInterceptOpenSSL 57  
TIdCookie 58  
TIdCookieCollection 58  
TIdDateTimeStamp 59  
TIdDayTime 59  
TIdDayTimeServer 60  
TIdDICTAuthEvent 234  
TIdDICTDefineEvent 234  
TIdDICTGetEvent 235  
TIdDICTMatchEvent 235  
TIdDICTOtherEvent 236  
TIdDICTServer 60  
TIdDICTShowEvent 236  
TIdDISCARDServer 61  
TIdDnsHdr 132  
TIdDNSHeader 61  
TIdDNSQuestionList 62  
TIdDNSResolver 62  
TIdDNSResourceList 63  
TIdEcho 63  
TIdECHOServer 64  
TIdEMailAddressItem 64  
TIdEMailAddressList 65  
TIdEtherAddr 132  
TIdEthernetHdr 132  
TIdExceptionEvent 237  
TIdFinger 65  
TIdFingerGetEvent 237  
TIdFingerServer 66  
TIdFTPTransferType 238  
TIdGopher 66  
TIdGopherMenu 67  
TIdGopherMenuEvent 238  
TIdGopherMenuItem 68  
TIdGopherServer 68

TidHeader 133  
TidHeaderInfo 69  
TidHeaderList 69  
TidHostNameServer 70  
TidHTTP 70  
TidHTTPGetEvent 238  
TidHTTPMethod 239  
TidHTTPOnRedirectEvent 239  
TidHTTPOtherEvent 240  
TidHTTPProtocolVersion 240  
TidHTTPRequestInfo 71  
TidHTTPResponseInfo 71  
TidHTTPServer 72  
TidHTTPSession 72  
TidHTTPSessionList 73  
TidIcmpClient 73  
TidIcmpEcho 133  
TidIcmpFrag 134  
TidIcmpHdr 134  
TidIcmpTs 134  
TidIgmpHdr 135  
TidIMAP4Server 74  
TidIMFDecoder 74  
TidIMFUUDecoder 75  
TidInAddr 135  
TidIpHdr 135  
TidIpOptions 136  
TidIPWatch 75  
TidIPWatchThread 76  
TidIrcFiveParmEvent 241  
TidIrcGetEvent 241  
TidIrcOneParmEvent 241  
TidIrcOtherEvent 242  
TidIRCServer 76  
TidIrcServerEvent 242  
TidIrcThreeParmEvent 242  
TidIrcTwoParmEvent 243  
TidIrcUserEvent 243  
TidLinger 243  
TidListenerThread 78  
TidLogBase 79  
TidLogDebugTarget 243  
TidMappedPortTCP 80  
TidMappedPortTCPData 80  
TidMessage 80  
TidMessageEvent 244  
TidMessagePart 82  
TidMessagePartClass 244  
TidMessageParts 82  
TidMessagePriority 244  
TidMimeTable 82  
TidNetTime 245  
TidNetworkCalculator 83  
TidNNTP 83  
TidNNTPServer 84  
TidPeerThread 85  
TidPhoneAttributes 245  
TidPID 246  
TidPOP3 85  
TidQOTD 86  
TidQOTDGetEvent 246  
TidQOTDServer 86  
TidQuotedPrintableDecoder 87  
TidQuotedPrintableEncoder 87  
TidRawBase 88  
TidRawClient 89  
TidRipHdr 136  
TidServeFile 247  
TidServerIntercept 89  
TidServerInterceptOpenSSL 90  
TidServerThreadEvent 247  
TidSimpleServer 90  
TidSMTP 91  
TidSNTP 91  
TidSocketHandle 92  
TidSocketHandles 92  
TidSocksRequest 137  
TidSocksResponse 137  
TidSSLAction 247  
TidSSLConnectionIntercept 93  
TidSSLContext 93  
TidSSLErrorMode 248  
TidSSLMode 248  
TidSSLOptions 93  
TidSSLServerIntercept 94  
TidSSLSocket 94  
TidSSLVerifyMode 248  
TidSSLVerifyModeSet 248  
TidSSLVersion 249  
TidStack 95  
TidStackSocketHandle 249  
TidStackVersion 95  
TidStackVersionWinsock 95  
TidStackWinsock 96

TidStatisticsOperation 249  
TidStatus 249  
TidStatusEvent 250  
TidStringMessageEvent 251  
TidSunB 138  
TidSunW 138  
TidTCPCClient 96  
TidTCPConnection 96  
TidTcpHdr 139  
TidTcpOptions 139  
TidTCPServer 97  
TidTCPServerConnection 97  
TidTelnet 98  
TidTelnetNegotiateEvent 251  
TidTelnetReadThread 98  
TidTelnetServer 99  
TidText 100  
TidTFTPMode 251  
TidThread 100  
TidThreadClass 252  
TidThreadMgr 101  
TidThreadMgrDefault 101  
TidThreadMgrPool 102  
TidThreadStopMode 252  
TidTime 103  
TidTimeServer 103  
TidTrivialFTP 104  
TidTrivialFTPServer 105  
TidTunnelMaster 105  
TidTunnelSlave 106  
TidUDPBase 107  
TidUDPCClient 108  
TidUdpHdr 139  
TidUDPLListenerThread 108  
TidUDPServer 109  
TidURI 109  
TidUUDecoder 109  
TidUUEncoder 110  
TidVCard 111  
TidVCardAddresses 112  
TidVCardBusinessInfo 112  
TidVCardEMailAddresses 112  
TidVCardEMailItem 113  
TidVCardEMailType 253  
TidVCardEmbeddedObject 113  
TidVCardGeog 113  
TidVCardMailingLabelItem 114  
TidVCardName 114  
TidVCardTelephones 114  
TidWhois 115  
TidWholsServer 115  
TidX509 115  
TidX509Name 116  
TidXXDecoder 116  
TidXXEncoder 116  
timeval 140  
TimeZoneBias 205  
TIMFCoderUsage 140  
TInAddr 253  
TInet\_AddrProc 253  
TInet\_NtoaProc 254  
TIntStringEvent 254  
TIoctlSocketProc 254  
TIpProperty 118  
TIpStruct 140  
TLinger 141  
TListenProc 254  
TLogger 119  
TLogItemEvent 255  
TLr 142  
tmConnect 525  
tmCustom 526  
tmData 526  
tmDisconnect 526  
tmError 526  
TMInfo 142  
TMInfoRecord 119  
TModeSetResult 255  
TModeType 256  
TMonths 256  
TMRecord 119  
TMX 143  
TMXRecord 120  
TNameRecord 120  
TNC\_AO 527  
TNC\_AYT 527  
TNC\_BREAK 527  
TNC\_DATA\_MARK 528  
TNC\_DO 528  
TNC\_DONT 528  
TNC\_EC 528  
TNC\_EL 529  
TNC\_EOR 529  
TNC\_GA 529

TNC\_IAC 530  
TNC\_IP 530  
TNC\_NOP 530  
TNC\_SB 530  
TNC\_SE 531  
TNC\_WILL 531  
TNC\_WONT 531  
TNetEnt 257  
TNetworkClass 257  
TNewsEvent 258  
TNewsTransportEvent 258  
TNO\_3270REGIME 532  
TNO\_AMSN 532  
TNO\_AUTH 532  
TNO\_BINARY 532  
TNO\_BYTE\_MACRO 533  
TNO\_DET 533  
TNO\_EA 533  
TNO\_ECHO 534  
TNO\_ENCRYPT 534  
TNO\_EOL 534  
TNO\_EOR 534  
TNO\_LINEMODE 535  
TNO\_LOGOUT 535  
TNO\_NAWS 535  
TNO\_OCRD 536  
TNO\_OFD 536  
TNO\_OHTD 536  
TNO\_OHTS 536  
TNO\_OLD 537  
TNO\_OLW 537  
TNO\_OM 537  
TNO\_OPS 538  
TNO\_OVT 538  
TNO\_OVTD 538  
TNO\_RCTE 538  
TNO\_RECONNECT 539  
TNO\_RFLOW 539  
TNO\_SGA 539  
TNO\_SL 540  
TNO\_STATUS 540  
TNO\_SUPDUP 540  
TNO\_SUPDUP\_OUTPUT 540  
TNO\_TACACS\_ID 541  
TNO\_TERM\_SPEED 541  
TNO\_TERMTYPE 541  
TNO\_TIMING\_MARK 542

TNO\_TLN 542  
TNO\_X3PAD 542  
TNO\_XDISPLOC 542  
TNOS\_NAME 543  
TNOS\_REPLY 543  
TNOS\_TERM\_IS 543  
TNOS\_TERMTYPE\_SEND 543  
TNtohlProc 258  
TNtohsProc 259  
TNTPGram 143  
TOnGetMessagePartStream 259  
TOnReplyEvent 259  
TOnSessionEndEvent 259  
TOnSessionStartEvent 260  
TOnTelnetCommand 260  
TOtherEvent 261  
TPasswordEvent 261  
TPeerInfo 155  
TPlusRequestEvent 261  
TPosProc 262  
TProceduralEvent 262  
TProtoEnt 262  
TPTRRecord 120  
TQuestionItem 121  
TQWord 155  
TRANSMIT\_FILE\_BUFFERS 263  
TransmitFile 291  
TRdata 156  
TReceiver 121  
TRecvFromProc 263  
TRecvProc 263  
TReplyStatus 157  
TReplyStatusTypes 263  
TRequestedRecord 264  
TRequestedRecords 264  
TRequestEvent 264  
TRY\_AGAIN 544  
TSelectProc 265  
TSender 121  
TSendMsgEvent 265  
TSendMsgEventC 265  
TSendProc 265  
TSendToProc 265  
TSendTrnEvent 266  
TSendTrnEventC 266  
TServEnt 266  
TSetSockOptProc 266

TShutDownProc 267  
TSlaveData 122  
TSlaveThread 122  
TSOA 157  
TSOARecord 123  
TSockAddr 267  
TSocket 267  
TSocketProc 267  
TSockProto 267  
TSocksAuthentication 268  
TSocksInfo 123  
TSocksVersion 268  
TStringEvent 268  
TTelnetCommand 269  
TTelnetData 123  
TThreadPriority 269  
TTimeVal 269  
TTnDataAvail 270  
TTnState 270  
TTransfer 271  
TTransferCompleteEvent 271  
TTransmitFileBuffers 271  
TTransmitFileProc 272  
TTunnelEvent 272  
TTunnelEventC 272  
TUDPRReadEvent 272  
TULong 158  
TVerifyPeerEvent 273  
TWKS 159  
TWKSBits 273  
TWKSRecord 124  
TWorkBeginEvent 273  
TWorkEndEvent 274  
TWorkEvent 274  
TWorkInfo 161  
TWorkMode 274  
TWSAAsyncGetHostByAddrProc 275  
TWSAAsyncGetHostByNameProc 275  
TWSAAsyncGetProtoByNameProc 275  
TWSAAsyncGetProtoByNumberProc 276  
TWSAAsyncGetServByNameProc 276  
TWSAAsyncGetServByPortProc 276  
TWSAAsyncSelectProc 276  
TWSACancelAsyncRequestProc 276  
TWSACancelBlockingCallProc 277  
TWSACleanupProc 277  
TWSAData 277

TWSAGetLastErrorProc 277  
TWSAIsBlockingProc 277  
TWSARecvExProc 278  
TWSASetBlockingHookProc 278  
TWSASetLastErrorProc 278  
TWSAStartupProc 278  
TWSAUnhookBlockingHookProc 278  
TZ\_ADT 544  
TZ\_AHST 544  
TZ\_AST 544  
TZ\_AT 545  
TZ\_BST 545  
TZ\_BT 545  
TZ\_CAT 545  
TZ\_CCT 545  
TZ\_CDT 546  
TZ\_CST 546  
TZ\_EADT 546  
TZ\_EAST 547  
TZ\_EDT 547  
TZ\_EET 547  
TZ\_EST 547  
TZ\_FST 547  
TZ\_FWT 548  
TZ\_GMT 548  
TZ\_GST 548  
TZ\_HDT 548  
TZ\_HST 549  
TZ\_IDLE 549  
TZ\_IDLW 549  
TZ\_JST 549  
TZ\_MDT 549  
TZ\_MEST 550  
TZ\_MESZ 550  
TZ\_MET 550  
TZ\_MEWT 550  
TZ\_MST 551  
TZ\_NT 551  
TZ\_NZDT 551  
TZ\_NZST 551  
TZ\_NZT 551  
TZ\_PDT 552  
TZ\_PST 552  
TZ\_SST 552  
TZ\_SWT 552  
TZ\_UT 553  
TZ\_UTC 553

TZ\_WADT 553  
TZ\_WAST 553  
TZ\_WAT 553  
TZ\_WET 554  
TZ\_YDT 554  
TZ\_YST 554  
TZ\_ZP4 554  
TZ\_ZP5 555  
TZ\_ZP6 555  
TZM\_A 555  
TZM\_Alpha 555  
TZM\_B 555  
TZM\_Bravo 556  
TZM\_C 556  
TZM\_Charlie 556  
TZM\_D 556  
TZM\_Delta 557  
TZM\_E 557  
TZM\_Echo 557  
TZM\_F 557  
TZM\_Foxtrot 557  
TZM\_G 558  
TZM\_Golf 558  
TZM\_H 558  
TZM\_Hotel 558  
TZM\_J 559  
TZM\_Juliet 559  
TZM\_K 559  
TZM\_Kilo 559  
TZM\_L 559  
TZM\_Lima 560  
TZM\_M 560  
TZM\_Mike 560  
TZM\_N 560  
TZM\_O 561  
TZM\_Oscar 561  
TZM\_P 561  
TZM\_Papa 561  
TZM\_Q 562  
TZM\_Quebec 562  
TZM\_R 562  
TZM\_Romeo 562  
TZM\_S 563  
TZM\_Sierra 563  
TZM\_T 563  
TZM\_Tango 563  
TZM\_U 563

TZM\_Uniform 564  
TZM\_V 564  
TZM\_Victor 564  
TZM\_W 564  
TZM\_Whiskey 565  
TZM\_X 565  
TZM\_XRay 565  
TZM\_Y 565  
TZM\_Yankee 565  
TZM\_Z 566  
TZM\_Zulu 566

## U

- u\_char 279
- u\_int 279
- u\_long 279
- u\_short 279
- UnloadWinsock 206
- UpCaseFirst 206
- URLDecode 206
- URLEncode 207
- UUBegin 566
- UUBEGINFound 566
- UUCodeTable 567
- UUDataStarted 567
- UUEnd 567
- UUEndFound 568
- UUErrIncompletePrivilege 568
- UUErrIncompletePrivilege2 568
- UUErrorDataEndWithoutEND 569
- UUErrorNoBEGINAfterTABLE 569
- UUErrorPrivilegeNotNumeric 569
- UUErrTableNotAtEnd 569
- UUInitialLength 570
- UULastCharFound 570
- UUPrivilegeFound 570
- UUStarted 571
- UUTable 571
- UUTableBeenRead 571
- UUTableBegin 571
- UUTableOneLine 572

## V

VerifyCallback 208  
vkana\_tbl 572

## W

What Is Internet Direct? 6  
WinsockLoaded 208  
WordRec 161  
WordStr 279  
WordToStr 208  
WSAAsyncGetHostByAddr 291  
WSAAsyncGetHostByName 291  
WSAAsyncGetProtoByName 291  
WSAAsyncGetProtoByNumber 292  
WSAAsyncGetServByName 292  
WSAAsyncGetServByPort 292  
WSAAsyncSelect 292  
WSABASEERR 572  
WSACancelAsyncRequest 293  
WSACancelBlockingCall 293  
WSACleanup 293  
WSAEACCES 573  
WSAEADDRINUSE 573  
WSAEADDRNOTAVAIL 573  
WSAEAFNOSUPPORT 573  
WSAEALREADY 573  
WSAEBADF 574  
WSAECONNABORTED 574  
WSAECONNREFUSED 574  
WSAECONNRESET 574  
WSAEDESTADDRREQ 575  
WSAEDISCON 575  
WSAEDQUOT 575  
WSAEFAULT 575  
WSAEHOSTDOWN 575  
WSAEHOSTUNREACH 576  
WSAEINPROGRESS 576  
WSAEINTR 576  
WSAEINVAL 576  
WSAEISCONN 577  
WSAELoop 577  
WSAEMFILE 577  
WSAEMSGSIZE 577

WSAENAMETOOLONG 577  
WSAENETDOWN 578  
WSAENETRESET 578  
WSAENETUNREACH 578  
WSAENOBUFS 578  
WSAENPROTOOPT 579  
WSAENOTCONN 579  
WSAENOTEMPTY 579  
WSAENOTSOCK 579  
WSAEOPNOTSUPP 579  
WSAEOPNOSUPPORT 580  
WSAEPROCLIM 580  
WSAEPROTONOSUPPORT 580  
WSAEPROTOTYPE 580  
WSAEREMOTE 581  
WSAESHUTDOWN 581  
WSAESOCKTNOSUPPORT 581  
WSAESTALE 581  
WSAETIMEDOUT 581  
WSAETOOMANYREFS 582  
WSAEUSERS 582  
WSAEWOULDBLOCK 582  
WSAGetAsyncBuflen 209  
WSAGetAsyncError 209  
WSAGetLastError 293  
WSAGetSelectError 209  
WSAGetSelectEvent 209  
WSAHOST\_NOT\_FOUND 582  
WSAIsBlocking 294  
WSAMakeSelectReply 210  
WSAMakeSyncReply 210  
WSANO\_ADDRESS 583  
WSANO\_DATA 583  
WSANO\_RECOVERY 583  
WSANOTINITIALISED 583  
WSARecvEx 294  
WSASetBlockingHook 294  
WSASetLastError 294  
WSAStartup 295  
WSATRY\_AGAIN 584  
WSAUnhookBlockingHook 295  
WSAVERNOTSUPPORTED 584  
wsErr 584  
wsOk 584



X

XXCodeTable 585