SIEMENS EDA

Automation Pro Batch Engines Reference

Release X-ENTP VX.2.13
Document Revision 10

SIEMENS

Unpublished work. © 2023 Siemens

This Documentation contains trade secrets or otherwise confidential information owned by Siemens Industry Software Inc. or its affiliates (collectively, “Siemens”), or its licensors. Access to and use of this Documentation is strictly limited as set forth in Customer’s applicable agreement(s) with Siemens. This Documentation may not be copied, distributed, or otherwise disclosed by Customer without the express written permission of Siemens, and may not be used in any way not expressly authorized by Siemens.

This Documentation is for information and instruction purposes. Siemens reserves the right to make changes in specifications and other information contained in this Documentation without prior notice, and the reader should, in all cases, consult Siemens to determine whether any changes have been made.

No representation or other affirmation of fact contained in this Documentation shall be deemed to be a warranty or give rise to any liability of Siemens whatsoever.

If you have a signed license agreement with Siemens for the product with which this Documentation will be used, your use of this Documentation is subject to the scope of license and the software protection and security provisions of that agreement. If you do not have such a signed license agreement, your use is subject to the Siemens Universal Customer Agreement, which may be viewed at https://www.sw.siemens.com/en-US/sw-terms/base/uca/, as supplemented by the product specific terms which may be viewed at https://www.sw.siemens.com/en-US/sw-terms/supplements/.

SIEMENS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS DOCUMENTATION INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. SIEMENS SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, LOST DATA OR PROFITS, EVEN IF SUCH DAMAGES WERE FORESEEABLE, ARISING OUT OF OR RELATED TO THIS DOCUMENTATION OR THE INFORMATION CONTAINED IN IT, EVEN IF SIEMENS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TRADEMARKS: The trademarks, logos, and service marks (collectively, "Marks") used herein are the property of Siemens or other parties. No one is permitted to use these Marks without the prior written consent of Siemens or the owner of the Marks, as applicable. The use herein of third party Marks is not an attempt to indicate Siemens as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A list of Siemens' Marks may be viewed at: www.plm.automation.siemens.com/global/en/legal/trademarks.html. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

About Siemens Digital Industries Software

Siemens Digital Industries Software is a leading global provider of product life cycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens Digital Industries Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens Digital Industries Software products and services, visit www.siemens.com/plm.

Support Center: support.sw.siemens.com
Send Feedback on Documentation: support.sw.siemens.com/doc_feedback_form

Revision History ISO-26262

Revision	Changes	Status/Date
10	Modifications to improve the readability and comprehension of the content. Approved by Kevin Chupp. All technical enhancements, changes, and fixes listed in the *Xpedition Enterprise Flow Release Notes* for this product are reflected in this document. Approved by Mike Bare.	Released March 2023
9	Modifications to improve the readability and comprehension of the content. Approved by Kevin Chupp. All technical enhancements, changes, and fixes listed in the *Xpedition Enterprise Flow Release Notes* for this product are reflected in this document. Approved by Mike Bare.	Released September 2022
8	Modifications to title page to reflect the latest product version supported. Approved by Kevin Chupp. All technical enhancements, changes, and fixes listed in the *Xpedition Enterprise Flow Release Notes* for this product are reflected in this document. Approved by Mike Bare.	Released March 2022
7	Modifications to title page to reflect the latest product version supported. Approved by Kevin Chupp. All technical enhancements, changes, and fixes listed in the *Xpedition Enterprise Flow Release Notes* for this product are reflected in this document. Approved by Mike Bare.	Released September 2021

Author: In-house procedures and working practices require multiple authors for documents. All associated authors for each topic within this document are tracked within the Siemens documentation source. For specific topic authors, contact the Siemens Digital Industries Software documentation department.

Revision History: Released documents include a revision history of up to four revisions. For earlier revision history, refer to earlier releases of documentation on Support Center.

Table of Contents

Chapter 1
Batch Engines Overview....17
Engines Automation Usage....18
Engine Error Access Through Automation....20

Chapter 2
Batch Engines Objects....23
Batch Engines Object Descriptions....24
DFFDRC Object....26
Go Method (DFFDRC Object)....30
DesignFileName Property (DFFDRC Object)....31
Errors Property (DFFDRC Object)....32
SetupFileName Property (DFFDRC Object)....33
DRC Object....34
Go Method (DRC Object)....40
DefaultClearanceValue Property (DRC Object)....41
DesignFileName Property (DRC Object)....42
ElementToElement Property (DRC Object)....43
ElementToElementLayerSpecific Property (DRC Object)....44
Errors Property (DRC Object)....45
Layers Property (DRC Object)....46
MinAnnularRing Property (DRC Object)....47
MissingConductivePads Property (DRC Object)....48
MissingConductivePadsOnLayers Property (DRC Object)....49
MissingCoverLayerPads Property (DRC Object)....50
MissingSolderPads Property (DRC Object)....51
NetRules Property (DRC Object)....52
Nets Property (DRC Object)....53
PadRules Property (DRC Object)....54
PadToPad Property (DRC Object)....55
PlaneRules Property (DRC Object)....56
Rules Property (DRC Object)....57
SetupElementsFileName Property (DRC Object)....58
SetupFileName Property (DRC Object)....59
TraceRules Property (DRC Object)....60
UnplacedParts Property (DRC Object)....61
UseDRCWindows Property (DRC Object)....62
UserLayerToElement Property (DRC Object)....63
UserLayerToElementLayerSpecific Property (DRC Object)....64
UserLayerToUserLayer Property (DRC Object)....65
ViasUnder Property (DRC Object)....66
DrillHole Object....67
CombineCharAndSize Property (DrillHole Object)....68
DrillSymbol Property (DrillHole Object)....69

DrillSymbolSize Property (DrillHole Object)..... 70
DrillSymbolType Property (DrillHole Object)..... 71
FCF Property (DrillHole Object)..... 72
Name Property (DrillHole Object)..... 73
DrawingFileWizard Object..... 74
CentralLibFileName Property (DrawingFileWizard Object)..... 80
DrawingTask Property (DrawingFileWizard Object)..... 81
EnableEssentialFiles Property (DrawingFileWizard Object)..... 82
Errors Property (DrawingFileWizard Object)..... 83
Go Method (DrawingFileWizard Object)..... 84
SourceMDFFileName Property (DrawingFileWizard Object)..... 85
TargetMDFFileName Property (DrawingFileWizard Object)..... 86
DXFExport Object..... 87
BoardContents Property (DXFExport Object)..... 89
DesignFileName Property (DXFExport Object)..... 90
DXFUnits Property (DXFExport Object)..... 91
Errors Property (DXFExport Object)..... 92
FillPads Property (DXFExport Object)..... 93
Go Method (DXFExport Object)..... 94
LayerCount Property (DXFExport Object)..... 95
MirrorOutput Property (DXFExport Object)..... 96
OutputFileName Property (DXFExport Object)..... 97
PutElementExport Method (DXFExport Object)..... 98
PutWBElementExport Method (DXFExport Object)..... 99
Scheme Property (DXFExport Object)..... 100
Error Object..... 101
ErrorString Property (Error Object)..... 102
Severity Property (Error Object)..... 103
ExtendedPrint Object..... 104
AdvancedSettings Property (ExtendedPrint Object)..... 107
BasicConfigurationFile Property (ExtendedPrint Object)..... 108
DesignFileName Property (ExtendedPrint Object)..... 109
Errors Property (ExtendedPrint Object)..... 110
FileFormat Property (ExtendedPrint Object)..... 111
Go Method (ExtendedPrint Object)..... 112
LogFileName Property (ExtendedPrint Object)..... 113
OutputFileName Property (ExtendedPrint Object)..... 114
SetupFileName Property (ExtendedPrint Object)..... 115
FCF Object..... 116
CharSymbol Property (FCF Object)..... 117
DatumReferenceA Property (FCF Object)..... 118
DatumReferenceB Property (FCF Object)..... 119
DatumReferenceC Property (FCF Object)..... 120
Tolerance Property (FCF Object)..... 121
GeneralInterfaceEngine Object..... 122
AddVariant Method (GeneralInterfaceEngine Object)..... 124
ClearVariants Method (GeneralInterfaceEngine Object)..... 125
Go Method (GeneralInterfaceEngine Object)..... 126
AllowNegativeCoords Property (GeneralInterfaceEngine Object)..... 127

CellOutline Property (GeneralInterfaceEngine Object)....128
CreatePadShapeFile Property (GeneralInterfaceEngine Object)....129
DesignFileName Property (GeneralInterfaceEngine Object)....130
DesignVariantNames Property (GeneralInterfaceEngine Object)....131
Errors Property (GeneralInterfaceEngine Object)....132
GenerateUniqueNetNamesForUnconnectedNets Property (GeneralInterfaceEngine Object)....133
IdentifyPartBy Property (GeneralInterfaceEngine Object)....134
IncludePartAttributes Property (GeneralInterfaceEngine Object)....135
IncludeVariantData Property (GeneralInterfaceEngine Object)....136
OutputDir Property (GeneralInterfaceEngine Object)....137
OutputFiducialAsSMDComponent Property (GeneralInterfaceEngine Object)....138
OutputMountingHolesAsVias Property (GeneralInterfaceEngine Object)....139
OutputTestpointAsComponent Property (GeneralInterfaceEngine Object)....140
OutputType Property (GeneralInterfaceEngine Object)....141
UnitsType Property (GeneralInterfaceEngine Object)....142
UseTemplatedComponents Property (GeneralInterfaceEngine Object)....143
Gerber Object....144
AddVariant Method (Gerber Object)....151
ClearVariants Method (Gerber Object)....152
Go Method (Gerber Object)....153
DesignFileName Property (Gerber Object)....154
DesignVariantNames Property (Gerber Object)....155
Errors Property (Gerber Object)....156
GenerateMacros Property (Gerber Object)....157
IncludeVariantData Property (Gerber Object)....158
MachineFormat Property (Gerber Object)....159
OffsetX Property (Gerber Object)....160
OffsetY Property (Gerber Object)....161
OutputFiles Property (Gerber Object)....162
SetupFileName Property (Gerber Object)....163
SpaceBetweenCopiesX Property (Gerber Object)....164
SpaceBetweenCopiesY Property (Gerber Object)....165
XAxisCopies Property (Gerber Object)....166
YAxisCopies Property (Gerber Object)....167
GerberMachineFormat Object....168
ArcStyle Property (GerberMachineFormat Object)....170
CharacterSet Property (GerberMachineFormat Object)....171
ClearShapeFillWidths Method (GerberMachineFormat Object)....172
Comments Property (GerberMachineFormat Object)....173
DataFormatLeadingDigits Property (GerberMachineFormat Object)....174
DataFormatTrailingDigits Property (GerberMachineFormat Object)....175
DataMode Property (GerberMachineFormat Object)....176
DataType Property (GerberMachineFormat Object)....177
DefaultDrawWidth Property (GerberMachineFormat Object)....178
Delimiter Property (GerberMachineFormat Object)....179
FileName Property (GerberMachineFormat Object)....180
PolygonFillMethod Property (GerberMachineFormat Object)....181
RecordLength Property (GerberMachineFormat Object)....182
SequenceNumbering Property (GerberMachineFormat Object)....183

ShapeFillWidths Property (GerberMachineFormat Object)..... 184
StepMode Property (GerberMachineFormat Object).....185
Unit Property (GerberMachineFormat Object).....186
ZeroTruncation Property (GerberMachineFormat Object).....187
GerberOutputFile Object..... 188
BoardItems Property (GerberOutputFile Object)..... 192
CellItems Property (GerberOutputFile Object)..... 193
CellItemsLayer Property (GerberOutputFile Object)..... 194
CellTypes Property (GerberOutputFile Object).....195
ConductiveItems Property (GerberOutputFile Object).....196
ConductiveLayer Property (GerberOutputFile Object)..... 197
DCodeMappingFileName Property (GerberOutputFile Object)..... 198
FlashPads Property (GerberOutputFile Object)..... 199
HeaderText Property (GerberOutputFile Object).....200
Name Property (GerberOutputFile Object).....201
ProcessUnconnectedPads Property (GerberOutputFile Object)..... 202
TrailerText Property (GerberOutputFile Object).....203
UserLayers Property (GerberOutputFile Object)..... 204
IDFExport Object.....205
Go Method (IDFExport Object).....207
BoardFilesDirectory Property (IDFExport Object)..... 208
BoardOutputFile Property (IDFExport Object).....209
BoardThickness Property (IDFExport Object)..... 210
ComponentSide Property (IDFExport Object)..... 211
DefaultComponentHeight Property (IDFExport Object).....212
DesignFileName Property (IDFExport Object)..... 213
Errors Property (IDFExport Object)..... 214
ExcludeCellRouteObstructs Property (IDFExport Object)..... 215
ExportComponentPinAndViaDrilledHoles Property (IDFExport Object)..... 216
ExportComponentPlacementOutlineWithHeight Property (IDFExport Object)..... 217
ExportSection Property (IDFExport Object).....218
LibraryOutputFile Property (IDFExport Object)..... 219
OtherOutlineLayer Property (IDFExport Object).....220
PackageOutlineLayer Property (IDFExport Object).....221
SetupFileName Property (IDFExport Object)..... 222
UseBoardThicknessFromStackup Property (IDFExport Object).....223
Version Property (IDFExport Object).....224
IOASISExportEngine Object..... 225
DesignFileName Property (IOASISExportEngine Object)..... 226
Errors Property (IOASISExportEngine Object).....227
Go Method (IOASISExportEngine Object)..... 228
OASISFileName Property (IOASISExportEngine Object)..... 229
SetupFileName Property (IOASISExportEngine Object).....230
ManufacturingOutputValidation Object.....231
Go Method (ManufacturingOutputValidation Object).....235
DesignFileName Property (ManufacturingOutputValidation Object).....236
Errors Property (ManufacturingOutputValidation Object)..... 237
GerberPlotEngine Property (ManufacturingOutputValidation Object).....238
NCDrillEngine Property (ManufacturingOutputValidation Object).....239

PrecisionLimit Property (ManufacturingOutputValidation Object)....240
SetupFileName Property (ManufacturingOutputValidation Object).... 241
UserInformation Property (ManufacturingOutputValidation Object)....242
NCDrill Object.... 243
Go Method (NCDrill Object).... 248
Chart Property (NCDrill Object)....249
DesignFileName Property (NCDrill Object).... 250
Errors Property (NCDrill Object)....251
MachineFormat Property (NCDrill Object)....252
OutputDirectory Property (NCDrill Object)....253
Parameters Property (NCDrill Object).... 254
SetupFileName Property (NCDrill Object)....255
UseWorkDirectory Property (NCDrill Object)....256
NCDrillChart Object....257
ResetTolerance Method (NCDrillChart Object)....259
AllSpansOnSingleChart Property (NCDrillChart Object).... 260
AssignDrillCharacters Property (NCDrillChart Object).... 261
Columns Property (NCDrillChart Object)....262
DrillHoles Property (NCDrillChart Object).... 263
DrillSymbolsOnSeparateLayers Property (NCDrillChart Object)....264
FCFFontName Property (NCDrillChart Object).... 265
FontName Property (NCDrillChart Object).... 266
FontSize Property (NCDrillChart Object)....267
HorizontalLineSeparator Property (NCDrillChart Object).... 268
IncludeContours Property (NCDrillChart Object)....269
LineSpacing Property (NCDrillChart Object).... 270
NegativeTolerance Property (NCDrillChart Object).... 271
PenWidth Property (NCDrillChart Object).... 272
PositiveTolerance Property (NCDrillChart Object)....273
PrecisionLeadingDigits Property (NCDrillChart Object).... 274
PrecisionTrailingDigits Property (NCDrillChart Object)....275
SpecialNotes Property (NCDrillChart Object)....276
TextFormatTitleCase Property (NCDrillChart Object)....277
Title Property (NCDrillChart Object).... 278
UnitEx Property (NCDrillChart Object).... 279
VerticalLineSeparator Property (NCDrillChart Object)....280
NCDrillChartColumn Object.... 281
Alignment Property (NCDrillChartColumn Object)....282
ColumnType Property (NCDrillChartColumn Object).... 283
DisplayName Property (NCDrillChartColumn Object).... 284
Name Property (NCDrillChartColumn Object)....285
PutUserColumnInput Method (NCDrillChartColumn Object)....286
UserColumnInput Property (NCDrillChartColumn Object)....287
NCDrillMachineFormat Object.... 288
ArcStyle Property (NCDrillMachineFormat Object).... 289
CharacterSet Property (NCDrillMachineFormat Object)....290
Comments Property (NCDrillMachineFormat Object).... 291
CommentStr Property (NCDrillMachineFormat Object)....292
DataFormatLeadingDigits Property (NCDrillMachineFormat Object).... 293

DataFormatTrailingDigits Property (NCDrillMachineFormat Object)..... 294
DataMode Property (NCDrillMachineFormat Object)..... 295
DataType Property (NCDrillMachineFormat Object)..... 296
Delimiter Property (NCDrillMachineFormat Object).....297
FileName Property (NCDrillMachineFormat Object)..... 298
RecordLength Property (NCDrillMachineFormat Object)..... 299
SequenceNumbering Property (NCDrillMachineFormat Object)..... 300
StepMode Property (NCDrillMachineFormat Object)..... 301
Unit Property (NCDrillMachineFormat Object)..... 302
ZeroTruncation Property (NCDrillMachineFormat Object).....303
NCDrillParameters Object.....304
AddFileHeader Method (NCDrillParameters Object).....305
AddFileNotes Method (NCDrillParameters Object)..... 306
ClearFileHeader Method (NCDrillParameters Object).....307
ClearFileNotes Method (NCDrillParameters Object).....308
Bandwidth Property (NCDrillParameters Object)..... 309
OutputFileExtension Property (NCDrillParameters Object).....310
PreDrillHolesLargerThan Property (NCDrillParameters Object).....311
SweepAxis Property (NCDrillParameters Object)..... 312
NeutralFileExport Object.....313
AddVariant Method (NeutralFileExport Object)..... 315
ClearVariants Method (NeutralFileExport Object)..... 316
Go Method (NeutralFileExport Object).....317
BoardOffsetX Property (NeutralFileExport Object).....318
BoardOffsetY Property (NeutralFileExport Object).....319
DesignFileName Property (NeutralFileExport Object).....320
DesignVariantNames Property (NeutralFileExport Object).....321
Errors Property (NeutralFileExport Object).....322
IncludeBoardAddedParts Property (NeutralFileExport Object).....323
IncludeBoardAttributeData Property (NeutralFileExport Object)..... 324
IncludeComponentsData Property (NeutralFileExport Object).....325
IncludeGeometriesData Property (NeutralFileExport Object).....326
IncludeHolesData Property (NeutralFileExport Object)..... 327
IncludeNetsData Property (NeutralFileExport Object).....328
IncludePanelAddedParts Property (NeutralFileExport Object).....329
IncludePadsData Property (NeutralFileExport Object)..... 330
IncludeTestpointsData Property (NeutralFileExport Object)..... 331
IncludeVariantData Property (NeutralFileExport Object)..... 332
MultiBoardDataOnPanel Property (NeutralFileExport Object).....333
SetupFileName Property (NeutralFileExport Object).....334
Units Property (NeutralFileExport Object)..... 335
ODBPPOutputEngine Object..... 336
AddVariant Method (ODBPPOutputEngine Object).....340
ClearAdvancedPackagingLayers Method (ODBPPOutputEngine Object)..... 341
ClearExportLayers Method (ODBPPOutputEngine Object)..... 342
ClearVariants Method (ODBPPOutputEngine Object).....343
Go Method (ODBPPOutputEngine Object)..... 344
AdvancedPackagingData Property (ODBPPOutputEngine Object)..... 345
AdvancedPackagingLayers Property (ODBPPOutputEngine Object)..... 346

AdvancedPackagingLayersToExport Property (ODBPPOutputEngine Object)..... 347
AppendToLogFile Property (ODBPPOutputEngine Object).....348
AssemblyLayers Property (ODBPPOutputEngine Object)..... 349
BoardPanelOutline Property (ODBPPOutputEngine Object)..... 350
CompressOutput Property (ODBPPOutputEngine Object)..... 351
DesignFileName Property (ODBPPOutputEngine Object)..... 352
DesignVariantNames Property (ODBPPOutputEngine Object)..... 353
DocumentationLayer Property (ODBPPOutputEngine Object).....354
DRCFeaturesLayer Property (ODBPPOutputEngine Object).....355
Errors Property (ODBPPOutputEngine Object).....356
ExportEmbeddedPassiveLayers Property (ODBPPOutputEngine Object).....357
ExportPartNumbers Property (ODBPPOutputEngine Object)..... 358
FabricationLayers Property (ODBPPOutputEngine Object)..... 359
GenerateSeparateVariantOutputs Property (ODBPPOutputEngine Object).....360
IgnoreComponentLayout Property (ODBPPOutputEngine Object)..... 361
IncludeAllVariantData Property (ODBPPOutputEngine Object)..... 362
LayersToExport Property (ODBPPOutputEngine Object).....363
LogFileDirectory Property (ODBPPOutputEngine Object)..... 364
NeutralizeNets Property (ODBPPOutputEngine Object)..... 365
NonFuncPinPadRemoval Property (ODBPPOutputEngine Object)..... 366
NonFuncViaPadRemoval Property (ODBPPOutputEngine Object)..... 367
NPINetTypes Property (ODBPPOutputEngine Object)..... 368
ODBPPExportMode Property (ODBPPOutputEngine Object).....369
ODBPPVersion Property (ODBPPOutputEngine Object)..... 370
OutputJobName Property (ODBPPOutputEngine Object)..... 371
OutputPath Property (ODBPPOutputEngine Object)..... 372
PackageOutlineLayer Property (ODBPPOutputEngine Object)..... 373
ReadDRCFeatures Property (ODBPPOutputEngine Object)..... 374
RemoveCadnetNetlist Property (ODBPPOutputEngine Object).....375
RemoveEDAData Property (ODBPPOutputEngine Object)..... 376
RoundCorners Property (ODBPPOutputEngine Object)..... 377
SetupFileName Property (ODBPPOutputEngine Object).....378
Units Property (ODBPPOutputEngine Object)..... 379
UseGeneratedSilkscreenData Property (ODBPPOutputEngine Object)..... 380
UserLayerEx Property (ODBPPOutputEngine Object).....381
PDFConductiveLayer Object.....382
ConductiveItems Property (PDFConductiveLayer Object)..... 383
PDFLayerOrder Object..... 384
Count Property (PDFLayerOrder Object)..... 385
Layers Property (PDFLayerOrder Object).....386
PDFOutput Object.....387
AddVariant Method (PDFOutput Object)..... 389
ClearVariants Method (PDFOutput Object)..... 390
Go Method (PDFOutput Object).....391
DesignFileName Property (PDFOutput Object).....392
DesignVariantNames Property (PDFOutput Object)..... 393
Errors Property (PDFOutput Object)..... 394
FillText Property (PDFOutput Object).....395
FitDrawing Property (PDFOutput Object).....396

IncludeVariantData Property (PDFOutput Object)....397
OutputFileName Property (PDFOutput Object)....398
SetupFileName Property (PDFOutput Object)....399
Sheets Property (PDFOutput Object)....400
PDFOutputSheet Object....401
BoardItems Property (PDFOutputSheet Object)....402
CellItems Property (PDFOutputSheet Object)....403
CellItemsLayer Property (PDFOutputSheet Object)....404
CellTypes Property (PDFOutputSheet Object)....405
ConductiveLayers Property (PDFOutputSheet Object)....406
HeaderText Property (PDFOutputSheet Object)....407
LayerOrder Property (PDFOutputSheet Object)....408
Mirror Property (PDFOutputSheet Object)....409
Name Property (PDFOutputSheet Object)....410
ProcessSheet Property (PDFOutputSheet Object)....411
TrailerText Property (PDFOutputSheet Object)....412
UserLayerOrder Property (PDFOutputSheet Object)....413
UserLayers Property (PDFOutputSheet Object)....414
SilkscreenGen Object....415
Go Method (SilkscreenGen Object)....417
BottomSideSilkscreenLayers Property (SilkscreenGen Object)....418
BottomSideUserLayers Property (SilkscreenGen Object)....419
BreakGraphicsPadClearance Property (SilkscreenGen Object)....420
BreakGraphicsViaClearance Property (SilkscreenGen Object)....421
BreakSilkscreenBy Property (SilkscreenGen Object)....422
BreakTextPadClearance Property (SilkscreenGen Object)....423
BreakTextViaClearance Property (SilkscreenGen Object)....424
DesignFileName Property (SilkscreenGen Object)....425
Errors Property (SilkscreenGen Object)....426
GraphicsLineWidth Property (SilkscreenGen Object)....427
Groups Property (SilkscreenGen Object)....428
ProcessAllGroups Property (SilkscreenGen Object)....429
SetupFileName Property (SilkscreenGen Object)....430
SilkScreenProcessSide Property (SilkscreenGen Object)....431
TextLineWidth Property (SilkscreenGen Object)....432
TopSideSilkscreenLayers Property (SilkscreenGen Object)....433
TopSideUserLayers Property (SilkscreenGen Object)....434
UseBreakGraphicsPadClearance Property (SilkscreenGen Object)....435
UseBreakGraphicsViaClearance Property (SilkscreenGen Object)....436
UseBreakTextPadClearance Property (SilkscreenGen Object)....437
UseBreakTextViaClearance Property (SilkscreenGen Object)....438
UserInformation Object....439
Address Property (UserInformation Object)....440
CompanyName Property (UserInformation Object)....441
EmailAddress Property (UserInformation Object)....442
FaxNumber Property (UserInformation Object)....443
FileName Property (UserInformation Object)....444
PhoneNumber Property (UserInformation Object)....445
UserName Property (UserInformation Object)....446

Chapter 3
Batch Engines Collections 447
Batch Engines Collection Descriptions 448
BoardItems Collection 449
Add Method (BoardItems Collection) 450
AddBySpan Method (BoardItems Collection) 451
AddDetailedView Method (BoardItems Collection) 452
AddFlexLayer Method (BoardItems Collection) 453
AddFlexPadLayer Method (BoardItems Collection) 454
CellItems Collection 455
Add Method (CellItems Collection) 456
CellTypes Collection 457
Add Method (CellTypes Collection) 458
ConductiveItems Collection 459
Add Method (ConductiveItems Collection) 460
DrillHoles Collection 461
Remove Method (DrillHoles Collection) 462
Count Property (DrillHoles Collection) 463
Item Property (DrillHoles Collection) 464
Errors Collection 465
Add Method (Errors Collection) 466
Remove Method (Errors Collection) 467
Count Property (Errors Collection) 468
Item Property (Errors Collection) 469
GerberOutputFiles Collection 470
Add Method (GerberOutputFiles Collection) 471
Remove Method (GerberOutputFiles Collection) 472
Count Property (GerberOutputFiles Collection) 473
Item Property (GerberOutputFiles Collection) 474
NCDrillChartColumns Collection 475
Add Method (NCDrillChartColumns Collection) 476
Remove Method (NCDrillChartColumns Collection) 477
Count Property (NCDrillChartColumns Collection) 478
Item Property (NCDrillChartColumns Collection) 479
PDFConductiveLayers Collection 480
Add Method (PDFConductiveLayers Collection) 481
Remove Method (PDFConductiveLayers Collection) 482
Count Property (PDFConductiveLayers Collection) 483
Item Property (PDFConductiveLayers Collection) 484
PDFOutputSheets Collection 485
Add Method (PDFOutputSheets Collection) 486
Remove Method (PDFOutputSheets Collection) 487
Count Property (PDFOutputSheets Collection) 488
Item Property (PDFOutputSheets Collection) 489
UserLayers Collection 490
Add Method (UserLayers Collection) 491

Chapter 4
Batch Engines Enumerated Types ... 493
Batch Engines Enumerated Type Descriptions ... 494
EEnginesAngleUnit Enum ... 497
EEnginesArcStyle Enum ... 498
EEnginesBoardItem Enum ... 499
EEnginesCellItem Enum ... 504
EEnginesCellOutlineType Enum ... 507
EEnginesCellSide Enum ... 508
EEnginesCellType Enum ... 509
EEnginesCharacterSet Enum ... 513
EEnginesColumnAlignment Enum ... 514
EEnginesConductiveItem Enum ... 515
EEnginesDataDelimiter Enum ... 518
EEnginesDataType Enum ... 519
EEnginesDrawingTask Enum ... 520
EEnginesDRCElementType Enum ... 521
EEnginesDRCFeaturesLayer Enum ... 526
EEnginesDRCMissingConductivePad Enum ... 527
EEnginesDRCMissingCoverLayerPad Enum ... 528
EEnginesDRCMissingSolderPad Enum ... 529
EEnginesDRCNetRule Enum ... 530
EEnginesDRCPadRule Enum ... 531
EEnginesDRCPadToPadCheck Enum ... 532
EEnginesDRCPlaneRule Enum ... 533
EEnginesDRCRules Enum ... 534
EEnginesDRCTraceRule Enum ... 535
EEnginesDRCViasUnder Enum ... 536
EEnginesDrillSymbol Enum ... 537
EEnginesDrillSymbolType Enum ... 539
EEnginesElementType Enum ... 540
EEnginesErrCode Enum ... 545
EEnginesErrorSeverity Enum ... 550
EEnginesExtPrintFileFormat Enum ... 551
EEnginesGeneralInterfaceIdentifyPartsType Enum ... 552
EEnginesGeneralInterfaceOutputType Enum ... 553
EEnginesGeneralInterfaceUnitsType Enum ... 554
EEnginesIDFExportComponentSideType Enum ... 555
EEnginesIDFExportPackageOutlineType Enum ... 556
EEnginesIDFExportSectionsType Enum ... 557
EEnginesIDFVersionType Enum ... 558
EEnginesMappedUserLayer Enum ... 559
EEnginesNCDrillChartUnits Enum ... 561
EEnginesNCDrillColumn Enum ... 562
EEnginesNCDrillDataType Enum ... 564
EEnginesNCDrillSweepAxis Enum ... 565
EEnginesNCDrillUnits Enum ... 566

EEnginesNonFuncPad Enum..... 567
EEnginesODBPPExportMode Enum..... 568
EEnginesOutputUnits Enum..... 569
EEnginesPackageOutlineLayer Enum..... 570
EEnginesPDFSheetPageSize Enum..... 571
EEnginesPlotColor Enum..... 573
EEnginesPolygonFill Enum..... 574
EEnginesSide Enum..... 575
EEnginesSilkScreenBreakType Enum..... 576
EEnginesSilkscreenDesignLayer Enum..... 577
EEnginesStepMode Enum..... 578
EEnginesUnit Enum..... 579
EEnginesZeroTruncation Enum..... 580

Chapter 5
Mask Engine..... 581
Mask Engine Overview..... 582
Point Arrays..... 582
How to Connect to the MEAC..... 583
How to Compute Placement Density and Board Area..... 583
Mask Engine Scripting Samples..... 585
Mask Engine Objects..... 586
Line Object..... 587
PointsArray Property (Line Object)..... 588
Mask Object..... 589
AbsorbHoles Method (Mask Object)..... 590
Area Property (Mask Object)..... 591
BooleanOp Method (Mask Object)..... 592
CheckExterior Method (Mask Object)..... 593
CheckInterior Method (Mask Object)..... 594
Clear Method (Mask Object)..... 595
ContainsPoint Method (Mask Object)..... 596
Delete Method (Mask Object)..... 597
Fracture Method (Mask Object)..... 598
Hatch Method (Mask Object)..... 599
IsEmpty Method (Mask Object)..... 600
Oversize Method (Mask Object)..... 601
Shapes Property (Mask Object)..... 602
Touch Method (Mask Object)..... 603
Vectorize Method (Mask Object)..... 604
VectorizeByChordAngle Method (Mask Object)..... 605
MaskEngine Object..... 606
BooleanOp Method (MaskEngine Object)..... 607
CurrentUnit Property (MaskEngine Object)..... 608
Masks Property (MaskEngine Object)..... 609
Resolution Property (MaskEngine Object)..... 610
Utility Property (MaskEngine Object)..... 611
Shape Object..... 612

PointsArray Property (Shape Object)....613
Solid Property (Shape Object)....614
Utility Object....615
CreateCircleXYR Method (Utility Object)....616
CreateRectXYR Method (Utility Object)....617
Mask Engine Collections....618
Lines Collection....619
Count Property (Lines Collection)....620
Item Property (Lines Collection)....621
Remove Method (Lines Collection)....622
Masks Collection....623
Add Method (Masks Collection)....624
AddByMask Method (Masks Collection)....625
Count Property (Masks Collection)....626
Delete Method (Masks Collection)....627
Item Property (Masks Collection)....628
Remove Method (Masks Collection)....629
Shapes Collection....630
AddByMask Method (Shapes Collection)....631
AddByPath Method (Shapes Collection)....632
AddByPointsArray Method (Shapes Collection)....633
AddByShape Method (Shapes Collection)....634
Count Property (Shapes Collection)....635
Delete Method (Shapes Collection)....636
Item Property (Shapes Collection)....637
Remove Method (Shapes Collection)....638
Mask Engine Enumerated Types....639
EMaskEngineBooleanOp Enum....640
EMaskEngineErrCode Enum....641
EMaskEngineHatchAngle Enum....642
EMaskEngineOversizeType Enum....643
EMaskEngineResolutionUnit Enum....644
EMaskEngineUnit Enum....645

Third-Party Information

Chapter 1
Batch Engines Overview

The Engines Automation layer is a Component Object Model (COM)-based layer that enables you to generate manufacturing outputs, perform DRC checks, and perform output validation in batch mode using automation languages like Visual Basic, VBScript, JScript or C++.

Engines Automation layer functionality includes the following:

- DFFDRC checking of pcb
- DRC checking of panel or pcb (files created from the layout and manufacturing tools)
- Drawing Wizard (for creating, moving, copying, and deleting drawing file directories)
- DXF Export
- Extended Print
- General Interface Output
- Generated Silkscreen Output
- Gerber Output
- IDF (Intermediate Data Format) Output
- Manufacturing Output Validation (MOV)
- NCDrill Output
- OASIS Export
- ODB++ Output
- PDF Output
- XE neutral file Output

Note:
Additional engines exist for Copper Balancing, DFFSliver Analysis, DXF Import, and IDF Import. These engines are documented in the *Layout Automation Reference (Part 2)*, *Layout Automation Reference (Part 3)*, and *Layout Automation Reference (Part 4)*.

- Copper Balancing

- DFF Sliver Analysis
- DXF Import
- IDF Import

Engines Automation Usage
Engine Error Access Through Automation

Engines Automation Usage

Automation supports a variety of engines.

The following engines are supported:

- DFFDRC on page 26
- DRC on page 34
- DrawingFileWizard on page 74
- DXFExport on page 87
- ExtendedPrint on page 104
- GeneralInterfaceEngine on page 122
- Gerber on page 144
- IDFExport on page 205
- IOASISExportEngine on page 225
- ManufacturingValidationOutput on page 231
- NCDrill on page 243
- NeutralFileExport on page 313
- ODBPPOutputEngine on page 336
- PDFOutput on page 387
- SilkscreenGen on page 415

The usage of these engines through Automation requires an Automation Pro license, with the exception of Drawing File Wizard.

Note:
The automation of the above engines operates only on saved data. If you have a design open, save the design before running your scripts.

The following example shows the automation calls needed to access various engines.

```
' To access the current engines, use the following calls:
option Explicit

' Create Drawing Wizard object for X-ENTP VX.1 (COM version 9)
Dim oDrawingWizard
Set oDrawingWizard = CreateObject("MGCPCBEngines.DrawingFileWizard.9")
Scripting.AddTypeLibrary("MGCPCBEngines.DrawingFileWizard")

' Create DRC Engine object
Dim oDRCEngine
Set oDRCEngine = CreateObject("MGCPCBEngines.DRC.9")
Scripting.AddTypeLibrary("MGCPCBEngines.DRC")

' Create DFFDRC Engine object
Dim oDFFDRCEngine
Set oDFFDRCEngine = CreateObject("MGCPCBEngines.DFFDRC.9")
Scripting.AddTypeLibrary("MGCPCBEngines.DFFDRC")

' Create DXFExport Engine object
Dim oDXFExportEngine
Set oDXFExportEngine = CreateObject("MGCPCBEngines.DXFExport.9")
Scripting.AddTypeLibrary("MGCPCBEngines.DXFExport")

' Create Extended Print Engine object
Dim oExtendedPrintEngine
Set oExtendedPrintEngine = CreateObject("MGCPCBEngines.ExtendedPrint.9")
Scripting.AddTypeLibrary("MGCPCBEngines.ExtendedPrint")

' Create General Interface Engine object
Dim oGeneralInterfaceEngine
Set oGeneralInterfaceEngine =
 CreateObject("MGCPCBEngines.GeneralInterfaceEngine.9")
Scripting.AddTypeLibrary("MGCPCBEngines.GeneralInterfaceEngine")

' Create Gerber Engine object
Dim oGerberEngine
Set oGerberEngine = CreateObject("MGCPCBEngines.Gerber.9")
Scripting.AddTypeLibrary("MGCPCBEngines.Gerber")

' Create IDF Export Engine object
Dim oIDFExportEngine
Set oIDFExportEngine = CreateObject("MGCPCBEngines.IDFExport.9")
Scripting.AddTypeLibrary("MGCPCBEngines.IDFExport")

' Create MOV Engine object
Dim oMOVEngine
```

```
Set oMOVEngine =
 CreateObject("MGCPCBEngines.ManufacturingOutputValidation.9")
Scripting.AddTypeLibrary("MGCPCBEngines.ManufacturingOutputValidation")

' Create NCDrill Engine object
Dim oNCDrillEngine
Set oNCDrillEngine = CreateObject("MGCPCBEngines.NCDrill.9")
Scripting.AddTypeLibrary("MGCPCBEngines.NCDrill")

' Create Neutral File Export Engine object
Dim oNeutralFileExportEngine
Set oNeutralFileExportEngine =
 CreateObject("MGCPCBEngines.NeutralFileExport.9")
Scripting.AddTypeLibrary("MGCPCBEngines.NeutralFileExport")

' Create ODB++ Output Engine object
Dim oODBPPOutputEngine
Set oODBPPOutputEngine = CreateObject("MGCPCBEngines.ODBPPOutputEngine.9")
Scripting.AddTypeLibrary("MGCPCBEngines.ODBPPOutputEngine")

' Create PDF Engine object
Dim oPDFEngine
Set oPDFEngine = CreateObject("MGCPCBEngines.PDFOutput.9")
Scripting.AddTypeLibrary("MGCPCBEngines.PDFOutput")

' Create Silkscreen Generator Engine object
Dim oSilkscreenGenEngine
Set oSilkscreenGenEngine = CreateObject("MGCPCBEngines.SilkscreenGen.9")
Scripting.AddTypeLibrary("MGCPCBEngines.SilkscreenGen")
```

Related Topics

Scripting With Multiple Installs [Common Automation Reference]

Engine Error Access Through Automation

Each of the engines output an error and warning log. The error logs are accessible through Automation.

The following code sample shows how to access the errors for the DRC Engine.

```
option Explicit
'************************************************************************
Sub RunDRC()
' Create DRC Engine object
Dim oDRCEngine
Set oDRCEngine = CreateObject("MGCPCBEngines.DRC.9")
Scripting.AddTypeLibrary("MGCPCBEngines.DRC")
' Set the design file name
oDRCEngine.DesignFileName = jobName
' Run the DRC Engine
oDRCEngine.Go
' Check errors
```

```
Dim oErrors, oErrCnt, oErr
Set oErrors = oDRCEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

Chapter 2
Batch Engines Objects

The Automation Pro Batch Engines objects enable you to set up and run batch export and output applications.

Batch Engines Object Descriptions

Batch Engines Object Descriptions

This section lists and describes the available Automation Pro batch engines objects, including information about the methods and properties available to each object.

Table 1. Batch Engines Objects

Object	Description
DFFDRC Object	For more information refer to Batch DFF dialog box by choosing **Analysis > Batch DFF**.
DRC Object	For more information refer to the Batch DRC dialog box by choosing **Analysis > Batch DRC**.
DrillHole Object	For more information, refer to the NC Drill Generation dialog box, by choosing **Output > NC Drill**
DrawingFileWizard Object	For more information on the Drawing Wizard see the the *Drawing Editor Guide*.
DXFExport Object	For more information, refer to the DXF Export dialog box, by choosing **File > Export > DXF**
Error Object	Used by all batch engines to return run time errors.
ExtendedPrint Object	For more information, refer to the Extended Print dialog box, by choosing **File > Printing > Extended Print**.
FCF Object	Refer to the Drill Chart Options tab and the FCF tab of the NC Drill Generation dialog box.
GeneralInterfaceEngine Object	Refer to the General Interfaces dialog box by choosing **File > Export > General Interfaces**
Gerber Object	Refer to the Gerber Output dialog box by choosing **Output > Gerber**
GerberMachineFormat Object	Refer to the Gerber Machine Format dialog box, by choosing **Setup > Gerber Machine Format**.
GerberOutputFile Object	For more information, refer to the Gerber Output dialog box, by choosing **Output > Gerber**.
IDFExport Object	For more information, refer to the Export IDF dialog box by choosing **File > Export > IDF**.
IOASISExportEngine Object	For more information, refer to the OASIS Export dialog box by choosing **File > Export > OASIS**.
ManufacturingOutputValidation Object	For more information, refer to the Manufacturing Output Validation dialog box by choosing **Output > Manufacturing Output Validation**.

Table 1. Batch Engines Objects (continued)

Object	Description
NCDrill Object	For more information, refer to the NC Drill Generation dialog box, by choosing **Output > NC Drill**
NCDrillChart Object	Refer to the Drill Chart Options tab of the NC Drill Generation dialog box.
NCDrillChartColumn Object	Refer to the Drill Chart Options tab of the NC Drill Generation dialog box.
NCDrillMachineFormat Object	The NCDrill machine format definitions.
NCDrillParameters Object	For more information, refer to the Drill Options tab of the NC Drill Generation dialog box by choosing **Output > NC Drill**
NeutralFileExport Object	For more information, refer to the Create XE Neutral File dialog box by choosing **Output > Neutral File**.
ODBPPOutputEngine Object	For more information, refer to the ODB++ Output dialog box by choosing **Output > ODB++**
PDFConductiveLayer Object	Used by the PDFOutputSheet object to define conductive layers.
PDFLayerOrder Object	Used by the ExtendedPrint object to define the layer order.
PDFOutput Object	Used by the ExtendedPrint object to define the PDF output.
PDFOutputSheet Object	Used by the ExtendedPrint object to define the PDF sheets.
SilkscreenGen Object	For more information refer to the Silkscreen Generator dialog box by choosing **Output > Silkscreen Generator**
UserInformation Object	Used by the ManufacturingOutputValidation object to return the user information.

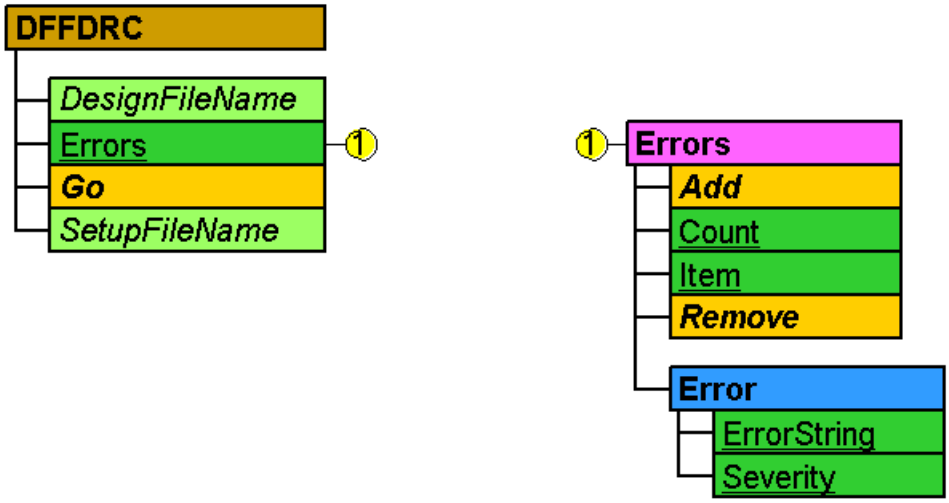
DFFDRC Object

The DFFDRC Object contains information that the DFFDRC Engine uses to run batch design for fabrication rules checking..

For more information on Batch DFF checks, refer to the Batch DFF dialog box by choosing **Analysis > Batch DFF**.

The following figure shows the DFFDRC Object Data Model.

Figure 1. DFFDRC Object Data Model



The following example shows a script sample that runs batch DFF DRC. The design file, “jobName” is specified in the main script. A VBScript dialog box “rtn” queries the user to use either the default setup file “(Final DFF).dfm” or to enter the full path to the desired setup file. The DFFDRC engine is then run using the setup file on the design.

Find the complete script at the following location:

SDD_HOME\standard\examples\pcb\Automation\AutoProEngines\RunDFFDRC.vbs

Example 1. Batch DFFDRC Example Script

```
Option Explicit

Dim args, argsCnt, app, docObj, jobName, jobConfigPath, jobDesignPath

On Error Resume Next
Set args = ScriptHelper.Arguments
argsCnt = args.Count
If argsCnt = 3 Then
    ' Called from command line with job name argument
    ' mgcscript RunDFFDRC.vbs [job name]
```

```
    jobName = args.item(3)
Else
    ' Called from the application, drag-and-drop,
    ' or command line without job name argument
    Set app = GetObject(, "MGCPCB.ExpeditionPCBApplication")
    Set docObj = GetLicensedDoc(app)
    jobName = docObj.FullName
End If
On Error Goto 0

Scripting.AddTypeLibrary("MGCPCBEngines.DFFDRC")

Call RunDFFDRC()

MsgBox "fini", VBInformation, "RunDFFDRC.vbs"

'*********************************************************************
Sub RunDFFDRC()
    'On Error Resume Next

    ' Create DRC Engine object
    Dim oDFFDRCEngine
    Set oDFFDRCEngine = CreateObject("MGCPCBEngines.DFFDRC")

 if Err Then
    Exit Sub
 End If

    ' Set the design file name
    oDFFDRCEngine.DesignFileName = jobName

    ' Get design Config directory
    Dim pcbPath, n, i
    jobName = Replace(jobName, "/", "\")
    pcbPath = Split(jobName, "\")
    n = UBound(pcbPath)
    jobDesignPath = ""
    For i = 0 to n - 1
        jobDesignPath = jobDesignPath + pcbPath(i) + "\"
    Next
    jobConfigPath = jobDesignPath + "Config\"

    Dim setupfile
    Dim rtn
    rtn = MsgBox("Using default setup file((Final DFF).dfm)?", VBYesNo +
 VBQuestion, "RunDFFDRC.vbs")
    If rtn = VBYes Then
        setupfile = jobConfigPath  + "(Final DFF).dfm"
        oDFFDRCEngine.SetupFileName = setupfile
    Else
        ' Ask for the full path to the user-defined setup file
        setupfile = InputBox("Enter full path to the setup file?",
 "RunDFFDRC.vbs")
        oDFFDRCEngine.SetupFileName = setupfile
```

```
    End If

    ' Run the DRC Engine
    oDFFDRCEngine.Go
    msgbox setupfile

    ' Check errors
    Dim oErrors
    Set oErrors = oDFFDRCEngine.Errors
End Sub

Public Function GetLicensedDoc(appObj)
    On Error Resume Next
    Dim key, licenseServer, licenseToken, docObj
    Set GetLicensedDoc = Nothing
    ' collect the active document
    Set docObj = appObj.ActiveDocument
    If Err Then
        Call appObj.Gui.StatusBarText("No active document: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' Ask Xpedition Layout’s document for the key
    key = docObj.Validate(0)
    ' Get token from license server
    Set licenseServer =
 CreateObject("MGCPCBAutomationLicensing.Application")
    licenseToken = licenseServer.GetToken(key)
    Set licenseServer = Nothing
    ' Ask the document to validate the license token
    Err.Clear
    Call docObj.Validate(licenseToken)
    If Err Then
        Call appObj.Gui.StatusBarText("No active document license: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' everything is OK, return document
    Set GetLicensedDoc = docObj
End Function
```

Table 2. DFFDRC Object Methods and Properties

Method or Property	Description
Go Method (DFFDRC Object)	Checks the design for fabrication rules and runs the DFFDRC engine.
DesignFileName Property (DFFDRC Object)	Sets the design file name, including path.
Errors Property (DFFDRC Object)	Returns the collection of errors.
SetupFileName Property (DFFDRC Object)	Sets the name of the batch DFF setup file, including path. This is the .dfm Scheme file that the DFFDRC engine

Table 2. DFFDRC Object Methods and Properties (continued)

Method or Property	Description
	uses to process output data and to load the Batch DFF dialog box. The Scheme option in the Batch DFF dialog box specifies this file.

Go Method (DFFDRC Object)

Object: DFFDRC Object

Prerequisites: None

Checks the design for fabrication rules and runs the DFFDRC engine.

Usage

DFFDRC.Go ()

Arguments

None

Examples

```
oDFFDRCEngine.Go
```

DesignFileName Property (DFFDRC Object)

Object: DFFDRC Object

Access: Write-Only

Prerequisites: None

Sets the design file name, including path.

Usage

DFFDRC.DesignFileName **=** String

Arguments

None

Return Values

String. A string that contains the name of the design, including path.

Examples

```
oDFFDRCEngine.DesignFileName = jobName
```

Errors Property (DFFDRC Object)

Object: DFFDRC Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

DFFDRC.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oDFFDRCEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

SetupFileName Property (DFFDRC Object)

Object: DFFDRC Object

Access: Write-Only

Prerequisites: None

Sets the name of the batch DFF setup file, including path. This is the *.dfm* Scheme file that the DFFDRC engine uses to process output data and to load the Batch DFF dialog box. The Scheme option in the Batch DFF dialog box specifies this file.

Usage

DFFDRC.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the name of the batch DFF setup file, including path.

Examples

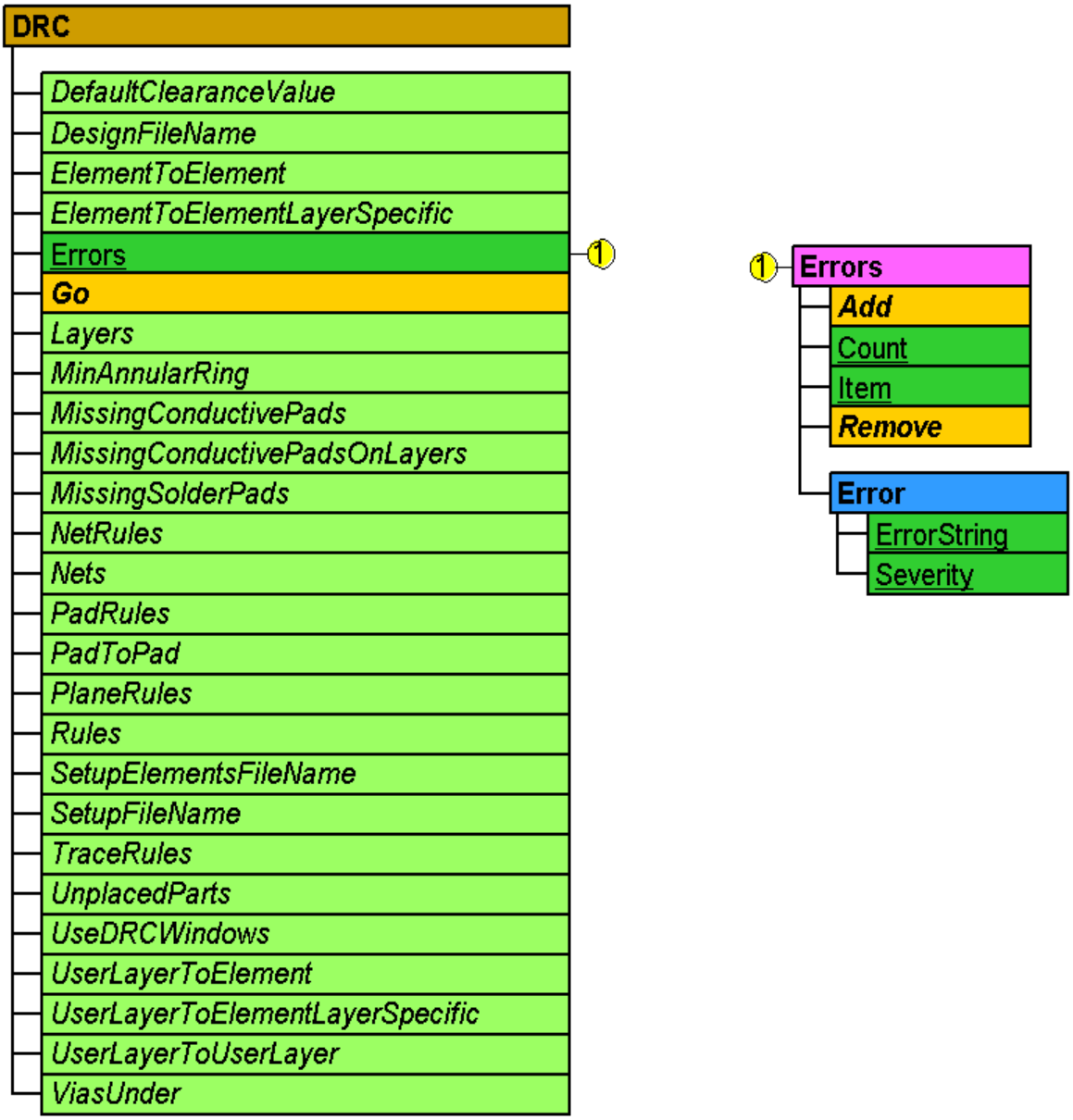
```
oDFFDRCEngine.SetupFileName = “mydesing/pcb/config/(FinalDFF).dfm”
```

DRC Object

The DRC Object contains the information the DRC Engine uses to run Batch DRC.

For more information, refer to the Batch DRC dialog box by choosing **Analysis > Batch DRC**.

Figure 2. DRC Data Model



The following example shows a script sample that runs batch DRC.

You can find the complete script at: *SDD_HOME\standard\examples\pcb\Automation\AutoProEngines \RunDRC.vbs*

Example 2. Batch DRC Example Script

```
Option Explicit

Dim args, app, docObj, jobName, jobConfigPath
Set args = ScriptHelper.Arguments
If args.Count = 3 Then
    ' Call from command line
    ' mgcscript RunDRC.vbs [job name]
    jobName = args.item(3)
Else
    ' Call from Xpedition Layout
    Set app = GetObject(, "MGCPCB.ExpeditionPCBApplication")
    Set docObj = GetLicensedDoc(app)
    jobName = docObj.FullName
End If

Scripting.AddTypeLibrary("MGCPCBEngines.DRC")

Call RunDRC()

MsgBox "fini", VBInformation, "RunDRC.vbs"

'*************************************************************************
Sub RunDRC()
    'On Error Resume Next

    ' Create DRC Engine object
    Dim oDRCEngine
    Set oDRCEngine = CreateObject("MGCPCBEngines.DRC")

    ' Set the design file name
    oDRCEngine.DesignFileName = jobName

    ' Get design Config directory
    Dim pcbPath
    pcbPath = Split(jobName, "PCB")
    jobConfigPath = pcbPath(0) + "PCB/Config/"

    Dim rtn
    rtn = MsgBox("Using existing setup files?", VBYesNo + VBQuestion,
 "RunDRC.vbs")

    If rtn = VBYes Then
        oDRCEngine.SetupFileName = jobConfigPath  + "(Final DRC).drc"
        oDRCEngine.SetupElementsFileName = jobConfigPath + "(Final
 DRC).e2e"
    Else
        Call SetupDRCSettings(oDRCEngine)
        Call SetupElementToElement(oDRCEngine)
        Call SetupSpecialRules(oDRCEngine)
    End If
```

```
    ' Run the DRC Engine
    oDRCEngine.Go

    ' Check errors
    Dim oErrors
    Set oErrors = oDRCEngine.Errors
End Sub

Sub SetupDRCSettings(oDRCEngine)
    'On Error Resume Next

    ' Same as the options in "DRC Settings" tab
    ' of the "Batch DRC" dialog, except the
    ' "Advanced Element to Element Rules"
    Dim netnames(0)
    netnames(0) = ""

    Dim lyrNums(0)
    lyrNums(0) = 0

    oDRCEngine.UseDRCWindows = FALSE
    oDRCEngine.Nets = netnames
    oDRCEngine.Rules =  eengDRCRulesAll
    oDRCEngine.PadToPad = eengDRCPadToPadAll
    oDRCEngine.Layers = lyrNums
End Sub

Sub SetupElementToElement(oDRCEngine)
  '  On Error Resume Next

     oDRCEngine.ElementToElement(eengDRCElementPartHoles,
 eengDRCElementPartHoles, eengUnitMils) = 4


     oDrcEngine.ElementToElementLayerSpecific(eengDRCElementAssemblyRefDes,
 "Bottom", eengDRCElementAssemblyPartNumbers, "Bottom", eengUnitMils) = 5

    oDrcEngine.UserLayerToElementLayerSpecific("User Layer 53",
 eengDRCElementViaPads, 0, eengUnitMils) = 10
    oDrcEngine.UserLayerToElementLayerSpecific("User Layer 53",
 eengDRCElementViaObstructs, 0, eengUnitMils) = 3
    oDrcEngine.UserLayerToUserLayer("User Layer 53", "User Layer 53",
 eengUnitMils) = 1

End Sub

Sub SetupSpecialRules(oDRCEngine)
    'On Error Resume Next

    oDRCEngine.TraceRules = eengDRCTraceRuleAll
    oDRCEngine.NetRules =  eengDRCNetRuleAll
    oDRCEngine.PlaneRules =  eengDRCPlaneRuleAll
    oDRCEngine.UnplacedParts = TRUE
```

```
    oDRCEngine.PadRules = eengDRCPadRuleAll
    oDRCEngine.ViasUnder = eengDRCViasUnderAll
    oDRCEngine.MissingConductivePads = eengDRCMissingCondPadAllLayers
    oDRCEngine.MissingSolderPads = eengDRCMissingSolderPadAll
End Sub

Public Function GetLicensedDoc(appObj)
    On Error Resume Next
    Dim key, licenseServer, licenseToken, docObj
    Set GetLicensedDoc = Nothing
    ' collect the active document
    Set docObj = appObj.ActiveDocument
    If Err Then
        Call appObj.Gui.StatusBarText("No active document: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' Ask Xpedition Layout's document for the key
    key = docObj.Validate(0)
    ' Get token from license server
    Set licenseServer =
 CreateObject("MGCPCBAutomationLicensing.Application")
    licenseToken = licenseServer.GetToken(key)
    Set licenseServer = Nothing
    ' Ask the document to validate the license token
    Err.Clear
    Call docObj.Validate(licenseToken)
    If Err Then
        Call appObj.Gui.StatusBarText("No active document license: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' everything is OK, return document
    Set GetLicensedDoc = docObj
End Function
```

Table 3. DRC Object Methods and Properties

Method or Property	Description
Go Method (DRC Object)	Checks the design rules and runs the DRC engine on the design.
DefaultClearanceValue Property (DRC Object)	Sets the clearance value against all other default elements, including itself.
DesignFileName Property (DRC Object)	Sets the name of the design file, including path.
ElementToElement Property (DRC Object)	Sets the clearance value between two non-layer specific elements. Refer to the Element to Element dialog box available from the Batch DRC dialog.
ElementToElementLayerSpecific Property (DRC Object)	Sets the clearance value for layer-specific elements. Refer to the Element to Element dialog box available from the Batch DRC dialog.

Table 3. DRC Object Methods and Properties (continued)

Method or Property	Description
Errors Property (DRC Object)	Returns the collection of errors. Use after the Go method has been run for the DRC object.
Layers Property (DRC Object)	Sets the layers to be included for proximity check. If 0 is specified in the first array entry, all layers are checked. Refer to the Batch DRC dialog box, DRC Settings tab, Proximity options.
MinAnnularRing Property (DRC Object)	Sets the minimum radial dimension value for Connectivity and Special Rules. Use this property to check for a minimum metal thickness between the hole and the shortest distance to the pad edge. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Minimum annular ring value.
MissingConductivePads Property (DRC Object)	Sets the check to make sure that all pins that go through a layer have a pad. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Missing pads, Conductive option.
MissingConductivePadsOnLayers Property (DRC Object)	Sets the check for missing pads on a pin based on selected layers. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Missing pads, Conductive, Only pads on selected layers option.
MissingCoverLayerPads Property (DRC Object)	Sets the check for missing part cover layer pads and/or via cover layer pads. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Missing pads, Other options.
MissingSolderPads Property (DRC Object)	Sets the check for missing soldermask pads and/or solderpaste pads. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Missing pads, Other options.
NetRules Property (DRC Object)	Sets the net rule options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Nets options.
Nets Property (DRC Object)	Sets the nets to check. If an empty string is specified in the first array entry, all nets are checked. Refer to the Batch DRC dialog box, DRC Settings tab, Check selected nets only option.
PadRules Property (DRC Object)	Sets the holes/pads/vias options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Holes/Pads/Vias options.

Table 3. DRC Object Methods and Properties (continued)

Method or Property	Description
PadToPad Property (DRC Object)	Sets the pad to pad check options. Refer to the Batch DRC dialog box, DRC Settings tab, Pad to pad checks.
PlaneRules Property (DRC Object)	Sets the planes options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Planes option.
Rules Property (DRC Object)	Sets the rules to check. Refer to the Batch DRC dialog box, DRC Settings tab, Rules to check options.
SetupElementsFileName Property (DRC Object)	Sets the name of the DRC elements to elements setup file, including path.
SetupFileName Property (DRC Object)	Sets the name of the DRC setup file, including path. This is the .drc Scheme file that the DRC engine uses to process output data and to load the Batch DRC dialog box. Refer to the Scheme option in the Batch DRC dialog box for more information.
TraceRules Property (DRC Object)	Sets the traces options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Traces options.
UnplacedParts Property (DRC Object)	Sets the check for all the parts that are not placed on the board. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Parts options.
UseDRCWindows Property (DRC Object)	Sets the check to use DRC windows for proximity check. If No DRC Windows are found, the whole design is checked. Refer to the Batch DRC dialog box, DRC Settings tab, Proximity area option.
UserLayerToElement Property (DRC Object)	Sets the user layer to element clearance value. Refer to the Element to Element dialog box available from the Batch DRC dialog.
UserLayerToElementLayerSpecific Property (DRC Object)	Sets the user layer to layer-specific element clearance value. Refer to the Element to Element dialog box available from the Batch DRC dialog.
UserLayerToUserLayer Property (DRC Object)	Sets the user layer to user layer clearance value. Refer to the Element to Element dialog box available from the Batch DRC dialog.
ViasUnder Property (DRC Object)	Sets the vias under options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Vias under options.

Go Method (DRC Object)

Object: DRC Object

Prerequisites: None

Checks the design rules and runs the DRC engine on the design.

Usage

DRC.Go

Arguments

None

Examples

```
oDRCEngine.Go
```

DefaultClearanceValue Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the clearance value against all other default elements, including itself.

Usage

```
DRC.DefaultClearanceValue (ByVal is As EEnginesDRCElementType,
        [ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double
```

Arguments

- eElement

  EEnginesDRCElementType Enum
- eUnit

  EEnginesUnit Enum (Optional) If nothing is specified, the current unit is used.

Return Values

Double. A double that contains the clearance value.

Examples

```
oDRCEngine.DefaultClearanceValue(eengDRCElementViaPads) = 1.0
```

DesignFileName Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

DRC.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the name of the design. (*.pcb* or *.pnl*)

Examples

```
oDRCEngine.DesignFileName = jobName
```

ElementToElement Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the clearance value between two non-layer specific elements. Refer to the Element to Element dialog box available from the Batch DRC dialog.

Usage

```
DRC.ElementToElement (ByVal eElement1 As EEnginesDRCElementType,
        ByVal eElement2 As EEnginesDRCElementType,
        [ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double
```

Arguments

- eElement1

  Specifies first element.(EEnginesDRCElementType Enum)
- eElement2

  Specifies second element. (EEnginesDRCElementType Enum)
- eUnit

  EEnginesUnit Enum (Optional) If nothing is specified then the current unit is used.

Return Values

Double. A double that contains the clearance value between the two non-layer specific elements.

Examples

```
oDRCEngine.ElementToElement(eengDRCElementPartHoles,
 eengDRCElementPartHoles, eengUnitMils) = 4.0
```

ElementToElementLayerSpecific Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the clearance value for layer-specific elements. Refer to the Element to Element dialog box available from the Batch DRC dialog.

Usage

```
DRC.ElementToElementLayerSpecific
        (ByVal eElement1 As EEnginesDRCElementType, ByVal layer1 As Variant,
        ByVal eElement2 As EEnginesDRCElementType,
        ByVal layer2 As Variant,
        [ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double
```

Arguments

- eElement1

  Specifies first DRC element (EEnginesDRCElementType Enum)
- layer1

  Specifies the layer of eElement1
- eElement2

  Specifies second DRC element(EEnginesDRCElementType Enum)
- layer2

  Specifies the layer of eElement2
- eUnit

  EEnginesUnit Enum (Optional) If nothing is specified then the current unit is used.

Return Values

Double. A double that contains the clearance value for layer specific elements.

Examples

```
oDrcEngine.ElementToElementLayerSpecific(eengDRCElementAssemblyRefDes,
 "Bottom", eengDRCElementAssemblyPartNumbers, "Bottom", eengUnitMils) = 5.0
```

Errors Property (DRC Object)

Object: DRC Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors. Use after the Go method has been run for the DRC object.

Usage

DRC.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oDRCEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

Layers Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the layers to be included for proximity check. If 0 is specified in the first array entry, all layers are checked. Refer to the Batch DRC dialog box, DRC Settings tab, Proximity options.

Usage

DRC.Layers = Value

Arguments

None

Return Values

Value. An array specifying which layers need to be included for the proximity check.

Examples

```
oDRCEngine.Layers = lyrNums
```

MinAnnularRing Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the minimum radial dimension value for Connectivity and Special Rules. Use this property to check for a minimum metal thickness between the hole and the shortest distance to the pad edge. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Minimum annular ring value.

Usage

DRC.MinAnnularRing ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  EEnginesUnit Enum. (Optional) If nothing is specified then the current unit is used.

Return Values

Double. A double that contains the minimum radial dimension parameter.

Examples

```
oDRCEngine.MinAnnularRing = 10.0
```

MissingConductivePads Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the check to make sure that all pins that go through a layer have a pad. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Missing pads, Conductive option.

Usage

DRC.MissingConductivePads = EEnginesDRCMissingConductivePad

Arguments

None

Return Values

Value. A EEnginesDRCFeaturesLayer Enum value that specifies the type of check for the pins.

Examples

```
oDRCEngine.MissingConductivePads = eengDRCMissingCondPadAllLayers
```

MissingConductivePadsOnLayers Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the check for missing pads on a pin based on selected layers. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Missing pads, Conductive, Only pads on selected layers option.

Usage

DRC.MissingConductivePadsOnLayers = Value

Arguments

None

Return Values

Value. Sets the layers to check.

Examples

```
oDRCEngine.MissingConductivePadsOnLayers = 1
```

MissingCoverLayerPads Property (DRC Object)

Object:

Access: Write-Only

Prerequisites: None

Sets the check for missing part cover layer pads and/or via cover layer pads. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Missing pads, Other options.

Usage

```
DRC.MissingCoverLayerPads ([ByVal eDRCMissingCoverLayerPadsAs
        EEnginesDRCMissingCoverLayerPad = eengDRCMissingCoverLayerPadAll]) = True | False
```

Arguments

- eDRCMissingCoverLayerPad

  The missing cover layer pad option to set or unset. (EEnginesDRCMissingCoverLayerPad Enum)

Return Values

Boolean. Set the specified cover layer pad options to True or False.

Examples

```
oDRCEngine.MissingCoverLayerPads = eengDRCMissingCoverLayerPadsAll
```

MissingSolderPads Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the check for missing soldermask pads and/or solderpaste pads. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Missing pads, Other options.

Usage

DRC.MissingSolderPads ([ByVal eDRCMissingSolderPad As EEnginesDRCMissingSolderPad = eengDRCMissingSolderPadAll]) = True | False

Arguments

- eDRCMissingSolderPad

  The missing solderpad option to set or unset. (EEnginesDRCMissingSolderPad Enum)

Return Values

Boolean. Set the specified soldermask and solderpaste options to True or False.

Examples

```
oDRCEngine.MissingSolderPads = eengDRCMissingSolderPadAll
```

NetRules Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the net rule options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Nets options.

Usage

DRC.NetRules ([ByVal eDRCNet As EEnginesDRCNetRule = eengDRCNetRuleAll]) = True | False

Arguments

- eDRCNet

  The net rule option to set or unset. (EEnginesDRCNetRule Enum)

Return Values

Boolean. Set the specified net rules to True or False.

Examples

```
oDRCEngine.NetRules =  eengDRCNetRuleAll
```

Nets Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the nets to check. If an empty string is specified in the first array entry, all nets are checked. Refer to the Batch DRC dialog box, DRC Settings tab, Check selected nets only option.

Usage

DRC.Nets = Value

Arguments

None

Return Values

Value. Net name or empty string (all nets).

Examples

```
oDRCEngine.Nets = netnames
```

PadRules Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the holes/pads/vias options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Holes/Pads/Vias options.

Usage

DRC.PadRules ([ByVal eDRCPad As EEnginesDRCPadRule = eengDRCPadRuleAll]) = True | False

Arguments

- eDRCPad

  The pad rule option to set/unset. (EEnginesDRCPadRule Enum)

Return Values

Boolean. Set the specified pads/vias/holes options to True or False.

Examples

```
oDRCEngine.PadRules = eengDRCPadRuleAll
```

PadToPad Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the pad to pad check options. Refer to the Batch DRC dialog box, DRC Settings tab, Pad to pad checks.

Usage

```
DRC.PadToPad ([ByVal
        ePadCheck As EEnginesDRCPadToPadCheck = eengDRCPadToPadAll]) = True | False
```

Arguments

- ePadCheck

  The pad to pad check options to set/unset. EEnginesDRCPadToPadCheck Enum)

Return Values

Boolean. Set to True to turn on option. Set to False to turn off option.

Examples

```
oDRCEngine.PadToPad = eengDRCPadToPadAll
```

PlaneRules Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the planes options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Planes option.

Usage

DRC.PlaneRules ([ByVal eDRCPlane As EEnginesDRCPlaneRule = eengDRCPlaneRuleAll]) = True | False

Arguments

- eDRCPlane

  The planes rules options to set/unset. (EEnginesDRCPlaneRule Enum)

Return Values

Boolean. True sets the specified plane rules options on. False sets the specified plane rules options off.

Examples

```
oDRCEngine.PlaneRules =  eengDRCPlaneRuleAll
```

Rules Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the rules to check. Refer to the Batch DRC dialog box, DRC Settings tab, Rules to check options.

Usage

DRC.Rules ([ByVal eDRCRules As EEnginesDRCRules = eengDRCRulesAll]) = True | False

Arguments

- eDRCRules

  The DRC rules options to set or unset. (EEnginesDRCRules Enum)

Return Values

Boolean. If True, the specified DRC rules are on. If False, the specified DRC rules are off.

Examples

```
oDRCEngine.Rules =  eengDRCRulesAll
```

SetupElementsFileName Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the name of the DRC elements to elements setup file, including path.

Usage

DRC.SetupElementsFileName = String

Arguments

None

Return Values

String. A string that contains the name of the DRC elements to elements setup file to load.

Examples

```
oDRCEngine.SetupElementsFileName = jobConfigPath + "(Final DRC).e2e"
```

SetupFileName Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the name of the DRC setup file, including path. This is the *.drc* Scheme file that the DRC engine uses to process output data and to load the Batch DRC dialog box. Refer to the Scheme option in the Batch DRC dialog box for more information.

Usage

DRC.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the name of the DRC setup file including path.

Examples

```
oDRCEngine.SetupFileName = jobConfigPath  + "(Final DRC).drc"
```

TraceRules Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the traces options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Traces options.

Usage

DRC.TraceRules ([ByVal eDRCTrace As EEnginesDRCTraceRule = eengDRCTraceRuleAll]) = True | False

Arguments

- eDRCTrace

  The trace rules options to set/unset. (EEnginesDRCTraceRule Enum)

Return Values

Boolean. True sets the specified rules to on. False sets the specified rules to off.

Examples

```
oDRCEngine.TraceRules = eengDRCTraceRuleAll
```

UnplacedParts Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the check for all the parts that are not placed on the board. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Parts options.

Usage

DRC.UnplacedParts = True | False

Arguments

None

Return Values

Boolean. True sets the check for unplaced parts. False unsets the check for unplaced parts.

Examples

```
oDRCEngine.UnplacedParts = TRUE
```

UseDRCWindows Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the check to use DRC windows for proximity check. If No DRC Windows are found, the whole design is checked. Refer to the Batch DRC dialog box, DRC Settings tab, Proximity area option.

Usage

DRC.UseDRCWindows = True | False

Arguments

None

Return Values

Boolean. True checks the area specified by the DRC Window. False checks the entire design.

Examples

```
oDRCEngine.UseDRCWindows = FALSE
```

UserLayerToElement Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the user layer to element clearance value. Refer to the Element to Element dialog box available from the Batch DRC dialog.

Usage

```
DRC.UserLayerToElement (ByVal sUserLayer As String,
        ByVal eElement As EEnginesDRCElementType,
        [ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Clearance
```

Arguments

- sUserLayer

  User layer.
- eElement

  EEnginesDRCElementType Enum
- eUnit

  EEnginesUnit Enum. (Optional) This specifies what units to use for the clearance value. If nothing is specified then the current unit is used.

Return Values

Clearance. A double that specifies the clearance to use in the layer to element check.

Examples

```
oDrcEngine.UserLayerToElement(“User Layer 53“, eengDRCElementViaPads) =
 10.0
```

UserLayerToElementLayerSpecific Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the user layer to layer-specific element clearance value. Refer to the Element to Element dialog box available from the Batch DRC dialog.

Usage

```
DRC.UserLayerToElementLayerSpecific (ByVal sUserLayer As String,
        ByVal eElement As EEnginesDRCElementType,
        ByVal layer As Variant,
        [ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Clearance
```

Arguments

- sUserLayer

  User layer
- eElement

  EEnginesDRCElementType Enum
- layer

  Element layer
- eUnit

  EEnginesUnit Enum. (Optional) This specifies what units to use for the clearance value. If nothing is specified, the current unit is used.

Return Values

Clearance. A double that specifies the clearance value to use in the check.

Examples

```
oDrcEngine.UserLayerToElementLayerSpecific("User Layer 53",
 eengDRCElementViaPads, 0, eengUnitMils) = 10.0
```

UserLayerToUserLayer Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the user layer to user layer clearance value. Refer to the Element to Element dialog box available from the Batch DRC dialog.

Usage

```
DRC.UserLayerToUserLayer
        (ByVal sUserLayer1 As String,
        ByVal sUserLayer2 As String,
        [ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Clearance
```

Arguments

- sUserLayer1

  User layer to check against sUserLayer2
- sUserLayer2

  User layer to check against sUserLayer1
- eUnit

  EEnginesUnit Enum. (Optional) This specifies what units to use for the clearance value. If nothing is specified then the current unit is used.

Return Values

Clearance. A double that specifies the clearance value to use in the check

Examples

```
oDrcEngine.UserLayerToUserLayer("User Layer 53", "User Layer 53",
 eengUnitMils) = 1.0
```

ViasUnder Property (DRC Object)

Object: DRC Object

Access: Write-Only

Prerequisites: None

Sets the vias under options for Connectivity and Special Rules. This property requires that the Rules property be set to true for the enumerator eengDRCConnectAndSpecialRules. Refer to the Batch DRC dialog box, Connectivity and Special Rules tab, Vias under options.

Usage

DRC.ViasUnder ([ByVal eDRCViaUnder As EEnginesDRCViasUnder = eengDRCViasUnderAll]) = True | False

Arguments

- eDRCViaUnder

  The vias under options to set/unset. (EEnginesDRCViasUnder Enum)

Return Values

Boolean. If True, set the via under option. If False, unset the via under option.

Examples

```
oDRCEngine.ViasUnder = eengDRCViasUnderAll
```

DrillHole Object

The DrillHole object represents a drilled location in a board.

The following section describes the methods and properties available in the DrilledHole object.

Table 4. DrillHole Object Properties

Property	Description
CombineCharAndSize Property (DrillHole Object)	Sets the option to combine the drill character symbol and hole size.
DrillSymbol Property (DrillHole Object)	Sets or returns the drill symbol. The value can be a character, a CustomDrillSymbol name, or an EEnginesDrillSymbol.
DrillSymbolSize Property (DrillHole Object)	Sets or returns the drill symbol size.
DrillSymbolType Property (DrillHole Object)	Sets or returns the drill symbol type.
FCF Property (DrillHole Object)	Returns the feature control frame that specifies the precise geometric tolerance information for this hole.
Name Property (DrillHole Object)	Returns the name of the drill hole.

CombineCharAndSize Property (DrillHole Object)

Object: DrillHole Object

Access: Write-Only

Prerequisites: None

Sets the option to combine the drill character symbol and hole size.

Usage

```
DrillHole.CombineCharAndSize = True | False
```

Arguments

None

Return Values

True | False. If True, the drill character symbol and hole size are combined. If False, the drill character symbol and hole size are not combined.

DrillSymbol Property (DrillHole Object)

Object: DrillHole Object

Access: Read/Write

Prerequisites: None

Sets or returns the drill symbol. The value can be a character, a CustomDrillSymbol name, or an EEnginesDrillSymbol.

Usage

```
DrillHole.DrillSymbol = Variant
```

Arguments

None

Return Values

Variant. A variant that can be a character, a custom drill symbol name or an EEnginesDrillSymbol Enum.

DrillSymbolSize Property (DrillHole Object)

Object: DrillHole Object

Access: Read/Write

Prerequisites: None

Sets or returns the drill symbol size.

Usage

```
DrillHole.DrillSymbolSize ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double
```

Arguments

- eUnit

  EEnginesUnit Enum (Optional) If nothing is specified then the current unit is used.

Return Values

Double. A double that contains the drill symbol size.

DrillSymbolType Property (DrillHole Object)

Object: DrillHole Object

Access: Read/Write

Prerequisites: None

Sets or returns the drill symbol type.

Usage

```
DrillHole.DrillSymbolType = EEnginesDrillSymbolType
```

Arguments

None

Return Values

EEnginesDrillSymbolType. The drill symbol type as EEnginesDrillSymbolType Enum.

FCF Property (DrillHole Object)

Object: DrillHole Object

Access: Read-Only

Prerequisites: None

Returns the feature control frame that specifies the precise geometric tolerance information for this hole.

Usage

DrillHole.FCF

Arguments

None

Return Values

FCF. Returns the FCF Object for this hole.

Name Property (DrillHole Object)

Object: DrillHole Object

Access: Read-Only

Prerequisites: None

Returns the name of the drill hole.

Usage

```
DrillHole.Name
```

Arguments

None

Return Values

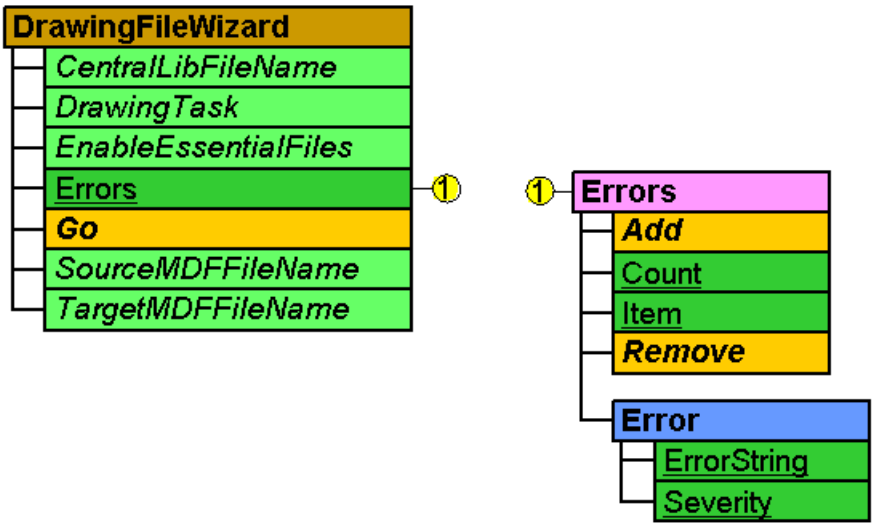
String. A string that contains the name of the drill hole.

DrawingFileWizard Object

The DrawingFileWizard Object contains the information to perform a specified drawing operation.

For more information on the Xpedition FabLink Drawing Wizard see the *Drawing Editor Guide*.

Figure 3. Drawing Wizard Data Model



The following example shows a script sample that runs Drawing Wizard automation using **mgcscript**.

You can find the complete script at: *SDD_HOME\standard\examples\pcb\Automation\AutoProEngines \RunDrawingWizard.vbs*

Example 3. Drawing Wizard Example Script

```
''''
'Visual Basic Script to Run DrawingFileWizard Automation.
''''
Option Explicit

' Parse arguments
Dim args
Set args = ScriptHelper.Arguments
If args.Count > 3 Then
    ' Call from command line
    ' mgcscript RunDrawingWizard.vbs [create]  [target MDF file]
  [source MDF file] [Central library]
    ' mgcscript RunDrawingWizard.vbs [copy]    [target MDF file]
  [source MDF file]
    ' mgcscript RunDrawingWizard.vbs [move]    [target MDF file]
  [source MDF file]
    ' mgcscript RunDrawingWizard.vbs [delete]  [target MDF file]
    ' mgcscript RunDrawingWizard.vbs [cleanup] [target MDF file]
```

```
  Dim TargetMDFFileName, SourceMDFFileName, CentralLibFileName
  Dim drawingTask, resultVal

  drawingTask = args.item(3)
  resultVal = False
  TargetMDFFileName = ""
  SourceMDFFileName = ""
  CentralLibFileName = ""

  Select Case drawingTask
  Case "create"
    If args.Count = 6 Then
      TargetMDFFileName = args.item(4)
      SourceMDFFileName = args.item(5)
      CentralLibFileName = args.item(6)
      resultVal = True
    End If

  Case "copy"
    If args.Count = 5 Then
      TargetMDFFileName = args.item(4)
      SourceMDFFileName = args.item(5)
      resultVal = True
    End If

  Case "move"
    If args.Count = 5 Then
      TargetMDFFileName = args.item(4)
      SourceMDFFileName = args.item(5)
      resultVal = True
    End If

  Case "delete"
    If args.Count = 4 Then
      TargetMDFFileName = args.item(4)
      resultVal = True
    End If

  Case "cleanup"
    If args.Count = 4 Then
      TargetMDFFileName = args.item(4)
      resultVal = True
    End If

  Case Else
      MsgBox "Cannot recognize drawing task."
      resultVal = False
  End Select

  If resultVal = True Then
    Call RunDrawingFileWizard(drawingTask, TargetMDFFileName,
SourceMDFFileName, CentralLibFileName)
    Else
```

```
      Call DoHelp
    End If

Else
  Call DoHelp
End If


'*************************************************************************
' RunDrawingFileWizard
'*************************************************************************
Sub RunDrawingFileWizard(drawingTask, TargetMDFFileName, SourceMDFFileName,
 CentralLibFileName)

  ' Create DrawingWizard Engine object
  Dim oDrawingFileWizard, forceDoDelete
  Set oDrawingFileWizard = CreateObject("MGCPCBEngines.DrawingFileWizard")

  If oDrawingFileWizard is Nothing Then
      MsgBox "Failed to create DrawingFileWizard object."
      Exit Sub
  End If

  'AddTypeLibrary
  Scripting.AddTypeLibrary("MGCPCBEngines.DrawingFileWizard")

  Select Case drawingTask
    Case "create" ' create
      forceDoDelete = False
      Call DoDelete(oDrawingFileWizard, TargetMDFFileName, forceDoDelete)
      Call DoCreate(oDrawingFileWizard, SourceMDFFileName,
 TargetMDFFileName, CentralLibFileName)

    Case "copy" ' copy
      forceDoDelete = False
      Call DoDelete(oDrawingFileWizard, TargetMDFFileName, forceDoDelete)
      Call DoCopy(oDrawingFileWizard, SourceMDFFileName, TargetMDFFileName)

    Case "move" ' move
      forceDoDelete = False
      Call DoDelete(oDrawingFileWizard, TargetMDFFileName, forceDoDelete)
      Call DoMove(oDrawingFileWizard, SourceMDFFileName, TargetMDFFileName)

    Case "delete" ' delete
      forceDoDelete = True
      Call DoDelete(oDrawingFileWizard, TargetMDFFileName, forceDoDelete)

    Case "cleanup" 'cleanup
      Call DoCleanup(oDrawingFileWizard, TargetMDFFileName)

  End Select

  'End of DrawingWizard
  Set oDrawingFileWizard = Nothing
```

```
  MsgBox "Finished DrawingWizard Sucessfully."

End Sub

'*************************
' DoCreate
'*************************
Sub DoCreate(oDrawingFileWizard, SourceMDFFileName, TargetMDFFileName,
 CentralLibFileName)
    MsgBox "Create DrawingFile : " + TargetMDFFileName
    oDrawingFileWizard.DrawingTask = eengDrawingTaskCreate
    oDrawingFileWizard.SourceMDFFileName = SourceMDFFileName
    oDrawingFileWizard.TargetMDFFileName = TargetMDFFileName
    oDrawingFileWizard.CentralLibFileName = CentralLibFileName
    oDrawingFileWizard.Go()
End Sub

'*************************
' DoCopy
'*************************
Sub DoCopy(oDrawingFileWizard, SourceMDFFileName, TargetMDFFileName)
    MsgBox "Copy DrawingFile: " + SourceMDFFileName + " -> " +
 TargetMDFFileName
    oDrawingFileWizard.DrawingTask = eengDrawingTaskCopy
    oDrawingFileWizard.SourceMDFFileName = SourceMDFFileName
    oDrawingFileWizard.TargetMDFFileName = TargetMDFFileName
    oDrawingFileWizard.EnableEssentialFiles = True
    oDrawingFileWizard.Go()
End Sub

'*************************
' DoDelete
'*************************
Sub DoDelete(oDrawingFileWizard, TargetMDFFileName, forceDoDelete)

  If forceDoDelete = False Then
    Dim oFS
    Set oFS = CreateObject("Scripting.FileSystemObject")
    If oFS.FileExists(TargetMDFFileName) = True Then
      forceDoDelete = True
    End If
    Set oFS = Nothing
  Else
    MsgBox "Delete DrawingFile: " + TargetMDFFileName
  End If

  If forceDoDelete = True Then
    oDrawingFileWizard.DrawingTask = eengDrawingTaskDelete
    oDrawingFileWizard.TargetMDFFileName = TargetMDFFileName
    oDrawingFileWizard.Go()
  End If
End Sub

'*************************
```

```
' DoMove
'*************************
Sub DoMove(oDrawingFileWizard, SourceMDFFileName, TargetMDFFileName)
    MsgBox "Move DrawingFile: " + SourceMDFFileName + " -> "  +
 TargetMDFFileName
    oDrawingFileWizard.DrawingTask = eengDrawingTaskMove
    oDrawingFileWizard.SourceMDFFileName = SourceMDFFileName
    oDrawingFileWizard.TargetMDFFileName = TargetMDFFileName
    oDrawingFileWizard.Go()
End Sub

'*************************
' DoCleanup
'*************************
Sub DoCleanup(oDrawingFileWizard, TargetMDFFileName)
    MsgBox "Cleanup DrawingFile: " + TargetMDFFileName
    oDrawingFileWizard.DrawingTask = eengDrawingTaskCleanup
    oDrawingFileWizard.TargetMDFFileName = TargetMDFFileName
    oDrawingFileWizard.Go()
End Sub

'*************************
' DoHelp
'*************************
Sub DoHelp

  MsgBox  "mgcscript RunDrawingWizard.vbs [create] [target MDF file]
 [source MDF file] [Central library]" & Chr(13) & _
      "mgcscript RunDrawingWizard.vbs [copy] [target MDF file] [source MDF
 file]" & Chr(13) &_
      "mgcscript RunDrawingWizard.vbs [move] [target MDF file] [source MDF
 file]" & Chr(13) &_
      "mgcscript RunDrawingWizard.vbs [delete] [target MDF file]" & Chr(13)
  &_
      "mgcscript RunDrawingWizard.vbs [cleanup] [target MDF file]"
End Sub
```

Table 5. DrawingFileWizard Object Methods and Properties

Method or Property	Description
CentralLibFileName Property (DrawingFileWizard Object)	Sets the central library file name (full pathname). The create operation requires this property.
DrawingTask Property (DrawingFileWizard Object)	Sets the operation to perform (create, copy, move, cleanup, or delete).
EnableEssentialFiles Property (DrawingFileWizard Object)	Copies only essential files if set to True. This property is available for the copy operation.
Errors Property (DrawingFileWizard Object)	Returns the collection of Errors.
Go Method (DrawingFileWizard Object)	Runs the Drawing Wizard and performs the specified operation.
SourceMDFFileName Property (DrawingFileWizard Object)	Sets the source drawing file (.mdf) name (full pathname). This is a path to the template for the create operation.

Table 5. DrawingFileWizard Object Methods and Properties (continued)

Method or Property	Description
TargetMDFFileName Property (DrawingFileWizard Object)	Sets the target drawing file (.mdf) name (full pathname). Create, copy, and move operations require this property.

CentralLibFileName Property (DrawingFileWizard Object)

Object: DrawingFileWizard Object

Access: Write-Only

Prerequisites: None

Sets the central library file name (full pathname). The create operation requires this property.

Usage

```
DrawingFileWizard.CentralLibFileName = String
```

Arguments

None

Return Values

String. A string that contains the full pathname to the central library including the (*.lmc*) extension.

DrawingTask Property (DrawingFileWizard Object)

Object: DrawingFileWizard Object

Access: Write-Only

Prerequisites: None

Sets the operation to perform (create, copy, move, cleanup, or delete).

Usage

```
DrawingFileWizard.DrawingTask =  EEnginesDrawingTask
```

Arguments

None

Return Values

EEnginesDrawingTask. The operation as EEnginesDrawingTask Enum.

EnableEssentialFiles Property (DrawingFileWizard Object)

Object: DrawingFileWizard Object

Access: Write-Only

Prerequisites: None

Copies only essential files if set to True. This property is available for the copy operation.

Usage

```
DrawingFileWizard.EnableEssentialFiles = True | False
```

Arguments

None

Return Values

True | False. If not set, the default is True. If True, does not include the *LogFiles* and *Work* directories in the copy operation. If False, includes the *LogFiles* and *Work* directories.

Errors Property (DrawingFileWizard Object)

Object: DrawingFileWizard Object

Access: Read-Only

Prerequisites: None

Returns the collection of Errors.

Usage

```
DrawingFileWizard.errors
```

Arguments

None

Return Values

Errors. The collection of errors (Errors Collection).

Go Method (DrawingFileWizard Object)

Object: DrawingFileWizard Object

Prerequisites: None

Runs the Drawing Wizard and performs the specified operation.

Usage

```
DrawingFileWizard.Go()
```

Arguments

None

SourceMDFFileName Property (DrawingFileWizard Object)

Object: DrawingFileWizard Object

Access: Write-Only

Prerequisites: None

Sets the source drawing file (*.mdf*) name (full pathname). This is a path to the template for the create operation.

Usage

DrawingFileWizard.SourceMDFFileName = String

Arguments

None

Return Values

String. A string that contains the full pathname to the source drawing file including the *.mdf* extension.

Examples

The following snippet shows a create operation specifying the source file as a 4 layer drawing template.

```
  oDrawingFileWizard.DrawingTask = eengDrawingTaskCreate
  oDrawingFileWizard.SourceMDFFileName = &_
"H:\CentralLibrary\Templates\Drawing\4LayerTemplate\template.mdf"
  oDrawingFileWizard.TargetMDFFileName = &_
"U:\MyDesign\Drawing\drawing.mdf"
  oDrawingFileWizard.CentralLibFileName = &_
"H:\CentralLibrary\CentralLibrary.lmc"
  oDrawingFileWizard.Go()
```

TargetMDFFileName Property (DrawingFileWizard Object)

Object: DrawingFileWizard Object

Access: Write-Only

Prerequisites: None

Sets the target drawing file (*.mdf*) name (full pathname). Create, copy, and move operations require this property.

Usage

```
DrawingFileWizard.TargetMDFFileName = String
```

Arguments

None

Return Values

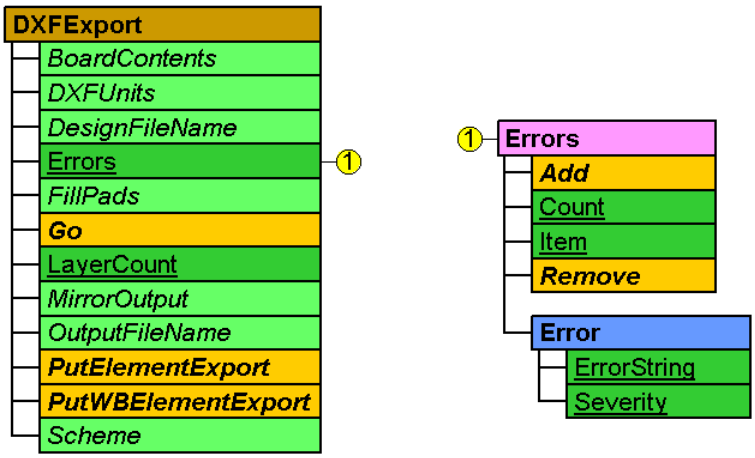
String. A string that contains the full pathname for the new or target drawing file including the *.mdf* extension.

DXFExport Object

The Drawing Interchange Format (DXF) allows the exchange of data between computer-aided design (CAD) systems. The DXFExport Object contains the information that the DXFExport Engine uses to perform the specified export operation.

For more information on exporting DXF refer to the DXF Export dialog box by choosing **File > Export > DXF**.

Figure 4. DXFExport Data Model



Find the sample script at this location:

*SDD_HOME\standard\examples\pcb\Automation\AutoProEngines\RunDXFExport.vbs*runs batch DXF Export.

Table 6. DXFExport Object Methods and Properties

Method or Property	Description
BoardContents Property (DXFExport Object)	Includes board data in the DXF output if set to True. (Panel designs only.)
DesignFileName Property (DXFExport Object)	Sets the design file name, including path.
DXFUnits Property (DXFExport Object)	Sets the output units for outputting DXF.
Errors Property (DXFExport Object)	Returns the collection of errors.
FillPads Property (DXFExport Object)	Fills pads if set to True.
Go Method (DXFExport Object)	Runs the export engine and creates DXF output.
LayerCount Property (DXFExport Object)	Returns the number of layers in the current design.
MirrorOutput Property (DXFExport Object)	Mirrors the output if set to True.
OutputFileName Property (DXFExport Object)	Sets the output file name, including path.

Table 6. DXFExport Object Methods and Properties (continued)

Method or Property	Description
PutElementExport Method (DXFExport Object)	Adds or modifies elements to export to DXF.
PutWBElementExport Method (DXFExport Object)	Adds or modifies bondwire or die pin elements to export to DXF.
Scheme Property (DXFExport Object)	Sets a scheme.

BoardContents Property (DXFExport Object)

Object: DXFExport Object

Access: Write-Only

Prerequisites: None

Includes board data in the DXF output if set to True. (Panel designs only.)

Usage

```
DXFExport.BoardContents = True | False
```

Arguments

None

Return Values

True | False. If True, include board data. If False, do not include board data.

DesignFileName Property (DXFExport Object)

Object: DXFExport Object

Access: Write-Only

Prerequisites: None

Sets the design file name, including path.

Usage

```
DXFExport.DesignFileName = String
```

Arguments

None

Return Values

String. A string that contains the design name including the path.

DXFUnits Property (DXFExport Object)

Object: DXFExport Object

Access: Write-Only

Prerequisites: None

Sets the output units for outputting DXF.

Usage

```
DXFExport.DXFUnits = EEnginesUnit
```

Arguments

None

Return Values

EEnginesUnit. The units as EEnginesUnit Enum.

Errors Property (DXFExport Object)

Object: DXFExport Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

```
DXFExport.Errors = Errors
```

Arguments

None

Return Values

Errors. The collection of errors (Errors Collection).

FillPads Property (DXFExport Object)

Object: DXFExport Object

Access: Write-Only

Prerequisites: None

Fills pads if set to True.

Usage

```
DXFExport.FillPads = True | False
```

Arguments

None

Return Values

True | False. If True, fill pads. If False, do not fill pads.

Go Method (DXFExport Object)

Object: DXFExport Object

Access: Write-Only

Prerequisites: None

Runs the export engine and creates DXF output.

Usage

```
DXFExport.Go()
```

Arguments

None

LayerCount Property (DXFExport Object)

Object: DXFExport Object

Access: Read-Only

Prerequisites: None

Returns the number of layers in the current design.

Usage

```
DXFExport.LayerCount = Long
```

Arguments

None

Return Values

Long. A long that contains the number of layers.

MirrorOutput Property (DXFExport Object)

Object: DXFExport Object

Access: Write-Only

Prerequisites: None

Mirrors the output if set to True.

Usage

```
DXFExport.MirrorOutput = True | False
```

Arguments

None

Return Values

True | False. If True, mirror output. If False, do not mirror output.

OutputFileName Property (DXFExport Object)

Object: DXFExport Object

Access: Write-Only

Prerequisites: None

Sets the output file name, including path.

Usage

```
DXFExport.OutputFileName = String
```

Arguments

None

Return Values

String. A string that contains the output file name and path.

PutElementExport Method (DXFExport Object)

Object: DXFExport Object

Prerequisites: None

Adds or modifies elements to export to DXF.

Usage

```
DXFExport.PutElementExport(
        ByVal eElementType As EEnginesElementType,
        ByVal DXFLayerName As String,
        ByVal layer As Variant,
        [ByVal exportLayer As Boolean = True])
```

Arguments

- eElementType

  A design element as EEnginesElementType Enum.

- DXFLayerName

  A string that contains the DXF layer name.

- layer

  A string that contains a design layer name or layer number.

- exportLayer

  (Optional) If True, export this layer. If False, do not export this layer. The default is to export the layer.

PutWBElementExport Method (DXFExport Object)

Object: DXFExport Object

Prerequisites: None

Adds or modifies bondwire or die pin elements to export to DXF.

Usage

```
DXFExport.PutWBElementExport(
        ByVal eElementType As EEnginesElementType,
        ByVal DXFLayerName As String,
        ByVal layer As Long,
        [ByVal stackLevel As Long],
        [ByVal elementSide As EEnginesSide = eengSideTop],
        [ByVal exportLayer As Boolean = True])
```

Arguments

- eElementType

  A design element as EEnginesElementType Enum. This can be eengElementTypeBondWires or eengElementTypeDiePins.

- DXFLayerName

  A string that contains the DXF layer name.

- layer

  A long that contains a layer number.

- stackLevel

  (Optional) A long that contains the level number.

- eElementSide

  (Optional) The element's side as EEnginesSide Enum. If not specified, the side is top.

- exportLayer

  (Optional) If True, export this layer. If False, do not export this layer. The default is to export the layer.

Scheme Property (DXFExport Object)

Object: DXFExport Object

Access: Write-Only

Prerequisites: None

Sets a scheme.

Usage

```
DXFExport.Scheme = String
```

Arguments

None

Return Values

String. A string that contains the name and path of the Scheme file.

Error Object

The Error Object contains information related to error reporting.

See also Errors Collection.

Table 7. Error Object Properties

Property	Description
ErrorString Property (Error Object)	Returns the error string.
Severity Property (Error Object)	Returns the severity of an error.

ErrorString Property (Error Object)

Object: Error Object

Access: Read-Only

Prerequisites: None

Returns the error string.

Usage

Error.ErrorString

Arguments

None

Return Values

String. An error string

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oDRCEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

Severity Property (Error Object)

Object: Error Object

Access: Read-Only

Prerequisites: None

Returns the severity of an error.

Usage

Error.Severity

Arguments

None

Return Values

EEnginesErrorSeverity

Examples

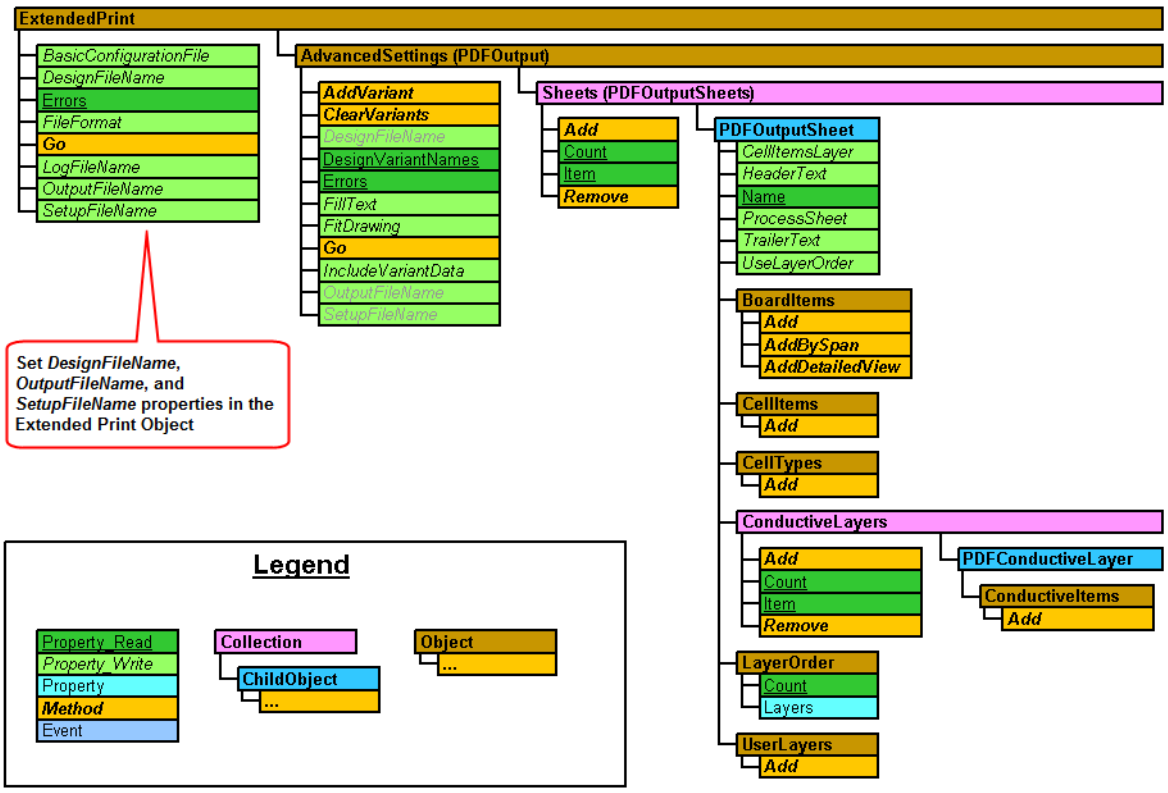
```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oDRCEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

ExtendedPrint Object

The ExtendedPrint Object contains the information the Extended Print Engine uses to perform the specified operation.

For more information on extended print, refer to the Extended Print dialog box by choosing **File > Printing > Extended Print**.

Figure 5. Extended Print Data Model



The following example shows a script sample that runs batch Extended Print.

Find the complete script at this location:*SDD_HOME\standard\examples\pcb\Automation\AutoProEngines\RunExtendedPrint.vbs*

Example 4. Extended Print Example Script

```
Option Explicit

Dim jobName
Dim args
Dim argsCnt
Dim setupFile
Dim configFile
Dim pcbPath
```

```
Dim outputFile

Set args = ScriptHelper.Arguments
argsCnt = args.count
setupFile = ""
If argsCnt >= 4  Then
    jobName = args.item(3)
    configFile= args.item(4)
    If(argsCnt = 5) Then
     setupFile = args.item(5)
    End If
    Call RunExtendedPrint
Else
    MsgBox "Usage: mgcscript TestMGCPCBEnginesExtPrint.vbs [job name]
 [setup_file]"
End If

'*************************************************************************
'*** Main Function
'*************************************************************************
Sub RunExtendedPrint()
    ' Create PDFOutput Engine object
    Dim oExtendedPrintEngine
    Set oExtendedPrintEngine = CreateObject("MGCPCBEngines.ExtendedPrint")
    Scripting.AddTypeLibrary("MGCPCBEngines.ExtendedPrint")

    If InStr(jobName, ".pnl") Then
        pos = InstrRev(jobName, dirSlash, -1, 1)
        pcbPath = Left(jobName, Len(jobName) - pos)
        outputFile = pcbPath(0) + "/Output/" + "ExtendedPrint.pdf"
    Else
        pcbPath = Split(jobName, "PCB")
        outputFile = pcbPath(0) + "PCB/Output/" + "ExtendedPrint.pdf"
    End If
    oExtendedPrintEngine.DesignFileName = jobName

    If setupFile <> "" Then
     oExtendedPrintEngine.SetupFileName = setupFile
    End if
    oExtendedPrintEngine.BasicConfigurationFile = configFile
    oExtendedPrintEngine.OutputFileName=outputFile

    ' Run the ExtendedPrint Engine
    oExtendedPrintEngine.Go()

    MsgBox "Finished Extended Print"
    ' done
End Sub
```

Table 8. ExtendedPrint Object Methods and Properties

Method or Property	Description
AdvancedSettings Property (ExtendedPrint Object)	Returns the handle to set the Advanced Settings (PDF Output).
BasicConfigurationFile Property (ExtendedPrint Object)	Sets the Extended Print basic configuration file. If the property is not specified, the software uses the last saved configuration.
DesignFileName Property (ExtendedPrint Object)	Sets the design file name, including path. You can specify a board, panel, or drawing design (pcb, .pnl or .mdf).
Errors Property (ExtendedPrint Object)	Returns the handle to the errors collection.
FileFormat Property (ExtendedPrint Object)	Sets the file format to output from Extended Print.
Go Method (ExtendedPrint Object)	Runs the Extended Print engine.
LogFileName Property (ExtendedPrint Object)	Sets the Extended Print log file name, including path.
OutputFileName Property (ExtendedPrint Object)	Sets the Extended Print output file name.
SetupFileName Property (ExtendedPrint Object)	Sets the Extended Print advanced setup file name, including path.

AdvancedSettings Property (ExtendedPrint Object)

Object: ExtendedPrint Object

Access: Read-Only

Prerequisites: None

Returns the handle to set the Advanced Settings (PDF Output).

Note:
Do not use DesignFileName, LogFileName, OutputFileName and SetupFileName properties of the PDFOutput object to setup the ExtendedPrint engine. This results in an error. Instead, use the ExtendedPrint properties.

Usage

```
ExtendedPrint.AdvancedSettings = IPDFOutput
```

Arguments

None

Return Values

IPDFOutput. The PDFOutput Object.

BasicConfigurationFile Property (ExtendedPrint Object)

Object: ExtendedPrint Object

Access: Write-Only

Prerequisites: None

Sets the Extended Print basic configuration file. If the property is not specified, the software uses the last saved configuration.

Usage

```
ExtendedPrint.BasicConfigurationFile = String
```

Arguments

None

Return Values

String. A string that contains the setup configuration.

DesignFileName Property (ExtendedPrint Object)

Object: ExtendedPrint Object

Access: Write-Only

Prerequisites: None

Sets the design file name, including path. You can specify a board, panel, or drawing design (*pcb*, *.pnl* or *.mdf*).

Usage

```
ExtendedPrint.DesignFileName = String
```

Arguments

None

Return Values

String. A string that contains the design name including full path.

Errors Property (ExtendedPrint Object)

Object: ExtendedPrint Object

Access: Read-Only

Prerequisites: None

Returns the handle to the errors collection.

Usage

```
ExtendedPrint.Errors = IErrors
```

Arguments

None

Return Values

IErrors. Returns the collection of Error Objects.

FileFormat Property (ExtendedPrint Object)

Object: ExtendedPrint Object

Access: Write-Only

Prerequisites: None

Sets the file format to output from Extended Print.

Usage

```
ExtendedPrint.FileFormat = EEnginesExtPrintFileFormat
```

Arguments

None

Return Values

EEnginesExtPrintFileFormat. The output format as EEnginesExtPrintFileFormat Enum.

Go Method (ExtendedPrint Object)

Object: ExtendedPrint Object

Prerequisites: None

Runs the Extended Print engine.

Usage

```
ExtendedPrint.Go()
```

Arguments

None

LogFileName Property (ExtendedPrint Object)

Object: ExtendedPrint Object

Access: Write-Only

Prerequisites: None

Sets the Extended Print log file name, including path.

Usage

```
ExtendedPrint.LogFileName = String
```

Arguments

None

Return Values

String. A string that contains the log file name including full path.

OutputFileName Property (ExtendedPrint Object)

Object: ExtendedPrint Object

Access: Write-Only

Prerequisites: None

Sets the Extended Print output file name.

Usage

```
ExtendedPrint.OutputFileName = String
```

Arguments

None

Return Values

String. A string that contains the name of the output file.

SetupFileName Property (ExtendedPrint Object)

PObject: ExtendedPrint Object

Access: Write-Only

Prerequisites: None

Sets the Extended Print advanced setup file name, including path.

Usage

```
ExtendedPrint.SetupFileName = String
```

Arguments

None

Return Values

String. A string that contains the setup file name.

FCF Object

The FCF Object contains information related to the Feature Control Frame.

Table 9. FCF Object Properties

Property	Description
CharSymbol Property (FCF Object)	Sets or returns the characteristic symbol for the defined FCF.
DatumReferenceA Property (FCF Object)	Sets or returns the FCF Datum Reference A.
DatumReferenceB Property (FCF Object)	Sets or returns the FCF Datum Reference B.
DatumReferenceC Property (FCF Object)	Sets or returns the FCF Datum Reference C.
Tolerance Property (FCF Object)	Sets or returns the FCF tolerance.

CharSymbol Property (FCF Object)

Object: FCF Object

Access: Read/Write

Prerequisites: None

Sets or returns the characteristic symbol for the defined FCF.

Usage

```
FCF.CharSymbol = String
```

Arguments

None

Return Values

String. The return/set type for this property.

DatumReferenceA Property (FCF Object)

Object: FCF Object

Access: Read/Write

Prerequisites: None

Sets or returns the FCF Datum Reference A.

Usage

```
FCF.DatumReferenceA = String
```

Arguments

None

Return Values

String. A string that contains datum reference A.

DatumReferenceB Property (FCF Object)

Object: FCF Object

Access: Read/Write

Prerequisites: None

Sets or returns the FCF Datum Reference B.

Usage

```
FCF.DatumReferenceB = String
```

Arguments

None

Return Values

String. A string that contains datum reference B.

DatumReferenceC Property (FCF Object)

Object: FCF Object

Access: Read/Write

Prerequisites: None

Sets or returns the FCF Datum Reference C.

Usage

```
FCF.DatumReferenceC = String
```

Arguments

None

Return Values

String. A string that contains datum reference C.

Tolerance Property (FCF Object)

Object: FCF Object

Access: Read/Write

Prerequisites: None

Sets or returns the FCF tolerance.

Usage

```
FCF.Tolerance = String
```

Arguments

None

Return Values

String. The return/set type for this property.

GeneralInterfaceEngine Object

The GeneralInterfaceEngine Object contains the information the General Interface Engine uses for the specified interface.

Refer to the General Interfaces dialog box by choosing **File > Export > General Interfaces**

Figure 6. GeneralInterfaceEngine Data Model

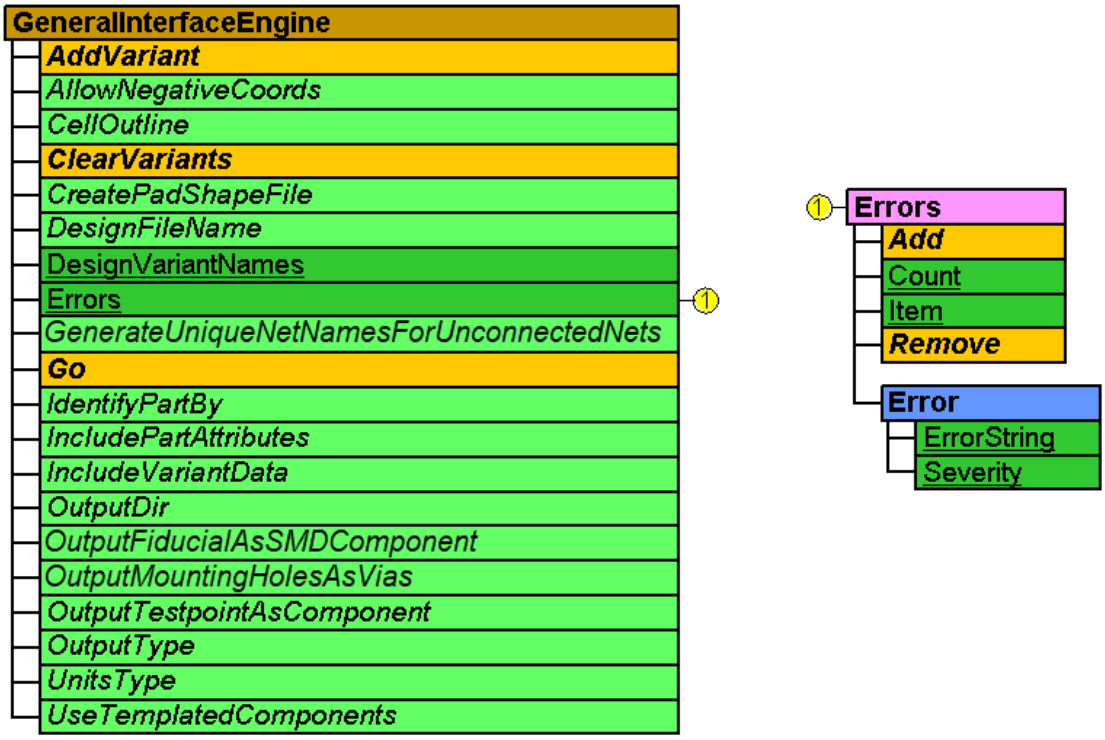


Table 10. GeneralInterfaceEngine Methods and Properties

Method or Property	Description
AddVariant Method (GeneralInterfaceEngine Object)	Adds the design variant to the output variant list.
ClearVariants Method (GeneralInterfaceEngine Object)	Clears the output variant list.
Go Method (GeneralInterfaceEngine Object)	Generates output based on the interface type.
AllowNegativeCoords Property (GeneralInterfaceEngine Object)	Sets the option to allow negative coordinates. (Fabmaster output only.) Refer to the Allow negative coordinates in output in the General Interfaces dialog box.
CellOutline Property (GeneralInterfaceEngine Object)	Sets the cell outline type.(Fabmaster output only.) Refer to the Select Cell Outline option in the General Interfaces dialog box.

Table 10. GeneralInterfaceEngine Methods and Properties (continued)

Method or Property	Description
CreatePadShapeFile Property (GeneralInterfaceEngine Object)	Creates the pad shape file. (C-Link output only.) Refer to the Create Pad Shape File option in the General Interfaces dialog box.
DesignFileName Property (GeneralInterfaceEngine Object)	Sets the name of the design file, including path.
DesignVariantNames Property (GeneralInterfaceEngine Object)	Returns the variant names for the current design.
Errors Property (GeneralInterfaceEngine Object)	Returns the collection of errors.
GenerateUniqueNetNamesForUnconnectedNets Property (GeneralInterfaceEngine Object)	Generates unique net names for unconnected nets in IPC-D-356 output.
IdentifyPartBy Property (GeneralInterfaceEngine Object)	Sets the option to identify parts. Refer to the Identify parts by option in the General Interfaces dialog box.
IncludePartAttributes Property (GeneralInterfaceEngine Object)	Includes part attributes in Gencad output if set to True.
IncludeVariantData Property (GeneralInterfaceEngine Object)	Includes the variant data (if it exists) if set to True. If no variant is specified, the property outputs the Master design.
OutputDir Property (GeneralInterfaceEngine Object)	Sets the output directory location.
OutputFiducialAsSMDComponent Property (GeneralInterfaceEngine Object)	Outputs fiducials as SMD components in IPC-D-356 output.
OutputMountingHolesAsVias Property (GeneralInterfaceEngine Object)	Outputs mounting holes as vias in IPC-D-356 output.
OutputTestpointAsComponent Property (GeneralInterfaceEngine Object)	Sets the option to output testpoints as components. (Fabmaster output only.) Refer to the Output Testpoints as Components option in the General Interfaces dialog box.
OutputType Property (GeneralInterfaceEngine Object)	Sets the type of output. Refer to the Interface Name option in the General Interfaces dialog box.
UnitsType Property (GeneralInterfaceEngine Object)	Sets the units type. (Fabmaster output only.) Refer to the Units option in the General Interfaces dialog box.
UseTemplatedComponents Property (GeneralInterfaceEngine Object)	Specifies to use templated components. (C-Link output only.) Refer to the Use Templated Components option in the General Interfaces dialog box.

AddVariant Method (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Prerequisites: None

Adds the design variant to the output variant list.

Usage

```
GeneralInterfaceEngine.AddVariant [ByVal variantName As String = “All”])
```

Arguments

- variantName

  (Optional) A string that contains the variant name. If not specified, the method adds all variant names to the output list.

ClearVariants Method (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Prerequisites: None

Clears the output variant list.

Usage

```
GeneralInterfaceEngine.ClearVariants()
```

Arguments

None

Go Method (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Prerequisites: None

Generates output based on the interface type.

Usage

GeneralInterfaceEngine.Go()

Arguments

None

Examples

```
oGeneralInterfaceEngine.Go
```

AllowNegativeCoords Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to allow negative coordinates. (Fabmaster output only.) Refer to the Allow negative coordinates in output in the General Interfaces dialog box.

Usage

GeneralInterfaceEngine.AllowNegativeCoords = True | False

Arguments

None

Return Values

Boolean. If True, allow negative coordinates. If False, do not allow negative coordinates.

Examples

```
oGeneralInterfaceEngine.AllowNegativeCoords = True
```

CellOutline Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Sets the cell outline type.(Fabmaster output only.) Refer to the Select Cell Outline option in the General Interfaces dialog box.

Usage

GeneralInterfaceEngine.CellOutline = EEnginesCellOutlineType

Arguments

None

Return Values

EEnginesCellOutlineType. A cell outline type (“EEngingesCellOutlineType” on page 507)

Examples

```
oGeneralInterfaceEngine.CellOutline = eengCellPlacementOutline
```

CreatePadShapeFile Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Creates the pad shape file. (C-Link output only.) Refer to the Create Pad Shape File option in the General Interfaces dialog box.

Usage

GeneralInterfaceEngine.CreatePadShapeFile = True | False

Arguments

None

Return Values

Boolean. If True, create the pad shape file. If False, do not create the pad shape file.

Examples

```
oGeneralInterfaceEngine.CreatePadShapeFile = True
```

DesignFileName Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

GeneralInterfaceEngine.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the name of the design file.

Examples

```
oGeneralInterfaceEngine.DesignFileName = jobName
```

DesignVariantNames Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Read-Only

Prerequisites: None

Returns the variant names for the current design.

Usage

```
GeneralInterfaceEngine.DesignVariantNames
```

Arguments

None

Return Values

Array. A array of variant names.

Errors Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

GeneralInterfaceEngine.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oGeneralInterfaceEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

GenerateUniqueNetNamesForUnconnectedNets Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Generates unique net names for unconnected nets in IPC-D-356 output.

Usage

GeneralInterfaceEngine.GenerateUniqueNetNamesForUnconnectedNets = True | False

Arguments

None

Return Values

Boolean. If True, generate unique net names for unconnected nets in IPC-D-356 output. If False, do not generate unique net names for unconnected nets in IPC-D-356 output.

Examples

```
oGeneralInterfaceEngine.GenerateUniqueNetNamesForUnconnectedNets = True
```

IdentifyPartBy Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to identify parts. Refer to the Identify parts by option in the General Interfaces dialog box.

Usage

GeneralInterfaceEngine.IdentifyPartBy = EEnginesGeneralInterfaceIdentifyPartsType

Arguments

None

Return Values

The identify parts type ("EEnginesGeneralInterfaceIdentifyPartsType" on page 552).

Examples

```
oGeneralInterfaceEngine.IdentifyPartBy = eengGenIntPartName
```

IncludePartAttributes Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Includes part attributes in Gencad output if set to True.

Usage

```
GeneralInterfaceEngine.IncludePartAttributes = True | False
```

Arguments

None

Return Values

Boolean. If True, includes part attributes. If False, do not include part attributes.

IncludeVariantData Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Includes the variant data (if it exists) if set to True. If no variant is specified, the property outputs the Master design.

Usage

```
GeneralInterfaceEngine.IncludeVariantData = True | False
```

Arguments

None

Return Values

Boolean. If True, include variant data. If False, do not include variant data.

OutputDir Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Sets the output directory location.

Usage

GeneralInterfaceEngine.OutputDir = String

Arguments

None

Return Values

String. A string that contains the output directory.

Examples

```
oGeneralInterfaceEngine.OutputDir = outputDir
```

OutputFiducialAsSMDComponent Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Outputs fiducials as SMD components in IPC-D-356 output.

Usage

GeneralInterfaceEngine.OutputFiducialAsSMDComponent = True | False

Arguments

None

Return Values

Boolean. If True, output fiducials as SMD components in IPC-D-356 output. If False, do not output fiducials as SMD components in IPC-D-356 output.

Examples

```
oGeneralInterfaceEngine.OutputFiducialAsSMDComponent = True
```

OutputMountingHolesAsVias Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Outputs mounting holes as vias in IPC-D-356 output.

Usage

GeneralInterfaceEngine.OutputMountingHolesAsVias = True | False

Arguments

None

Return Values

Boolean. If True, output mounting holes as vias in IPC-D-356 output. If False, do not output mounting holes as vias in IPC-D-356 output.

Examples

```
oGeneralInterfaceEngine.OutputMountingHolesAsVias = True
```

OutputTestpointAsComponent Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to output testpoints as components. (Fabmaster output only.) Refer to the Output Testpoints as Components option in the General Interfaces dialog box.

Usage

GeneralInterfaceEngine.OutputTestpointAsComponent = True | False

Arguments

None

Return Values

Boolean. If True, testpoints are output as components. If False, testpoints are not output as components.

Examples

```
oGeneralInterfaceEngine.OutputTestpointAsComponent = True
```

OutputType Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Sets the type of output. Refer to the Interface Name option in the General Interfaces dialog box.

Usage

GeneralInterfaceEngine.OutputType = EEnginesGeneralInterfaceOutputType

Arguments

None

Return Values

The output format type (“EEnginesGeneralInterfaceOutputType” on page 553).

Examples

```
oGeneralInterfaceEngine.OutputType = eengGenIntOutputFabMaster
```

UnitsType Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Sets the units type. (Fabmaster output only.) Refer to the Units option in the General Interfaces dialog box.

Usage

GeneralInterfaceEngine.UnitsType = EEnginesGeneralInterfaceUnitsType

Arguments

None

Return Values

The unit type. (“EEnginesGeneralInterfaceUnitsType” on page 554)

Examples

```
oGeneralInterfaceEngine.UnitsType = eengGeneralInterfaceUnitOfOneTenth
```

UseTemplatedComponents Property (GeneralInterfaceEngine Object)

Object: GeneralInterfaceEngine Object

Access: Write-Only

Prerequisites: None

Specifies to use templated components. (C-Link output only.) Refer to the Use Templated Components option in the General Interfaces dialog box.

Usage

GeneralInterfaceEngine.UseTemplatedComponents = True | False

Arguments

None

Return Values

Boolean. If True, use templated components. If False, do not use templated components.

Examples

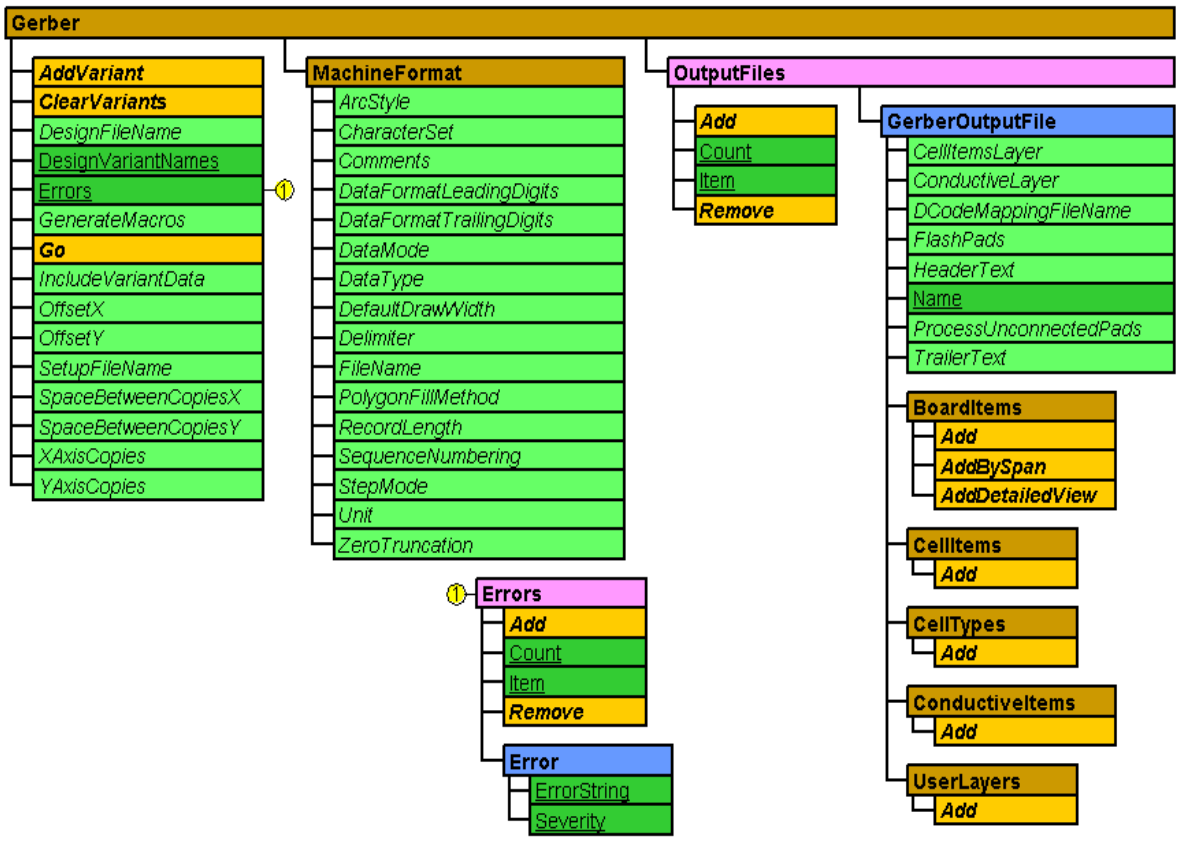
```
oGeneralInterfaceEngine.UseTemplatedComponents = True
```

Gerber Object

The Gerber Object contains the information the Gerber Engine uses for Gerber output.

Refer to the Gerber Output dialog box by choosing **Output > Gerber**

Figure 7. Gerber Data Model



The following example shows a script sample that runs batch Gerber output.

Find the complete script at this location: *SDD_HOME\standard\examples\pcb\Automation \AutoProEngines\RunGerber.vbs*

Example 5. Gerber Example Script

```
Option Explicit

Dim args, app, docObj, jobName, jobConfigPath
Set args = ScriptHelper.Arguments
If args.Count = 3 Then
    ' Call from command line
    ' mgcscript RunGerber.vbs [job name]
    jobName = args.item(3)
Else
    ' Call from Xpedition Layout
```

```
    Set app = GetObject(, "MGCPCB.ExpeditionPCBApplication")
    Set docObj = GetLicensedDoc(app)
    jobName = docObj.FullName
End If

Scripting.AddTypeLibrary("MGCPCBEngines.Gerber")

Call RunGerber()

MsgBox "fini", VBInformation, "RunGerber.vbs"

'*********************************************************************
***
Sub RunGerber()
    'On Error Resume Next

    ' Create Gerber Engine object
    Dim oGerberEngine
    Set oGerberEngine = CreateObject("MGCPCBEngines.Gerber")

    ' Set the design file name
    oGerberEngine.DesignFileName = jobName

    ' Get design Config directory
    Dim pcbPath
    pcbPath = Split(jobName, "PCB")
    jobConfigPath = pcbPath(0) + "PCB/Config/"

    Dim rtn
    rtn = MsgBox("Using existing setup files?", VBYesNo + VBQuestion,
 "RunGerber.vbs")

    If rtn = VBYes Then
        oGerberEngine.SetupFileName = jobConfigPath + "PlotSetup.gpf"
    Else
        ' Setup files for Gerber output
        Call SetupMachineFormat(oGerberEngine)
        Call SetupOutputFiles(oGerberEngine)
    End If

    ' Run the Gerber Engine
    oGerberEngine.Go

    ' Check errors
    Dim oErrors
    Set oErrors = oGerberEngine.Errors
End Sub

Sub SetupMachineFormat(oGerberEngine)
    'On Error Resume Next

'    Dim oGerberMachineFormat
'    Set oGerberMachineFormat = oGerberEngine.MachineFormat
```

```
'    oGerberMachineFormat.FileName = jobConfigPath +
 "UserGerberMachineFile.gmf"
'    oGerberMachineFormat.DataType = eengGerber274X
'    oGerberMachineFormat.DataMode = TRUE
'    oGerberMachineFormat.StepMode = eengStepIncremental
'    oGerberMachineFormat.DataFormatLeadingDigits = 3
'    oGerberMachineFormat.DataFormatTrailingDigits = 2
'    oGerberMachineFormat.ZeroTruncation = eengZeroTruncationNone
'    oGerberMachineFormat.CharacterSet = eengCharacterSetASCII
'    oGerberMachineFormat.ArcStyle = eengArcStyleQuadrant
'    oGerberMachineFormat.Delimiter = eengDataDelimiterStar
'    oGerberMachineFormat.Comments = TRUE
'    oGerberMachineFormat.SequenceNumbering = TRUE
'    oGerberMachineFormat.Unit = eengUnitInch
'    oGerberMachineFormat.PolygonFillMethod = eengPolygonFillDraw
'    oGerberMachineFormat.RecordLength = 0
End Sub

Sub SetupOutputFiles(oGerberEngine)
    'On Error Resume Next

    Dim oGerberOutputFiles
    Set oGerberOutputFiles = oGerberEngine.OutputFiles

    Dim oGOFAddEach
    Set oGOFAddEach = oGerberOutputFiles.Add("EachElements.gdo")
    oGOFAddEach.FlashPads = FALSE
    oGOFAddEach.HeaderText = "Gerber Output File Header Text"
    oGOFAddEach.TrailerText = "Gerber Output File Trailer Text"
    oGOFAddEach.ProcessUnconnectedPads = FALSE
    'oGOFAddEach.DCodeMappingFileName =
 "W:\2005\standard\config\pcb\sample.dmf"

    ' Add all board items individually
    Dim oBoardItems
    Set oBoardItems = oGOFAddEach.BoardItems
    'Call oBoardItems.Add (eengBoardItemBoardObstruct)
   ' oBoardItems.Add eengBoardItemBoardOrigin
 '   oBoardItems.Add eengBoardItemBoardOutline
    oBoardItems.Add eengBoardItemContours
   ' oBoardItems.Add eengBoardItemDetailViews
  '  oBoardItems.Add eengBoardItemDocumentation
  '  oBoardItems.Add eengBoardItemDrcWindow
    oBoardItems.Add eengBoardItemDrillDrawingThru
    oBoardItems.Add eengBoardItemGeneratedSilkscreenBottom
    oBoardItems.Add eengBoardItemGeneratedSilkscreenTop
  '  oBoardItems.Add eengBoardItemManufacturingOutline
    oBoardItems.Add eengBoardItemMountingHoles
    'oBoardItems.Add eengBoardItemNCDrillOrigin
  '  oBoardItems.Add eengBoardItemPanelBorder
 '   oBoardItems.Add eengBoardItemPanelHole
  '  oBoardItems.Add eengBoardItemPanelOrigin
   ' oBoardItems.Add eengBoardItemPanelOutline
```

```
    oBoardItems.Add eengBoardItemPartHoles
   'oBoardItems.Add eengBoardItemPlacementObstructBottom
   ' oBoardItems.Add eengBoardItemPlacementObstructTop
  '  oBoardItems.Add eengBoardItemRoomBottom
  '  oBoardItems.Add eengBoardItemRoomTop
  '  oBoardItems.Add eengBoardItemRouteBorder
   ' oBoardItems.Add eengBoardItemScoringLine
  '  oBoardItems.Add eengBoardItemShearingHoles
  '  oBoardItems.Add eengBoardItemShearingLine
    oBoardItems.Add eengBoardItemSoldermaskBottom
    oBoardItems.Add eengBoardItemSoldermaskTop
    oBoardItems.Add eengBoardItemSolderpasteBottom
    oBoardItems.Add eengBoardItemSolderpasteTop
   ' oBoardItems.Add eengBoardItemToolingHoles

    ' Add all cell types individually
    Dim oCellTypes
    Set oCellTypes = oGOFAddEach.CellTypes
   ' oCellTypes.Add eengCellTypeBadBoardIdentifier
   ' oCellTypes.Add eengCellTypeBreakAwayTab
    oCellTypes.Add eengCellTypeBuried
    oCellTypes.Add eengCellTypeConnector
    oCellTypes.Add eengCellTypeDiscreteAxial
    oCellTypes.Add eengCellTypeDiscreteChip
    oCellTypes.Add eengCellTypeDiscreteOther
    oCellTypes.Add eengCellTypeDiscreteRadial
    oCellTypes.Add eengCellTypeEdgeConnector
    'oCellTypes.Add eengCellTypeElectricalTestIdentifier
    oCellTypes.Add eengCellTypeGeneral
   ' oCellTypes.Add eengCellTypeGeneralPanelCell
    oCellTypes.Add eengCellTypeGraphic
    oCellTypes.Add eengCellTypeICBareDie
    oCellTypes.Add eengCellTypeICBGA
    oCellTypes.Add eengCellTypeICDIP
    oCellTypes.Add eengCellTypeICFlipChip
    oCellTypes.Add eengCellTypeICLCC
    oCellTypes.Add eengCellTypeICOther
    oCellTypes.Add eengCellTypeICPGA
    oCellTypes.Add eengCellTypeICPLCC
    oCellTypes.Add eengCellTypeICSIP
    oCellTypes.Add eengCellTypeICSOIC
    oCellTypes.Add eengCellTypeJumper
    oCellTypes.Add eengCellTypeMechanical
   ' oCellTypes.Add eengCellTypePanelIdentifier
   ' oCellTypes.Add eengCellTypePanelStiffener
   ' oCellTypes.Add eengCellTypeRegistrationPin
   ' oCellTypes.Add eengCellTypeRegistrationPinGrid
    'oCellTypes.Add eengCellTypeSkyhook
   ' oCellTypes.Add eengCellTypeSolderPalette
  '  oCellTypes.Add eengCellTypeTestCoupon
    oCellTypes.Add eengCellTypeTestPoint

    oGOFAddEach.CellItemsSide = eengCellSideTop
```

```
    ' Add all cell items individually
    Dim oCellItems
    Set oCellItems = oGOFAddEach.CellItems
    oCellItems.Add eengCellItemAssemblyOutline
    oCellItems.Add eengCellItemAssemblyPartNumber
    oCellItems.Add eengCellItemAssemblyReferenceDesignator
   ' oCellItems.Add eengCellItemCellOrigin
  '  oCellItems.Add eengCellItemGlueSpots
    oCellItems.Add eengCellItemInsertionOutline
    oCellItems.Add eengCellItemPlacementOutline
    oCellItems.Add eengCellItemSilkScreenOutline
    oCellItems.Add eengCellItemSilkScreenPartNumber
    oCellItems.Add eengCellItemSilkScreenReferenceDesignator
   ' oCellItems.Add eengCellItemTestPointObstruct

    ' Add all conductive items individually
    oGOFAddEach.ConductiveLayer = 1
    Dim oConductiveItems
    Set oConductiveItems = oGOFAddEach.ConductiveItems
    oConductiveItems.Add eengCondItemActualPlaneShapes
    oConductiveItems.Add eengCondItemCopperBalancing
    oConductiveItems.Add eengCondItemEtchedText
    oConductiveItems.Add eengCondItemFiducials
    oConductiveItems.Add eengCondItemPartPadsSMD
    oConductiveItems.Add eengCondItemPartPadsTest
    oConductiveItems.Add eengCondItemPartPadsThru
   ' oConductiveItems.Add eengCondItemPinNumbers
    oConductiveItems.Add eengCondItemPlaneData
  '  oConductiveItems.Add eengCondItemPlaneNoConnect
   ' oConductiveItems.Add eengCondItemPlaneObstruct
  '  oConductiveItems.Add eengCondItemPlaneShape
    oConductiveItems.Add eengCondItemResistorAreas
  '  oConductiveItems.Add eengCondItemRouteObstruct
    'oConductiveItems.Add eengCondItemRuleArea
    oConductiveItems.Add eengCondItemTraces
    oConductiveItems.Add eengCondItemViaHoles
    oConductiveItems.Add eengCondItemViaPads


    ' Add all items using "All" constants
    Dim oGOFAddAll
    Set oGOFAddAll = oGerberOutputFiles.Add("AllElements.gdo")
    oGOFAddAll.FlashPads = TRUE
    oGOFAddAll.HeaderText = "Gerber Output File Header Text"
    oGOFAddAll.TrailerText = "Gerber Output File Trailer Text"
    oGOFAddAll.ProcessUnconnectedPads = TRUE
    'oGOFAddAll.DCodeMappingFileName =
 "W:\2005\standard\config\pcb\sample.dmf"
    oGOFAddAll.BoardItems.Add eengBoardItemAll
    oGOFAddAll.CellTypes.Add eengCellTypeAll
    oGOFAddAll.CellItemsSide = eengCellSideTop
    oGOFAddAll.CellItems.Add eengCellItemAll
```

```
    oGOFAddAll.ConductiveLayer = 1
    oGOFAddAll.ConductiveItems.Add eengCondItemAll
End Sub

Public Function GetLicensedDoc(appObj)
    On Error Resume Next
    Dim key, licenseServer, licenseToken, docObj
    Set GetLicensedDoc = Nothing
    ' collect the active document
    Set docObj = appObj.ActiveDocument
    If Err Then
        Call appObj.Gui.StatusBarText("No active document: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' Ask Xpedition Layout's document for the key
    key = docObj.Validate(0)
    ' Get token from license server
    Set licenseServer =
 CreateObject("MGCPCBAutomationLicensing.Application")
    licenseToken = licenseServer.GetToken(key)
    Set licenseServer = Nothing
    ' Ask the document to validate the license token
    Err.Clear
    Call docObj.Validate(licenseToken)
    If Err Then
        Call appObj.Gui.StatusBarText("No active document license: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' everything is OK, return document
    Set GetLicensedDoc = docObj
End Function
```

Table 11. Gerber Object Methods and Properties

Method or Property	Description
AddVariant Method (Gerber Object)	Adds the design variant to the output variant list.
ClearVariants Method (Gerber Object)	Clears the output variant list.
Go Method (Gerber Object)	Generates Gerber data for all output files.
DesignFileName Property (Gerber Object)	Sets the name of the design file, including path.
DesignVariantNames Property (Gerber Object)	Returns the variant names for the current design.
Errors Property (Gerber Object)	Returns the collection of errors.
GenerateMacros Property (Gerber Object)	Specifies whether to use automatically generated macros that utilize an array of shapes and lines to create custom flashes.
IncludeVariantData Property (Gerber Object)	Includes variant data (if it exists) if set to True. If no variant is specified, the property outputs the Master design.

Table 11. Gerber Object Methods and Properties (continued)

Method or Property	Description
MachineFormat Property (Gerber Object)	Returns the Gerber Machine Format object. Refer to the Gerber Output dialog box, Gerber Machine Format file.
OffsetX Property (Gerber Object)	Sets the X offset of the data from the Gerber machine origin. Refer to the Gerber Output dialog box, Parameters tab, Offset from origin option.
OffsetY Property (Gerber Object)	Sets the Y offset of the data from the Gerber machine origin. Refer to the Gerber Output dialog box, Parameters tab, Offset from origin option.
OutputFiles Property (Gerber Object)	Returns the collection of Gerber output files. Refer to the Gerber Output dialog box, Parameters tab, Output files list.
SetupFileName Property (Gerber Object)	Sets the name of the Gerber plot setup file, including path. This is the .gpf file that the Gerber engine uses to process output data and to load the Gerber Output dialog box. Refer to the Gerber Output dialog box, Gerber Plot Setup file.
SpaceBetweenCopiesX Property (Gerber Object)	Sets the space between copies in the X direction. Refer to the Gerber Output dialog box, Parameters tab, Space between origins option.
SpaceBetweenCopiesY Property (Gerber Object)	Sets the space between copies in the Y direction. Refer to the Gerber Output dialog box, Parameters tab, Space between origins option.
XAxisCopies Property (Gerber Object)	Sets the number of copies in the X direction. Refer to the Gerber Output dialog box, Parameters tab, Number of copies per axis option.
YAxisCopies Property (Gerber Object)	Sets the number of copies in the Y direction. Refer to the Gerber Output dialog box, Parameters tab, Number of copies per axis option.

AddVariant Method (Gerber Object)

Object: Gerber Object

Prerequisites: None

Adds the design variant to the output variant list.

Usage

```
Gerber.AddVariant [ByVal variantName As String = “All”])
```

Arguments

- variantName

  (Optional) A string that contains the variant name. If not specified, the method adds all variant names to the output list.

ClearVariants Method (Gerber Object)

Object: Gerber Object

Prerequisites: None

Clears the output variant list.

Usage

```
Gerber.ClearVariants()
```

Arguments

None

Go Method (Gerber Object)

Object: Gerber Object

Prerequisites: None

Generates Gerber data for all output files.

Usage

Gerber.Go ()

Arguments

None

Examples

```
oGerberEngine.Go()
```

DesignFileName Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

Gerber.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the Gerber design file name.

Examples

```
oGerberEngine.DesignFileName = doc.FullName
```

DesignVariantNames Property (Gerber Object)

Object: Gerber Object

Access: Read-Only

Prerequisites: None

Returns the variant names for the current design.

Usage

```
Gerber.DesignVariantNames
```

Arguments

None

Return Values

Array. A array of variant names.

Errors Property (Gerber Object)

Object: Gerber Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

Gerber.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oGerber.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

GenerateMacros Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Specifies whether to use automatically generated macros that utilize an array of shapes and lines to create custom flashes.

Usage

```
Gerber.GenerateMacros = True | False
```

Arguments

None

Return Values

Boolean. If True, to use macros. If False, do not use macros.

IncludeVariantData Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Includes variant data (if it exists) if set to True. If no variant is specified, the property outputs the Master design.

Usage

```
Gerber.IncludeVariantData = True | False
```

Arguments

None

Return Values

Boolean. If True, include variant data. If False, do not include variant data.

MachineFormat Property (Gerber Object)

Object: Gerber Object

Access: Read-Only

Prerequisites: None

Returns the Gerber Machine Format object. Refer to the Gerber Output dialog box, Gerber Machine Format file.

Usage

Gerber.MachineFormat

Arguments

None

Return Values

GerberMachineFormat. Returns a GerberMachineFormat Object

Examples

```
oGerberMachineFormat = oGerberEngine.MachineFormat
```

OffsetX Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Sets the X offset of the data from the Gerber machine origin. Refer to the Gerber Output dialog box, Parameters tab, Offset from origin option.

Usage

Gerber.OffsetX ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  EEnginesUnit Enum. (Optional) If nothing is specified then the current unit is used.

Return Values

Double. A double that defines the X offset of the data from Gerber machine origin.

Examples

```
oGerberEngine.OffsetX = 0
```

OffsetY Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Sets the Y offset of the data from the Gerber machine origin. Refer to the Gerber Output dialog box, Parameters tab, Offset from origin option.

Usage

Gerber.OffsetY ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  EEnginesUnit Enum (Optional) If nothing is specified then the current unit is used.

Return Values

Double. A double that defines theY offset of the data from Gerber machine origin.

Examples

```
oGerberEngine.OffsetY = 0
```

OutputFiles Property (Gerber Object)

Object: Gerber Object

Access: Read-Only

Prerequisites: None

Returns the collection of Gerber output files. Refer to the Gerber Output dialog box, Parameters tab, Output files list.

Usage

Gerber.OutputFiles

Arguments

None

Return Values

GerberOutputFiles. Returns a GerberOutputFiles Collection.

Examples

```
oGerberOutputFiles = oGerberEngine.OutputFiles
```

SetupFileName Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Sets the name of the Gerber plot setup file, including path. This is the *.gpf* file that the Gerber engine uses to process output data and to load the Gerber Output dialog box. Refer to the Gerber Output dialog box, Gerber Plot Setup file.

Usage

Gerber.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the name of the Gerber plot setup file.

Examples

```
oGerberEngine.SetupFileName = “mydesign\pcb\Config\plotsetup.gpf”
```

SpaceBetweenCopiesX Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Sets the space between copies in the X direction. Refer to the Gerber Output dialog box, Parameters tab, Space between origins option.

Usage

Gerber.SpaceBetweenCopiesX ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  EEnginesUnit Enum (Optional) If nothing is specified then the current unit is used.

Return Values

Double. A double that defines the space between copies in the X direction.

Examples

```
oGerberEngine.SpaceBetweenCopiesX = 10
```

SpaceBetweenCopiesY Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Sets the space between copies in the Y direction. Refer to the Gerber Output dialog box, Parameters tab, Space between origins option.

Usage

Gerber.SpaceBetweenCopiesY ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  EEnginesUnit Enum (Optional) If nothing is specified then the current unit is used.

Return Values

Double. A double that defines the space between copies in the Y direction.

Examples

```
oGerberEngine.SpaceBetweenCopiesY = 10
```

XAxisCopies Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Sets the number of copies in the X direction. Refer to the Gerber Output dialog box, Parameters tab, Number of copies per axis option.

Usage

Gerber.XAxisCopies = Long

Arguments

None

Return Values

Long. A long that defines the number of copies in X direction.

Examples

```
oGerberEngine.XAxisCopies = 2
```

YAxisCopies Property (Gerber Object)

Object: Gerber Object

Access: Write-Only

Prerequisites: None

Sets the number of copies in the Y direction. Refer to the Gerber Output dialog box, Parameters tab, Number of copies per axis option.

Usage

Gerber.YAxisCopies = Long

Arguments

None

Return Values

Long. A long that defines the number of copies in Y direction.

Examples

```
oGerberEngine.YAxisCopies = 2
```

GerberMachineFormat Object

The GerberMachineFormat Object contains the information the Gerber Machine Format Engine uses for Gerber formatting.

Refer to the Gerber Machine Format dialog box, by choosing **Setup > Gerber Machine Format**.

See Also Gerber Object.

The following is a example subroutine to set up the Gerber machine format:

```
Sub SetupMachineFormat(oGerberEngine)
    'On Error Resume Next

    Dim oGerberMachineFormat
    Set oGerberMachineFormat = oGerberEngine.MachineFormat

    oGerberMachineFormat.FileName = jobConfigPath + _
  "UserGerberMachineFile.gmf"
    oGerberMachineFormat.DataType = eengGerber274X
    oGerberMachineFormat.DataMode = TRUE
    oGerberMachineFormat.StepMode = eengStepIncremental
    oGerberMachineFormat.DataFormatLeadingDigits = 3
    oGerberMachineFormat.DataFormatTrailingDigits = 2
    oGerberMachineFormat.ZeroTruncation = eengZeroTruncationNone
    oGerberMachineFormat.CharacterSet = eengCharacterSetASCII
    oGerberMachineFormat.ArcStyle = eengArcStyleQuadrant
    oGerberMachineFormat.Delimiter = eengDataDelimiterStar
    oGerberMachineFormat.Comments = TRUE
    oGerberMachineFormat.SequenceNumbering = TRUE
    oGerberMachineFormat.Unit = eengUnitInch
    oGerberMachineFormat.PolygonFillMethod = eengPolygonFillDraw
    oGerberMachineFormat.RecordLength = 0
End Sub
```

Table 12. GerberMachineFormat Object Methods and Properties

Method or Property	Description
ArcStyle Property (GerberMachineFormat Object)	Sets the Gerber data arc style.
CharacterSet Property (GerberMachineFormat Object)	Sets the Gerber data character set.
ClearShapeFillWidths Method (GerberMachineFormat Object)	Clears all shape fill widths.
Comments Property (GerberMachineFormat Object)	Sets the option to include or exclude comments in the output file.
DataFormatLeadingDigits Property (GerberMachineFormat Object)	Sets the leading digits of Gerber data format.
DataFormatTrailingDigits Property (GerberMachineFormat Object)	Sets the trailing digits of Gerber data format.

Table 12. GerberMachineFormat Object Methods and Properties (continued)

Method or Property	Description
DataMode Property (GerberMachineFormat Object)	Sets the Gerber data mode.
DataType Property (GerberMachineFormat Object)	Sets the Gerber data type.
DefaultDrawWidth Property (GerberMachineFormat Object)	Sets the Gerber data default draw width.
Delimiter Property (GerberMachineFormat Object)	Sets the Gerber data delimiter.
FileName Property (GerberMachineFormat Object)	Sets the name of the Gerber machine format file, including path.
PolygonFillMethod Property (GerberMachineFormat Object)	Sets the Gerber polygon shape fill method.
RecordLength Property (GerberMachineFormat Object)	Sets the Gerber data record length.
SequenceNumbering Property (GerberMachineFormat Object)	Sets the option for sequence numbering.
ShapeFillWidths Property (GerberMachineFormat Object)	Appends the list of shape fill widths.
StepMode Property (GerberMachineFormat Object)	Sets the Gerber step mode.
Unit Property (GerberMachineFormat Object)	Sets the Gerber data unit.
ZeroTruncation Property (GerberMachineFormat Object)	Sets the Zero truncation value.

ArcStyle Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber data arc style.

Usage

GerberMachineFormat.ArcStyle = EEnginesArcStyle

Arguments

None

Return Values

EEnginesArcStyle Enum

Examples

```
oGerberMachineFormat.ArcStyle = eengArcStyleQuadrant
```

CharacterSet Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber data character set.

Usage

GerberMachineFormat.CharacterSet = EEnginesCharacterSet

Arguments

None

Return Values

EEnginesCharacterSet Enum

Examples

```
oGerberMachineFormat.CharacterSet = eengCharacterSetASCII
```

ClearShapeFillWidths Method (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Prerequisites: None

Clears all shape fill widths.

Usage

ClearShapeFillWidths()

Arguments

None

Examples

```
ClearShapeFillWidths()
```

Comments Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the option to include or exclude comments in the output file.

Usage

GerberMachineFormat.Comments = True | False

Arguments

None

Return Values

Boolean. True includes comments. False excludes comments.

Examples

```
oGerberMachineFormat.Comments = TRUE
```

DataFormatLeadingDigits Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the leading digits of Gerber data format.

Usage

GerberMachineFormat.DataFormatLeadingDigits = Long

Arguments

None

Return Values

Long. A long that defines number of leading digits of the Gerber data format.

Examples

```
oGerberMachineFormat.DataFormatLeadingDigits = 3
```

DataFormatTrailingDigits Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the trailing digits of Gerber data format.

Usage

GerberMachineFormat.DataFormatTrailingDigits = Long

Arguments

None

Return Values

Long. A long that defines the number of trailing digits of the Gerber data format.

Examples

```
oGerberMachineFormat.DataFormatTrailingDigits = 2
```

DataMode Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber data mode.

Usage

GerberMachineFormat.DataMode = True | False

Arguments

None

Return Values

Boolean. True for modal. False for non-modal.

Examples

```
oGerberMachineFormat.DataMode = TRUE
```

DataType Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber data type.

Usage

GerberMachineFormat.DataType = EEnginesDataType

Arguments

None

Return Values

EEnginesDataType Enum

Examples

```
oGerberMachineFormat.DataType = eengGerber274X
```

DefaultDrawWidth Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber data default draw width.

Usage

```
GerberMachineFormat.DefaultDrawWidth([ByVal eUnit As EEnginesUnit = eengUnitInch]) = Double
```

Arguments

- eUnit

  (Optional) The tunitso use draw width. If nothing is specified, inches are used.

Return Values

Double. A double that defines the default draw width.

Delimiter Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber data delimiter.

Usage

GerberMachineFormat.Delimiter = EEnginesDataDelimiter

Arguments

None

Return Values

EEnginesDataDelimiter Enum

Examples

```
oGerberMachineFormat.Delimiter = eengDataDelimiterStar
```

FileName Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the name of the Gerber machine format file, including path.

Usage

GerberMachineFormat.FileName = String

Arguments

None

Return Values

String. A string that contains the Gerber machine format file (*.gmf*) name.

Examples

```
oGerberMachineFormat.FileName = jobConfigPath + _
 "UserGerberMachineFile.gmf"
```

PolygonFillMethod Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber polygon shape fill method.

Usage

GerberMachineFormat.PolygonFillMethod = EEnginesPolygonFill

Arguments

None

Return Values

EEnginesPolygonFill Enum

Examples

```
oGerberMachineFormat.PolygonFillMethod = eengPolygonFillDraw
```

RecordLength Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber data record length.

Usage

GerberMachineFormat.RecordLength = Long

Arguments

None

Return Values

Long. A long that specifies the Gerber data record length.

Examples

```
oGerberMachineFormat.RecordLength = 0
```

SequenceNumbering Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the option for sequence numbering.

Usage

GerberMachineFormat.SequenceNumbering = True | False

Arguments

None

Return Values

Boolean. True sets sequence numbering on. False turns sequence numbering off.

Examples

```
oGerberMachineFormat.SequenceNumbering = TRUE
```

ShapeFillWidths Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Appends the list of shape fill widths.

Usage

GerberMachineFormat.ShapeFillWidths = EEEnginesUnit

Arguments

None

Return Values

EEnginesUnit Enum

Examples

```
Property Let ShapeFillWidths([ByVal eUnit As EEngi9nesUnit = eengUnitInch])
```

StepMode Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber step mode.

Usage

GerberMachineFormat.StepMode = EEnginesStepMode

Arguments

None

Return Values

EEnginesStepMode Enum

Examples

```
oGerberMachineFormat.StepMode = eengStepIncremental
```

Unit Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Gerber data unit.

Usage

GerberMachineFormat.Unit = EEnginesUnit

Arguments

None

Return Values

EEnginesUnit Enum

Examples

```
oGerberMachineFormat.Unit = eengUnitInch
```

ZeroTruncation Property (GerberMachineFormat Object)

Object: GerberMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Zero truncation value.

Usage

GerberMachineFormat.ZeroTruncation = EEnginesZeroTruncation

Arguments

None

Return Values

Value. EEnginesZeroTruncation Enum

Examples

```
oGerberMachineFormat.ZeroTruncation = eengZeroTruncationNone
```

GerberOutputFile Object

The GerberOutputFile Object contains information to include in the Gerber output file.

For more information, refer to the Gerber Output dialog box, by choosing **Output > Gerber**.

```
Sub SetupOutputFiles(oGerberEngine)
    'On Error Resume Next

    Dim oGerberOutputFiles
    Set oGerberOutputFiles = oGerberEngine.OutputFiles

    Dim oGOFAddEach
    Set oGOFAddEach = oGerberOutputFiles.Add("EachElements.gdo")
    oGOFAddEach.FlashPads = FALSE
    oGOFAddEach.HeaderText = "Gerber Output File Header Text"
    oGOFAddEach.TrailerText = "Gerber Output File Trailer Text"
    oGOFAddEach.ProcessUnconnectedPads = FALSE
    'oGOFAddEach.DCodeMappingFileName = _
  "W:\2005\standard\config\pcb\sample.dmf"

    ' Add all board items individually
    Dim oBoardItems
    Set oBoardItems = oGOFAddEach.BoardItems
    oBoardItems.Add eengBoardItemBoardObstruct
    oBoardItems.Add eengBoardItemBoardOrigin
    oBoardItems.Add eengBoardItemBoardOutline
    oBoardItems.Add eengBoardItemContours
    oBoardItems.Add eengBoardItemDetailViews
    oBoardItems.Add eengBoardItemDocumentation
    oBoardItems.Add eengBoardItemDrcWindow
    oBoardItems.Add eengBoardItemDrillDrawingThru
    oBoardItems.Add eengBoardItemGeneratedSilkscreenBottom
    oBoardItems.Add eengBoardItemGeneratedSilkscreenTop
    oBoardItems.Add eengBoardItemManufacturingOutline
    oBoardItems.Add eengBoardItemMountingHoles
    oBoardItems.Add eengBoardItemNCDrillOrigin
    oBoardItems.Add eengBoardItemPanelBorder
    oBoardItems.Add eengBoardItemPanelHole
    oBoardItems.Add eengBoardItemPanelOrigin
    oBoardItems.Add eengBoardItemPanelOutline
    oBoardItems.Add eengBoardItemPartHoles
    oBoardItems.Add eengBoardItemPlacementObstructBottom
    oBoardItems.Add eengBoardItemPlacementObstructTop
    oBoardItems.Add eengBoardItemRoomBottom
    oBoardItems.Add eengBoardItemRoomTop
    oBoardItems.Add eengBoardItemRouteBorder
    oBoardItems.Add eengBoardItemScoringLine
    oBoardItems.Add eengBoardItemShearingHoles
    oBoardItems.Add eengBoardItemShearingLine
    oBoardItems.Add eengBoardItemSoldermaskBottom
    oBoardItems.Add eengBoardItemSoldermaskTop
```

```
oBoardItems.Add eengBoardItemSolderpasteBottom
oBoardItems.Add eengBoardItemSolderpasteTop
oBoardItems.Add eengBoardItemToolingHoles

' Add all cell types individually
Dim oCellTypes
Set oCellTypes = oGOFAddEach.CellTypes
oCellTypes.Add eengCellTypeBadBoardIdentifier
oCellTypes.Add eengCellTypeBreakAwayTab
oCellTypes.Add eengCellTypeBuried
oCellTypes.Add eengCellTypeConnector
oCellTypes.Add eengCellTypeDiscreteAxial
oCellTypes.Add eengCellTypeDiscreteChip
oCellTypes.Add eengCellTypeDiscreteOther
oCellTypes.Add eengCellTypeDiscreteRadial
oCellTypes.Add eengCellTypeEdgeConnector
oCellTypes.Add eengCellTypeElectricalTestIdentifier
oCellTypes.Add eengCellTypeGeneral
oCellTypes.Add eengCellTypeGeneralPanelCell
oCellTypes.Add eengCellTypeGraphic
oCellTypes.Add eengCellTypeICBareDie
oCellTypes.Add eengCellTypeICBGA
oCellTypes.Add eengCellTypeICDIP
oCellTypes.Add eengCellTypeICFlipChip
oCellTypes.Add eengCellTypeICLCC
oCellTypes.Add eengCellTypeICOther
oCellTypes.Add eengCellTypeICPGA
oCellTypes.Add eengCellTypeICPLCC
oCellTypes.Add eengCellTypeICSIP
oCellTypes.Add eengCellTypeICSOIC
oCellTypes.Add eengCellTypeJumper
oCellTypes.Add eengCellTypeMechanical
oCellTypes.Add eengCellTypePanelIdentifier
oCellTypes.Add eengCellTypePanelStiffener
oCellTypes.Add eengCellTypeRegistrationPin
oCellTypes.Add eengCellTypeRegistrationPinGrid
oCellTypes.Add eengCellTypeSkyhook
oCellTypes.Add eengCellTypeSolderPalette
oCellTypes.Add eengCellTypeTestCoupon
oCellTypes.Add eengCellTypeTestPoint

oGOFAddEach.CellItemsSide = eengCellSideTop

' Add all cell items individually
Dim oCellItems
Set oCellItems = oGOFAddEach.CellItems
oCellItems.Add eengCellItemAssemblyOutline
oCellItems.Add eengCellItemAssemblyPartNumber
oCellItems.Add eengCellItemAssemblyReferenceDesignator
oCellItems.Add eengCellItemCellOrigin
oCellItems.Add eengCellItemGlueSpots
oCellItems.Add eengCellItemInsertionOutline
oCellItems.Add eengCellItemPlacementOutline
```

```
    oCellItems.Add eengCellItemSilkScreenOutline
    oCellItems.Add eengCellItemSilkScreenPartNumber
    oCellItems.Add eengCellItemSilkScreenReferenceDesignator
    oCellItems.Add eengCellItemTestPointObstruct

    ' Add all conductive items individually
    oGOFAddEach.ConductiveLayer = 1
    Dim oConductiveItems
    Set oConductiveItems = oGOFAddEach.ConductiveItems
    oConductiveItems.Add eengCondItemActualPlaneShapes
    oConductiveItems.Add eengCondItemCopperBalancing
    oConductiveItems.Add eengCondItemEtchedText
    oConductiveItems.Add eengCondItemFiducials
    oConductiveItems.Add eengCondItemPartPadsSMD
    oConductiveItems.Add eengCondItemPartPadsTest
    oConductiveItems.Add eengCondItemPartPadsThru
    oConductiveItems.Add eengCondItemPinNumbers
    oConductiveItems.Add eengCondItemPlaneData
    oConductiveItems.Add eengCondItemPlaneNoConnect
    oConductiveItems.Add eengCondItemPlaneObstruct
    oConductiveItems.Add eengCondItemPlaneShape
    oConductiveItems.Add eengCondItemResistorAreas
    oConductiveItems.Add eengCondItemRouteObstruct
    oConductiveItems.Add eengCondItemRuleArea
    oConductiveItems.Add eengCondItemTraces
    oConductiveItems.Add eengCondItemViaHoles
    oConductiveItems.Add eengCondItemViaPads


    ' Add all items using "All" constants
    Dim oGOFAddAll
    Set oGOFAddAll = oGerberOutputFiles.Add("AllElements.gdo")
    oGOFAddAll.FlashPads = TRUE
    oGOFAddAll.HeaderText = "Gerber Output File Header Text"
    oGOFAddAll.TrailerText = "Gerber Output File Trailer Text"
    oGOFAddAll.ProcessUnconnectedPads = TRUE
    'oGOFAddAll.DCodeMappingFileName =
  "W:\2005\standard\config\pcb\sample.dmf"
    oGOFAddAll.BoardItems.Add eengBoardItemAll
    oGOFAddAll.CellTypes.Add eengCellTypeAll
    oGOFAddAll.CellItemsSide = eengCellSideTop
    oGOFAddAll.CellItems.Add eengCellItemAll
    oGOFAddAll.ConductiveLayer = 1
    oGOFAddAll.ConductiveItems.Add eengCondItemAll
End Sub
```

Table 13. GerberOutputFile Object Properties

Property	Description
BoardItems Property (GerberOutputFile Object)	Returns a collection of Board Items. Refer to the Gerber Output dialog box, Contents tab, Board items option.
CellItems Property (GerberOutputFile Object)	Returns a collection of Cell Items. Refer to the Gerber Output dialog box, Contents tab, Cells, Items option.

Table 13. GerberOutputFile Object Properties (continued)

Property	Description
CellItemsLayer Property (GerberOutputFile Object)	Sets the layer of the board for cell fabrication data items. Refer to the Gerber Output dialog box, Contents tab, Cells, Items, Layer option.
CellTypes Property (GerberOutputFile Object)	Returns a collection of Cell Types. Refer to the Gerber Output dialog box, Contents tab, Cells, Types option.
ConductiveItems Property (GerberOutputFile Object)	Returns a collection of Conductive Items. Refer to the Gerber Output dialog box, Contents tab, Conductors, Items option.
ConductiveLayer Property (GerberOutputFile Object)	Sets the conductive layer for the Gerber output file. Refer to the Gerber Output dialog box, Contents tab, Conductors, Layer option.
DCodeMappingFileName Property (GerberOutputFile Object)	Sets the name of the D-Code Mapping file for the Gerber output file. Refer to the Gerber Output dialog box, Parameters tab, Output files, D-Code Mapping File column.
FlashPads Property (GerberOutputFile Object)	Sets the option to flash pads or to draw pads. Used for Gerber data types 274D and 274-X. Refer to the Gerber Output dialog box, Parameters tab, Output files, Flash/Draw Pads column (274D and 274-X data types only).
HeaderText Property (GerberOutputFile Object)	Sets the header text for the Gerber output file. Refer to the Gerber Output dialog box, Parameters tab, Header text option.
Name Property (GerberOutputFile Object)	Returns the Gerber output file name. Refer to the Gerber Output dialog box, Parameters tab, File Name column.
ProcessUnconnectedPads Property (GerberOutputFile Object)	Sets the option to process unconnected pads for conductive items. Refer to the Gerber Output dialog box, Contents tab, Only process pads that are connected by a trace or plane.
TrailerText Property (GerberOutputFile Object)	Sets trailer text for Gerber output file. Refer to the Gerber Output dialog box, Parameters tab, Trailer text option.
UserLayers Property (GerberOutputFile Object)	Returns a collection of user layers. Refer to the Gerber Output dialog box, Contents tab, User-defined layers option.

BoardItems Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Read-Only

Prerequisites: None

Returns a collection of Board Items. Refer to the Gerber Output dialog box, Contents tab, Board items option.

Usage

GerberOutputFile.BoardItems

Arguments

None

Return Values

BoardItems. Returns a collection of EEnginesBoardItem Enums.

Examples

```
oBoardItems = oGOFAddEach.BoardItems
```

CellItems Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Read-Only

Prerequisites: None

Returns a collection of Cell Items. Refer to the Gerber Output dialog box, Contents tab, Cells, Items option.

Usage

GerberOutputFile.CellItems

Arguments

None

Return Values

CellItems. Returns a collection of EEnginesCellItem Enums.

Examples

```
oCellItems = oGOFAddEach.CellItems
```

CellItemsLayer Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Write-Only

Prerequisites: None

Sets the layer of the board for cell fabrication data items. Refer to the Gerber Output dialog box, Contents tab, Cells, Items, Layer option.

Usage

GerberOutputFile.CellItemsLayer = Long

Arguments

None

Return Values

Long. A long that contains the layer number of the board for cell fabrication data items.

Examples

```
cellItems.CellItemsLayer = 1
```

CellTypes Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Read-Only

Prerequisites: None

Returns a collection of Cell Types. Refer to the Gerber Output dialog box, Contents tab, Cells, Types option.

Usage

GerberOutputFile.CellTypes

Arguments

None

Return Values

CellTypes. Returns a collection of EEnginesCellType Enums.

Examples

```
oCellTypes = oGOFAddEach.CellTypes
```

ConductiveItems Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Read-Only

Prerequisites: None

Returns a collection of Conductive Items. Refer to the Gerber Output dialog box, Contents tab, Conductors, Items option.

Usage

GerberOutputFile.ConductiveItems

Arguments

None

Return Values

ConductiveItems. Returns a collection of EEnginesConductiveItem Enums.

Examples

```
oConductiveItems = oGOFAddEach.ConductiveItems
```

ConductiveLayer Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Write-Only

Prerequisites: None

Sets the conductive layer for the Gerber output file. Refer to the Gerber Output dialog box, Contents tab, Conductors, Layer option.

Usage

GerberOutputFile.ConductiveLayer = Value

Arguments

None

Return Values

Value. The conductive layer for this Gerber output file.

Examples

```
oGOFAddAll.ConductiveLayer = 1
```

DCodeMappingFileName Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Write-Only

Prerequisites: None

Sets the name of the D-Code Mapping file for the Gerber output file. Refer to the Gerber Output dialog box, Parameters tab, Output files, D-Code Mapping File column.

Usage

GerberOutputFile.DCodeMappingFileName = String

Arguments

None

Return Values

String. A string that contains the D-Code mapping file name.

Examples

```
oGOFAddEach.DCodeMappingFileName = "W:\2005\standard\config\pcb\sample.dmf"
```

FlashPads Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Write-Only

Prerequisites: None

Sets the option to flash pads or to draw pads. Used for Gerber data types 274D and 274-X. Refer to the Gerber Output dialog box, Parameters tab, Output files, Flash/Draw Pads column (274D and 274-X data types only).

Usage

GerberOutputFile.FlashPads = True | False

Arguments

None

Return Values

Boolean. Set to True to flash pads. Set to False to draw pads. The default is False.

Examples

```
oGOFAddEach.FlashPads = FALSE
```

HeaderText Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Write-Only

Prerequisites: None

Sets the header text for the Gerber output file. Refer to the Gerber Output dialog box, Parameters tab, Header text option.

Usage

GerberOutputFile.HeaderText = String

Arguments

None

Return Values

String. Set the header text for the Gerber output file.

Examples

```
oGOFAddEach.HeaderText = "Gerber Output File Header Text"
```

Name Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Read-Only

Prerequisites: None

Returns the Gerber output file name. Refer to the Gerber Output dialog box, Parameters tab, File Name column.

Usage

GerberOutputFile.Name

Arguments

None

Return Values

String. Returns the name of the Gerber output file.

Examples

```
fileName = oGOFAddEach.Name
```

ProcessUnconnectedPads Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Write-Only

Prerequisites: None

Sets the option to process unconnected pads for conductive items. Refer to the Gerber Output dialog box, Contents tab, Only process pads that are connected by a trace or plane.

Usage

GerberOutputFile.ProcessUnconnectedPads = True | False

Arguments

None

Return Values

Boolean. Set True to only process pads that are connected by a trace or plane. Set False to process all pads.

Examples

```
oGOFAddEach.ProcessUnconnectedPads = FALSE
```

TrailerText Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Write-Only

Prerequisites: None

Sets trailer text for Gerber output file. Refer to the Gerber Output dialog box, Parameters tab, Trailer text option.

Usage

GerberOutputFile.TrailerText = String

Arguments

None

Return Values

String. The trailer text for the selected Gerber output file.

Examples

```
oGOFAddEach.TrailerText = "Gerber Output File Trailer Text"
```

UserLayers Property (GerberOutputFile Object)

Object: GerberOutputFile Object

Access: Read-Only

Prerequisites: None

Returns a collection of user layers. Refer to the Gerber Output dialog box, Contents tab, User-defined layers option.

Usage

GerberOutputFile.UserLayers

Arguments

None

Return Values

UserLayers. Returns a UserLayers Collection.

Examples

```
userLayerColl = oGOFAddEach.UserLayers
```

IDFExport Object

The Intermediate Data Format (IDF) allows the exchange of mechanical information between the PCB environment and the mechanical design environment. The IDFExport object supports exporting data to the mechanical environment.

The oIDFImport object (defined in the *Layout Automation Reference (Part 4)*) supports importing data into the PCB environment. For more information, refer to the Export IDF dialog box by choosing **File > Export > IDF**.

Figure 8. IDFExport Data Model

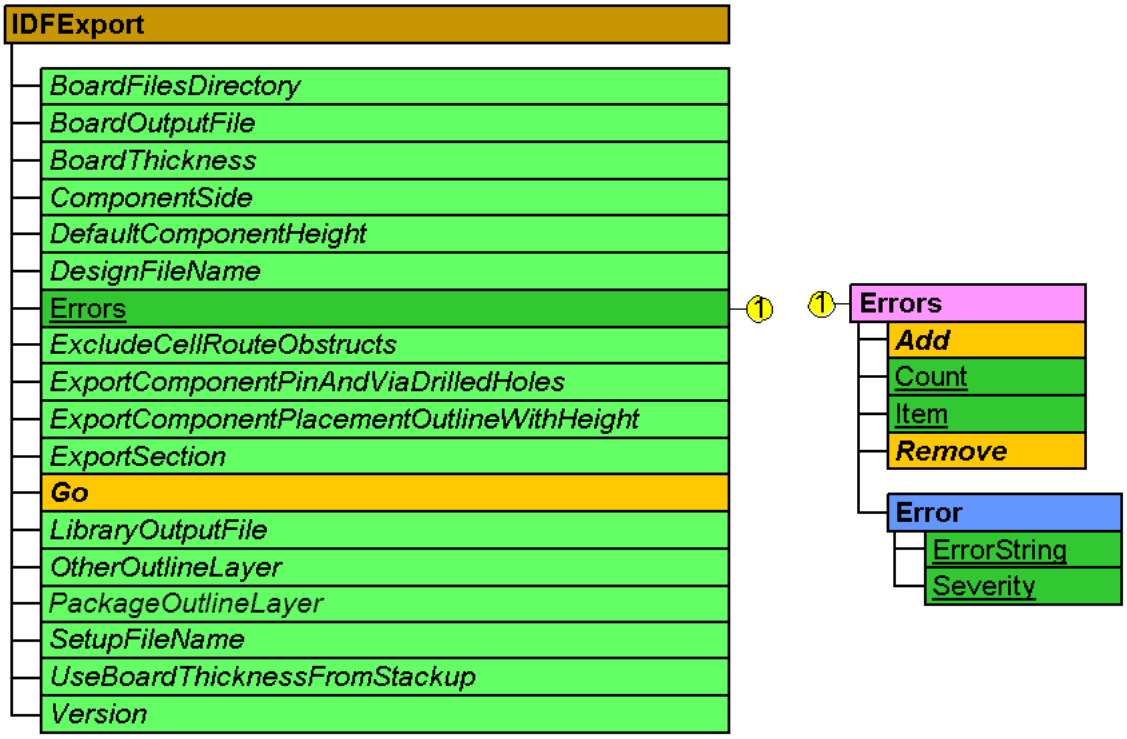


Table 14. IDFExport Object Methods and Properties

Method or Property	Description
Go Method (IDFExport Object)	Generates IDF output.
BoardFilesDirectory Property (IDFExport Object)	Sets the board files directory. (For panel only.) Refer to the Output Files tab, Board files directory option of the IDF Export dialog box.
BoardOutputFile Property (IDFExport Object)	Sets the file and path for the board output file. Refer to the Output files, Board option in the Export IDF dialog box.
BoardThickness Property (IDFExport Object)	Sets the board thickness. Using this property automatically sets the UseBoardThicknessFromStackup property to False. Refer to the Board thickness option in the Export IDF dialog box.
ComponentSide Property (IDFExport Object)	Sets the components to export by the mountside. (For panel only.) Use the ExportSection property to set the option to

Table 14. IDFExport Object Methods and Properties (continued)

Method or Property	Description
	export components. Refer to the Options tab, Components option of the IDF Export dialog box.
DefaultComponentHeight Property (IDFExport Object)	Sets the default component height. Refer to the Default component height option in the Export IDF dialog box.
DesignFileName Property (IDFExport Object)	Sets the name of the design file, including path.
Errors Property (IDFExport Object)	Returns the collection of errors.
ExcludeCellRouteObstructs Property (IDFExport Object)	Excludes the cell route obstructs. This is only used when route obstructs are exported. Use the ExportSection property to set the option to Export Route Obstructs. Refer to the Exclude Cell Route Obstructs option in the Export IDF dialog box.
ExportComponentPinAndViaDrilledHoles Property (IDFExport Object)	Sets the option to export the component pin and via-drilled holes. Refer to the Export Component pin and Via drilled holes option in the Export IDF dialog box.
ExportComponentPlacementOutlineWithHeight Property (IDFExport Object)	Sets the option to export the component placement outline with height. Refer to the Export IDF dialog box.
ExportSection Property (IDFExport Object)	Sets the items to export. Refer to the Options tab, Export Sections options.
LibraryOutputFile Property (IDFExport Object)	Sets the file and path for the library output file. Refer to the Library option in the Export IDF dialog box.
OtherOutlineLayer Property (IDFExport Object)	Sets the Other outline layer to export. (For panel only.) Use the ExportSection property to set the option to export other outline. Refer to the Options tab, Other outline layer option.
PackageOutlineLayer Property (IDFExport Object)	Sets the package outline layer.
SetupFileName Property (IDFExport Object)	Sets the name of the IDF setup file, including path.
UseBoardThicknessFromStackup Property (IDFExport Object)	Sets the option to use the board thickness defined by the layer stackup. Refer to the Use Board thickness from layer stackup option in the Export IDF dialog box.
Version Property (IDFExport Object)	Sets the IDF version. Refer to the Version options in the Export IDF dialog box.

Go Method (IDFExport Object)

Object: IDFExport Object

Prerequisites: None

Generates IDF output.

Usage

IDFExport.Go()

Arguments

None

Examples

```
oIDFExportEngine.Go
```

BoardFilesDirectory Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the board files directory. (For panel only.) Refer to the Output Files tab, Board files directory option of the IDF Export dialog box.

Usage

IDFExport.BoardFilesDirectory = String

Arguments

None

Return Values

String. A string that contains the board files directory.

Examples

```
oIDFExportEngine.BoardFilesDirectory = BoardFilesDir
```

BoardOutputFile Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the file and path for the board output file. Refer to the Output files, Board option in the Export IDF dialog box.

Usage

IDFExport.BoardOutputFile = String

Arguments

None

Return Values

String. A string that contains the name of the board output file, including path.

Examples

```
oIDFExportEngine.BoardOutputFile = boardFile
```

BoardThickness Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the board thickness. Using this property automatically sets the UseBoardThicknessFromStackup property to False. Refer to the Board thickness option in the Export IDF dialog box.

Usage

IDFExport.BoardThickness ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  (Optional) The tunitso use for the board thickness. If nothing is specified the current units are used.

Return Values

Double. A double that contains the board thickness.

Examples

```
oIDFExportEngine.BoardThickness = 85
```

ComponentSide Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the components to export by the mountside. (For panel only.) Use the ExportSection property to set the option to export components. Refer to the Options tab, Components option of the IDF Export dialog box.

Usage

IDFExport.ComponentSide = EEnginesIDFExportComponentSideType

Arguments

None

Return Values

EEnginesIDFExportComponentSideType. The component side type to export. ()

Examples

```
oIDFExportEngine.ExportSection (eengComponents) = True
oIDFExportEngine.ComponentSide = eengComponentTopAndBottom
```

DefaultComponentHeight Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the default component height. Refer to the Default component height option in the Export IDF dialog box.

Usage

IDFExport.DefaultComponentHeight ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  (Optional) The tunitso use for the component height. If nothing is specified the current units are used.

Return Values

Double. A double that contains the default component height.

Examples

```
oIDFExportEngine.DefaultComponentHeight = 10
```

DesignFileName Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

IDFExport.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the name of the design file. (*.pcb* or *.pnl*)

Examples

```
oIDFExportEngine.DesignFileName = jobName
```

Errors Property (IDFExport Object)

Object: IDFExport Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

IDFExport.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oIDFExportEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

ExcludeCellRouteObstructs Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Excludes the cell route obstructs. This is only used when route obstructs are exported. Use the ExportSection property to set the option to Export Route Obstructs. Refer to the Exclude Cell Route Obstructs option in the Export IDF dialog box.

Usage

IDFExport.ExcludeCellRouteObstructs = True | False

Arguments

None

Return Values

Boolean. If True, exclude cell route obstructs. If False, include cell route obstructs.

Examples

```
oIDFExportEngine.ExportSection (eengRouteObstruct) = True
oIDFExportEngine.ExcludeCellRouteObstructs = True
```

ExportComponentPinAndViaDrilledHoles Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the option to export the component pin and via-drilled holes. Refer to the Export Component pin and Via drilled holes option in the Export IDF dialog box.

Usage

IDFExport.ExportComponentPinAndViaDrilledHoles = True | False

Arguments

None

Return Values

Boolean. If True, export the component pin and via drilled holes. If False, do not export the items.

Examples

```
oIDFExportEngine.ExportComponentPinAndViaDrilledHoles = True
```

ExportComponentPlacementOutlineWithHeight Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the option to export the component placement outline with height. Refer to the Export IDF dialog box.

Usage

IDFExport.ExportComponentPlacementOutlineWithHeight = True | False

Arguments

None

Return Values

Boolean. If True, export the component placement outline with height. If False, do not export the item.

Examples

```
oIDFExportEngine.ExportComponentPlacementOutlineWithHeight = True
```

ExportSection Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the items to export. Refer to the Options tab, Export Sections options.

Usage

IDFExport.ExportSection (ByVal eSection As EEnginesIDFExportSectionsType) = True | False

Arguments

- eSection

  The items () EEnginesIDFExportSectionsTypeto export.

Return Values

Boolean. If True, export the specified item. If False, do not export the specified item.

Examples

```
oIDFExportEngine.ExportSection (eengPanelOutline) = True
oIDFExportEngine.ExportSection (eengPanelBorder) = True
oIDFExportEngine.ExportSection (eengPanelObstruct) = True
oIDFExportEngine.ExportSection (eengRouteObstruct) = True
oIDFExportEngine.ExportSection (eengOtherOutline) = True
oIDFExportEngine.OtherOutlineLayer = “MISC_DIMENSION”
oIDFExportEngine.ExportSection (eengDrillHoles) = True
oIDFExportEngine.ExportSection (eengComponents) = True
oIDFExportEngine.ComponentSide = eengComponentTopAndBottom
```

LibraryOutputFile Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the file and path for the library output file. Refer to the Library option in the Export IDF dialog box.

Usage

IDFExport.LibraryOutputFile = String

Arguments

None

Return Values

String. A string that contains the name of the library output file, including path.

Examples

```
oIDFExportEngine.LibraryOutputFile = libraryFile
```

OtherOutlineLayer Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the Other outline layer to export. (For panel only.) Use the ExportSection property to set the option to export other outline. Refer to the Options tab, Other outline layer option.

Usage

IDFExport.OtherOutlineLayer = String

Arguments

None

Return Values

String. A string that contains the layer name.

Examples

```
oIDFExportEngine.ExportSection (eengOtherOutline) = True
oIDFExportEngine.OtherOutlineLayer = “MISC_DIMENSION”
```

PackageOutlineLayer Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the package outline layer.

Usage

IDFExport.PackageOutlineLayer As EEnginesIDFExportPackageOutlineType

Arguments

None

Return Values

EEnginesIDFExportPackageOutlineType Enum

SetupFileName Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the name of the IDF setup file, including path.

Usage

IDFExport.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the name of the setup file.

Examples

```
oIDFExportEngine.SetupFileName = SetupFile
```

UseBoardThicknessFromStackup Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the option to use the board thickness defined by the layer stackup. Refer to the Use Board thickness from layer stackup option in the Export IDF dialog box.

Usage

IDFExport.UseBoardThicknessFromStackup = True | False

Arguments

None

Return Values

Boolean. If True, use the board thickness from stackup. If False, use the defined Board thickness.

Examples

```
oIDFExportEngine.UseBoardThicknessFromStackup = True
```

Version Property (IDFExport Object)

Object: IDFExport Object

Access: Write-Only

Prerequisites: None

Sets the IDF version. Refer to the Version options in the Export IDF dialog box.

Usage

IDFExport.Version = EEnginesIDFVersionType

Arguments

None

Return Values

EEnginesIDFVersionType. The IDF version to output (“EEnginesIDFVersionType” on page 558)

Examples

```
oIDFExportEngine.Version = eengVersion3
```

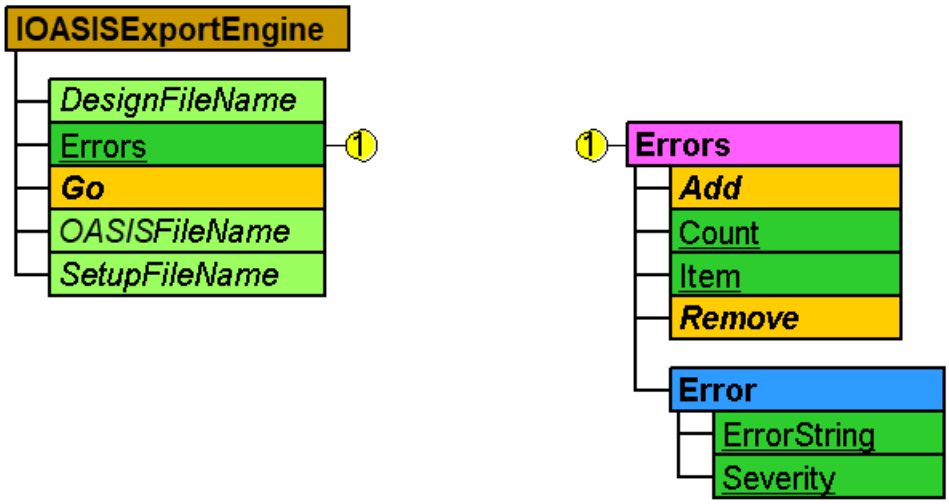
IOASISExportEngine Object

The IOASISExportEngine object contains the information that the OASIS export engine uses for OASIS output.

For more information, refer to the OASIS Export dialog box by choosing **File > Export > OASIS**

The following figure shows the IOASISExportEngine Object Data Model.

Figure 9. IOASISExportEngine Object Data Model



DesignFileName Property (IOASISExportEngine Object)
Errors Property (IOASISExportEngine Object)
Go Method (IOASISExportEngine Object)
OASISFileName Property (IOASISExportEngine Object)
SetupFileName Property (IOASISExportEngine Object)

DesignFileName Property (IOASISExportEngine Object)

Object: IOASISExportEngine Object

Access: Write-Only

Prerequisites: None

Sets the design file name, including path.

Usage

```
IOASISExportEngine.DesignFileName = String
```

Arguments

None

Return Values

String. A string that contains the design file name, including path.

Errors Property (IOASISExportEngine Object)

IOASISExportEngine Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

```
IOASISExportEngine.Errors = Errors
```

Arguments

None

Return Values

IErrors. The collection of errors (Errors Collection).

Go Method (IOASISExportEngine Object)

Object: IOASISExportEngine Object

Prerequisites: None

Runs the OASIS export engine based on the options defined in the OASIS configuration file.

Usage

```
IOASISExportEngine.Go()
```

Arguments

None

Return Values

Nothing

OASISFileName Property (IOASISExportEngine Object)

Object: IOASISExportEngine Object

Access: Write-Only

Prerequisites: None

Sets the OASIS file name, including path.

Usage

```
IOASISExportEngine.OASISFileName = String
```

Arguments

None

Return Values

String. A string that contains the OASIS file name, including path.

SetupFileName Property (IOASISExportEngine Object)

Object: IOASISExportEngine Object

Access: Write-Only

Prerequisites: None

Sets the OASIS configuration file name, including path.

Usage

```
IOASISExportEngine.SetupFileName = String
```

Arguments

None

Return Values

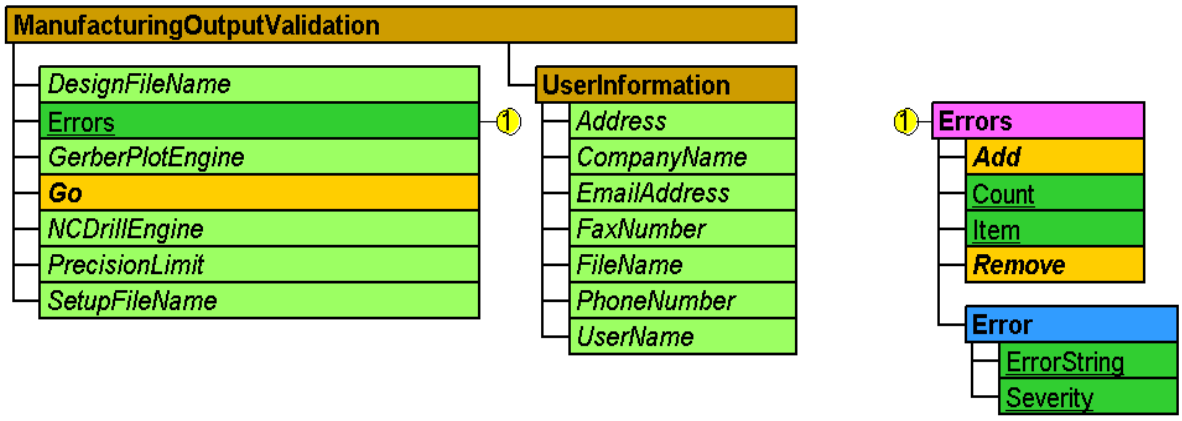
String. A string that contains the OASIS configuration file name, including path.

ManufacturingOutputValidation Object

The ManufacturingOutputValidation Object contains the information the Manufacturing Output Validation Engine uses to process data for output.

For more information, refer to the Manufacturing Output Validation dialog box by choosing **Output > Manufacturing Output Validation**.

Figure 10. ManufacturingOutputValidation Data Model



The following example shows a script sample that runs batch MOV.

Find the complete script at this location:

SDD_HOME\standard\examples\pcb\Automation\AutoProEngines\RunMOV.vbs

Example 6. Batch MOV Example Script

```
Option Explicit

Dim args, app, docObj, jobName, jobConfigPath
Set args = ScriptHelper.Arguments
If args.Count = 3 Then
    ' Call from command line
    ' mgcscript RunMOV.vbs [job name]
    jobName = args.item(3)
Else
    ' Call from Xpedition Layout
    Set app = GetObject(, "MGCPCB.ExpeditionPCBApplication")
    Set docObj = GetLicensedDoc(app)
    jobName = docObj.FullName
End If

Scripting.AddTypeLibrary("MGCPCBEngines.ManufacturingOutputValidation")
```

```
Call RunMOV()

MsgBox "fini", VBInformation, "RunMOV.vbs"

'*********************************************************************
Sub RunMOV()
    'On Error Resume Next

    ' Create MOV Engine object
    Dim oMOVEngine
    Set oMOVEngine =
 CreateObject("MGCPCBEngines.ManufacturingOutputValidation")

    ' Set the design file name
    oMOVEngine.DesignFileName = jobName

    ' Get design Config directory
    Dim pcbPath
    pcbPath = Split(jobName, "PCB")
    jobConfigPath = pcbPath(0) + "PCB/Config/"

    Dim rtn
    rtn = MsgBox("Using existing setup files?", VBYesNo + VBQuestion,
 "RunMOV.vbs")

    If rtn = VBYes Then
        oMOVEngine.SetupFileName = jobConfigPath + "MovSetupFile.MFD"
        oMOVEngine.UserInformation.FileName = jobConfigPath + "MovUser.HKP"
    Else
        oMOVEngine.PrecisionLimit = 1.0
        Call SetupGerber(oMOVEngine)
        Call SetupNCDrill(oMOVEngine)
        Call SetupUserInformation(oMOVEngine)
    End If

    ' Run the MOV Engine
    oMOVEngine.Go

    ' Check errors
    Dim oErrors
    Set oErrors = oMOVEngine.Errors
End Sub

Sub SetupGerber(oMOVEngine)
    'On Error Resume Next

    ' Create Gerber Engine object
    Dim oGerberEngine
    Set oGerberEngine = CreateObject("MGCPCBEngines.Gerber")

    oGerberEngine.DesignFileName = jobName
    oGerberEngine.SetupFileName = jobConfigPath + "PlotSetup.gpf"
    oGerberEngine.Go
```

```
    Set oMOVEngine.GerberPlotEngine = oGerberEngine
End Sub

Sub SetupNCDrill(oMOVEngine)
    'On Error Resume Next

    ' Create NCDrill Engine object
    Dim oNCDrillEngine
    Set oNCDrillEngine = CreateObject("MGCPCBEngines.NCDrill")

    oNCDrillEngine.DesignFileName = jobName
    oNCDrillEngine.SetupFileName = jobConfigPath + "DrillPreferences.hkp"
    oNCDrillEngine.Go

    Set oMOVEngine.NCDrillEngine = oNCDrillEngine
End Sub

Sub SetupUserInformation(oMOVEngine)
    On Error Resume Next

    Dim oUserInfo
    Set oUserInfo = oMOVEngine.UserInformation

    ' Optional if you set other UserInfomation properties.
    oUserInfo.FileName = jobConfigPath + "MovUser.HKP"
    oUserInfo.UserName = "MOV User"
    oUserInfo.CompanyName = "Mentor Graphics"
    oUserInfo.Address = "5030 Bradford Drive"
    oUserInfo.Address = "Building 1, Suite 130"
    oUserInfo.Address = "Huntsville, AL 35805"
    oUserInfo.PhoneNumber = "(256)864-0000"
    oUserInfo.FaxNumber = "(256)864-0001"
    oUserInfo.EmailAddress = "mov_user@mentor.com"
End Sub

Public Function GetLicensedDoc(appObj)
    On Error Resume Next
    Dim key, licenseServer, licenseToken, docObj
    Set GetLicensedDoc = Nothing
    ' collect the active document
    Set docObj = appObj.ActiveDocument
    If Err Then
        Call appObj.Gui.StatusBarText("No active document: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' Ask Xpedition Layout’s document for the key
    key = docObj.Validate(0)
    ' Get token from license server
    Set licenseServer =
 CreateObject("MGCPCBAutomationLicensing.Application")
    licenseToken = licenseServer.GetToken(key)
    Set licenseServer = Nothing
```

```
    ' Ask the document to validate the license token
    Err.Clear
    Call docObj.Validate(licenseToken)
    If Err Then
        Call appObj.Gui.StatusBarText("No active document license: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' everything is OK, return document
    Set GetLicensedDoc = docObj
End Function
```

Table 15. ManufacturingOutputValidation Object Methods and Properties

Method or Property	Description
Go Method (ManufacturingOutputValidation Object)	Runs Manufacturing Output Validation engine.
DesignFileName Property (ManufacturingOutputValidation Object)	Sets the name of the design file, including path.
Errors Property (ManufacturingOutputValidation Object)	Returns the collection of errors.
GerberPlotEngine Property (ManufacturingOutputValidation Object)	Sets the Gerber Plot Engine interface as input.
NCDrillEngine Property (ManufacturingOutputValidation Object)	Sets the NCDrill Engine interface as input.
PrecisionLimit Property (ManufacturingOutputValidation Object)	Sets the precision limit and enables on the generation of precision errors. Refer to the Validation option, Precision errors in the Manufacturing Output Validation dialog box.
SetupFileName Property (ManufacturingOutputValidation Object)	Sets the name of the MOV setup file, including path. This is the MovSetupFile.MFD file that the MOV engine uses to process output data and to load the dialog box.
UserInformation Property (ManufacturingOutputValidation Object)	Returns the user information object. Refer to the MOV user information dialog box, by clicking the Validate Checked Output Files button.

Go Method (ManufacturingOutputValidation Object)

Object: ManufacturingOutputValidation Object

Prerequisites: None

Runs Manufacturing Output Validation engine.

Usage

ManufacturingOutputValidation.Go ()

Arguments

None

Examples

```
oMOVEngine.Go()
```

DesignFileName Property (ManufacturingOutputValidation Object)

Object: ManufacturingOutputValidation Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

ManufacturingOutputValidation.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the name of the design.

Examples

```
oMOVEngine.DesignFileName = jobName
```

Errors Property (ManufacturingOutputValidation Object)

Object: ManufacturingOutputValidation Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

ManufacturingOutputValidation.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oMOVEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

GerberPlotEngine Property (ManufacturingOutputValidation Object)

Object: ManufacturingOutputValidation Object

Access: Write-Only

Prerequisites: None

Sets the Gerber Plot Engine interface as input.

Usage

ManufacturingOutputValidation.GerberPlotEngine = Gerber

Arguments

None

Return Values

Gerber. A Gerber Object.

Examples

```
Set oMOVEngine.Gerber = GerberObj
```

NCDrillEngine Property (ManufacturingOutputValidation Object)

Object: ManufacturingOutputValidation Object

Access: Write-Only

Prerequisites: None

Sets the NCDrill Engine interface as input.

Usage

ManufacturingOutputValidation.NCDrillEngine = NCDrill

Arguments

None

Return Values

NCDrill. An NCDrill Object.

Examples

```
Set oMOVEngine.NCDrillEngine = oNCDrillEngine
```

PrecisionLimit Property (ManufacturingOutputValidation Object)

Object: ManufacturingOutputValidation Object

Access: Write-Only

Prerequisites: None

Sets the precision limit and enables on the generation of precision errors. Refer to the Validation option, Precision errors in the Manufacturing Output Validation dialog box.

Usage

ManufacturingOutputValidation.PrecisionLimit([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  EEnginesUnit Enum. (Optional) Specifies the units to use to for the precision limit If nothing is specified then the current unit is used.

Return Values

Double. A double that specifies the precision limit.

Examples

```
oMOVEngine.PrecisionLimit = .1
```

SetupFileName Property (ManufacturingOutputValidation Object)

Object: ManufacturingOutputValidation Object

Access: Write-Only

Prerequisites: None

Sets the name of the MOV setup file, including path. This is the *MovSetupFile.MFD* file that the MOV engine uses to process output data and to load the dialog box.

Usage

ManufacturingOutputValidation.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the MOV setup file name.

Examples

```
oMOVEngine.SetupFileName = jobConfigPath + "MovSetupFile.MFD"
```

UserInformation Property (ManufacturingOutputValidation Object)

Object: ManufacturingOutputValidation Object

Access: Read-Only

Prerequisites: None

Returns the user information object. Refer to the MOV user information dialog box, by clicking the Validate Checked Output Files button.

Usage

ManufacturingOutputValidation.UserInformation

Arguments

None

Return Values

UserInformation. A UserInformation Object

Examples

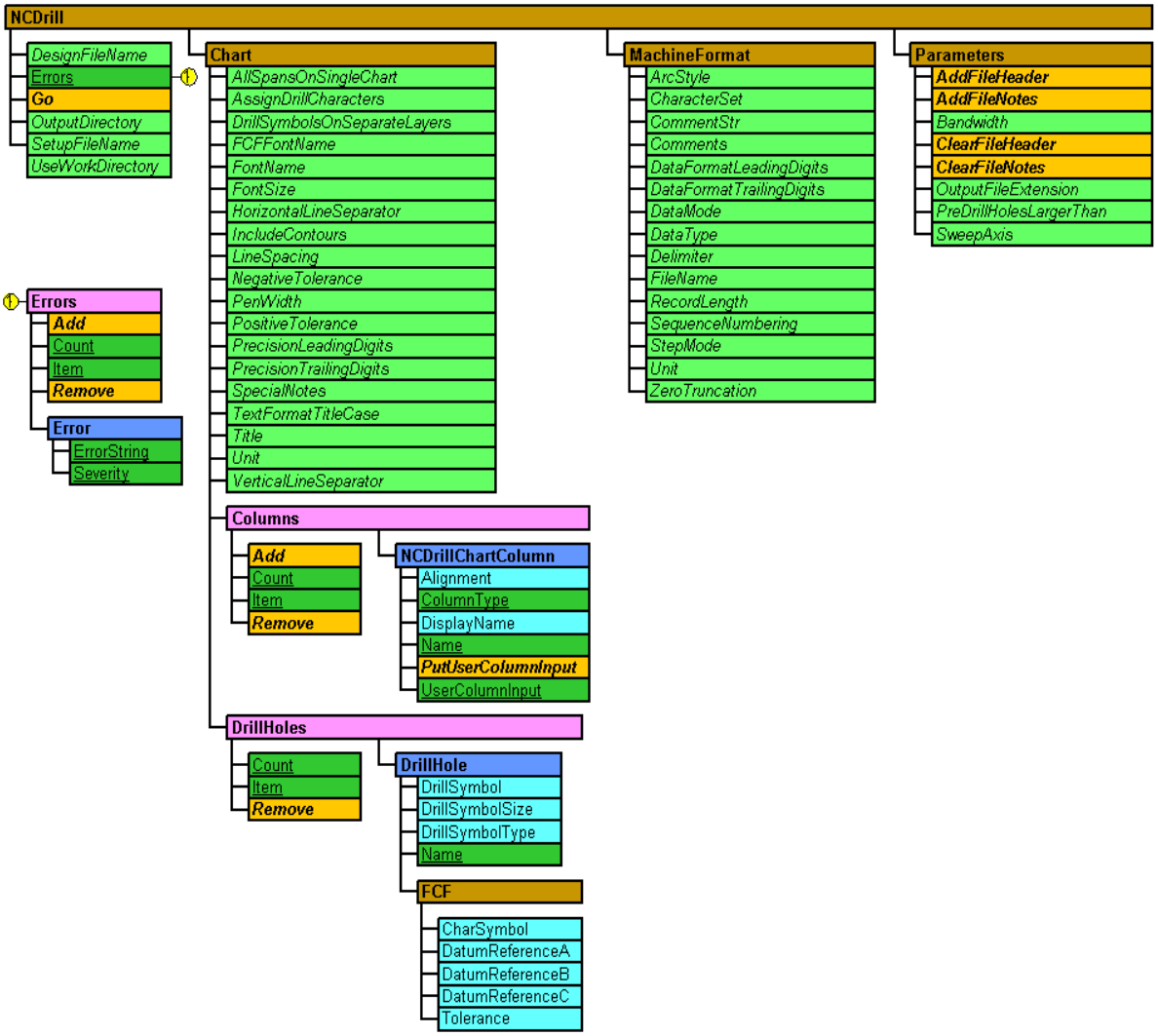
```
oUserInfo = oMOVEngine.UserInformation
```

NCDrill Object

The NCDrill Object contains the information the NCDrill Engine uses to generate numerically-controlled (NC) Drill data.

For more information, refer to the NC Drill Generation dialog box, by choosing **Output > NC Drill**.

Figure 11. NCDrill Data Model



The following example shows a script sample that runs batch NC Drill generation.

Find the complete script at this location:

SDD_HOME\standard\examples\pcb\Automation\AutoProEngines\RunNCDrill.vbs

Example 7. NC Drill Generation Example Script

```
Option Explicit

Dim args, app, docObj, jobName, jobConfigPath
Set args = ScriptHelper.Arguments
If args.Count = 3 Then
    ' Call from command line
    ' mgcscript RunNCDrill.vbs [job name]
    jobName = args.item(3)
Else
    ' Call from Xpedition Layout
    Set app = GetObject(, "MGCPCB.ExpeditionPCBApplication")
    Set docObj = GetLicensedDoc(app)
    jobName = docObj.FullName
End If

Scripting.AddTypeLibrary("MGCPCBEngines.NCDrill")

Call RunNCDrill()

MsgBox "fini", VBInformation, "RunNCDrill.vbs"

'*********************************************************************
***
Sub RunNCDrill()
    'On Error Resume Next

    ' Create NCDrill Engine object
    Dim oNCDrillEngine
    Set oNCDrillEngine = CreateObject("MGCPCBEngines.NCDrill")

    ' Set the design file name
    oNCDrillEngine.DesignFileName = jobName

    ' Get design Config directory
    Dim pcbPath
    pcbPath = Split(jobName, "PCB")
    jobConfigPath = pcbPath(0) + "PCB/Config/"

    Dim rtn
    rtn = MsgBox("Using existing setup files?", VBYesNo + VBQuestion,
 "RunNCDrill.vbs")

    If rtn = VBYes Then
        oNCDrillEngine.SetupFileName = jobConfigPath +
 "DrillPreferences.hkp"
    Else
        Call SetupMachineFormat(oNCDrillEngine)
        Call SetupParameters(oNCDrillEngine)
        Call SetupChart(oNCDrillEngine)
    End If
```

```
    ' Run the NCDrill Engine
    oNCDrillEngine.Go

    ' Check errors
    Dim oErrors
    Set oErrors = oNCDrillEngine.Errors
End Sub

Sub SetupMachineFormat(oNCDrillEngine)
    'On Error Resume Next

    Dim oNCDrillMF
    Set oNCDrillMF = oNCDrillEngine.MachineFormat

    Dim rtn
    rtn = MsgBox("English or Metric" & vbCrLf & "(Yes for English, No
 for Metric)?", VBYesNo + VBQuestion, "Select Drill Format")

    If rtn = VBYes Then
        ' Match DrillEnglish.dff
        oNCDrillMF.DataType                 = eengExcellon
        oNCDrillMF.Unit                     = eengUnitInch
        oNCDrillMF.DataFormatLeadingDigits  = 2
        oNCDrillMF.DataFormatTrailingDigits = 4
        oNCDrillMF.StepMode                 = eengStepAbsolute
        oNCDrillMF.ZeroTruncation           = eengZeroTruncationTrailing
        oNCDrillMF.DataMode                 = TRUE
        oNCDrillMF.ArcStyle                 = eengArcStyleRadius
        oNCDrillMF.SequenceNumbering        = FALSE
        oNCDrillMF.CharacterSet             = eengCharacterSetASCII
        oNCDrillMF.Delimiter                = ""
        oNCDrillMF.Comments                 = TRUE
        oNCDrillMF.CommentStr               = "; "
        oNCDrillMF.RecordLength             = 0
    Else
        ' Match DrillMetric.dff
        oNCDrillMF.DataType                 = eengExcellon
        oNCDrillMF.Unit                     = eengUnitMM
        oNCDrillMF.DataFormatLeadingDigits  = 3
        oNCDrillMF.DataFormatTrailingDigits = 3
        oNCDrillMF.StepMode                 = eengStepAbsolute
        oNCDrillMF.ZeroTruncation           = eengZeroTruncationLeading
        oNCDrillMF.DataMode                 = TRUE
        oNCDrillMF.ArcStyle                 = eengArcStyleRadius
        oNCDrillMF.SequenceNumbering        = FALSE
        oNCDrillMF.CharacterSet             = eengCharacterSetASCII
        oNCDrillMF.Delimiter                = ""
        oNCDrillMF.Comments                 = TRUE
        oNCDrillMF.CommentStr               = "; "
        oNCDrillMF.RecordLength             = 0
    End If

    ' Optional, where to write the format file
    oNCDrillMF.FileName = jobConfigPath + "UserDrillMachineFormat.dff"
```

```
End Sub

Sub SetupParameters(oNCDrillEngine)
    'On Error Resume Next

    Dim oNCDrillParameters
    Set oNCDrillParameters = oNCDrillEngine.Parameters

    ' Same as the options in "Drill Options"
    ' tab of the "NC Drill Generation" dialog
    oNCDrillParameters.SweepAxis = eengSweepaxisHorizontal
    oNCDrillParameters.Bandwidth(eengUnitInch) =  0.1
    oNCDrillParameters.PreDrillHolesLargerThan(eengUnitInch) =  1
    oNCDrillParameters.OutputFileExtension = ".ncd"
 Call oNCDrillParameters.ClearFileHeader()
 Call oNCDrillParameters.ClearFileNotes()
    Call oNCDrillParameters.AddFileHeader( "header")
    Call oNCDrillParameters.AddFileNotes ("notes")
End Sub

Sub SetupChart(oNCDrillEngine)
    'On Error Resume Next

    Dim oNCDrillChart
    Set oNCDrillChart = oNCDrillEngine.Chart

    ' Same as the options in "Drill Chart Options"
    ' tab of the "NC Drill Generation" dialog
    oNCDrillChart.FontName = "VeriBest Gerber 0"
    oNCDrillChart.FontSize(eengUnitInch) =  0.05
    oNCDrillChart.LineSpacing(eengUnitInch) = 0.05
    oNCDrillChart.PenWidth(eengUnitInch) =  0.005
    oNCDrillChart.PrecisionLeadingDigits = 1
    oNCDrillChart.PrecisionTrailingDigits = 4
    oNCDrillChart.Unit = eengUnitsEnglish
    oNCDrillChart.PositiveTolerance(eengUnitInch) =  1
    oNCDrillChart.NegativeTolerance(eengUnitInch) =  1
    oNCDrillChart.AssignDrillCharacters = FALSE
    oNCDrillChart.Title = "title"
    oNCDrillChart.SpecialNotes = "notes"

    Dim oNCDrillChartColumns
    Set oNCDrillChartColumns = oNCDrillChart.Columns

    oNCDrillChartColumns.Add eengColumnSymbol
    oNCDrillChartColumns.Add eengColumnDiameter
    oNCDrillChartColumns.Add eengColumnTolerance
    oNCDrillChartColumns.Add eengColumnPlated
    oNCDrillChartColumns.Add eengColumnPunched
    oNCDrillChartColumns.Add eengColumnHolename
    oNCDrillChartColumns.Add eengColumnQuantity
End Sub

Public Function GetLicensedDoc(appObj)
```

```
    On Error Resume Next
    Dim key, licenseServer, licenseToken, docObj
    Set GetLicensedDoc = Nothing
    ' collect the active document
    Set docObj = appObj.ActiveDocument
    If Err Then
        Call appObj.Gui.StatusBarText("No active document: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' Ask Xpedition Layout's document for the key
    key = docObj.Validate(0)
    ' Get token from license server
    Set licenseServer =
 CreateObject("MGCPCBAutomationLicensing.Application")
    licenseToken = licenseServer.GetToken(key)
    Set licenseServer = Nothing
    ' Ask the document to validate the license token
    Err.Clear
    Call docObj.Validate(licenseToken)
    If Err Then
        Call appObj.Gui.StatusBarText("No active document license: " &
 Err.Description, epcbStatusFieldError)
        Exit Function
    End If
    ' everything is OK, return document
    Set GetLicensedDoc = docObj
End Function
```

Table 16. NCDrill Object Methods and Properties

Method or Property	Description
Go Method (NCDrill Object)	Generates NC Drill output files.
Chart Property (NCDrill Object)	Returns the NC Drill chart. Refer to the Drill Chart Options tab of the NC Drill Generation dialog box.
DesignFileName Property (NCDrill Object)	Sets the name of the design file, including path.
Errors Property (NCDrill Object)	Returns the collection of errors.
MachineFormat Property (NCDrill Object)	Returns the NC Drill machine format. Refer to the Drill Options tab of the NC Drill Generation dialog box.
OutputDirectory Property (NCDrill Object)	Sets the name of the output directory.
Parameters Property (NCDrill Object)	Returns the NC Drill parameters. Refer to the Drill Options tab of the NC Drill Generation dialog box.
SetupFileName Property (NCDrill Object)	Sets the name of the NC Drill setup file. This is the DrillPreferences.hkp file that the NCDrill engine uses to process output data and to load the NC Drill dialog box.
UseWorkDirectory Property (NCDrill Object)	Sets the option to use the work directory.

Go Method (NCDrill Object)

Object: NCDrill Object

Prerequisites: None

Generates NC Drill output files.

Usage

NCDrill.Go ()

Arguments

None

Examples

```
oNCDrillEnging.Go
```

Chart Property (NCDrill Object)

Object: NCDrill Object

Access: Read-Only

Prerequisites: None

Returns the NC Drill chart. Refer to the Drill Chart Options tab of the NC Drill Generation dialog box.

Usage

NCDrill.Chart

Arguments

None

Return Values

NCDrillChart. Returns a NCDrillChart Object

Examples

```
oNCDrillChart = oNCDrillEngine.Chart
```

DesignFileName Property (NCDrill Object)

Object: NCDrill Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

NCDrill.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the design file name.

Examples

```
oNCDrillEngine.DesignFileName = jobName
```

Errors Property (NCDrill Object)

Object: NCDrill Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

NCDrill.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oNCDrillEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

MachineFormat Property (NCDrill Object)

Object: NCDrill Object

Access: Read-Only

Prerequisites: None

Returns the NC Drill machine format. Refer to the Drill Options tab of the NC Drill Generation dialog box.

Usage

NCDrill.MachineFormat

Arguments

None

Return Values

NCDrillMachineFormat. Return a NCDrillMachineFormat Object

Examples

```
oNCDrillMF = oNCDrillEngine.MachineFormat
```

OutputDirectory Property (NCDrill Object)

Object: NCDrill Object

Access: Write-Only

Prerequisites: None

Sets the name of the output directory.

Usage

NCDrill.OutputDirectory = String

Arguments

None

Return Values

String. A string that contains the output directory.

Examples

```
oNCDrillEngine.OutputDirectory = outputDirectory
```

Parameters Property (NCDrill Object)

Object: NCDrill Object

Access: Read-Only

Prerequisites: None

Returns the NC Drill parameters. Refer to the Drill Options tab of the NC Drill Generation dialog box.

Usage

NCDrill.Parameters

Arguments

None

Return Values

NCDrillParamaters. Returns a handle to the NCDrillParameters Object.

Examples

```
oNCDrillParameters = oNCDrillEngine.Parameters
```

SetupFileName Property (NCDrill Object)

Object: NCDrill Object

Access: Write-Only

Prerequisites: None

Sets the name of the NC Drill setup file. This is the *DrillPreferences.hkp* file that the NCDrill engine uses to process output data and to load the NC Drill dialog box.

Usage

NCDrill.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the NC Drill setup file name.

Examples

```
oNCDrillEngine.SetupFileName = jobConfigPath + "DrillPreferences.hkp"
```

UseWorkDirectory Property (NCDrill Object)

Object: NCDrill Object

Access: Write-Only

Prerequisites: None

Sets the option to use the work directory.

Usage

```
NCDrill.UseWorkDirectory = True | False
```

Arguments

None

Return Values

Boolean. True to use work directory. False to use saved layout data.

Description

Use this flag if you are running a script inside Expedition and want to use the work directory. By default this is set to true. Set it to false to use saved layout data.

NCDrillChart Object

The NCDrillChart Object contains information the NC Drill Generation Engine uses to set numerically-controlled (NC) drill options.

Refer to the Drill Chart Options tab of the NC Drill Generation dialog box.

See also NCDrill Object.

Table 17. NCDrillChart Object Properties

Property	Description
ResetTolerance Method (NCDrillChart Object)	Resets the positive and negative tolerances to the default values.
AllSpansOnSingleChart Property (NCDrillChart Object)	Sets the option to automatically assign characters or symbols to drill sizes. Refer to the Drill Chart Options tab, Auto assign drill symbols option.
AssignDrillCharacters Property (NCDrillChart Object)	Sets the option to automatically assign characters or symbols to drill sizes. Refer to the Drill Chart Options tab, Auto assign drill symbols option.
Columns Property (NCDrillChart Object)	Returns the collection of NC Drill chart columns. Refer to the Drill Chart Options tab, Columns option.
DrillHoles Property (NCDrillChart Object)	Returns the collection of Holes.
DrillSymbolsOnSeparateLayers Property (NCDrillChart Object)	Sets the option to put drill symbols on separate layers.
FCFFontName Property (NCDrillChart Object)	Sets the FCF font name for the NC drill chart.
FontName Property (NCDrillChart Object)	Sets the font name for the NC drill chart.
FontSize Property (NCDrillChart Object)	Sets the font size for the NC drill chart.
HorizontalLineSeparator Property (NCDrillChart Object)	Adds a horizontal separator if set to True.
IncludeContours Property (NCDrillChart Object)	Sets the option to include contours in the main drill chart. Refer to the Drill Chart Options tab, Include contours option.
LineSpacing Property (NCDrillChart Object)	Sets the line spacing for the NC drill chart.
NegativeTolerance Property (NCDrillChart Object)	Sets the default negative tolerance.
PenWidth Property (NCDrillChart Object)	Sets the pen width for the NC drill chart.
PositiveTolerance Property (NCDrillChart Object)	Sets the default positive tolerance.
PrecisionLeadingDigits Property (NCDrillChart Object)	Sets the drill chart precision leading digits.
PrecisionTrailingDigits Property (NCDrillChart Object)	Sets the drill chart precision trailing digits.
SpecialNotes Property (NCDrillChart Object)	Sets the drill chart special notes.
TextFormatTitleCase Property (NCDrillChart Object)	Uses the title case if set to True. Otherwise, uses upper case.
Title Property (NCDrillChart Object)	Sets the drill chart title.

Table 17. NCDrillChart Object Properties (continued)

Property	Description
UnitEx Property (NCDrillChart Object)	Sets the drill chart unit.
VerticalLineSeparator Property (NCDrillChart Object)	Adds a vertical separator if set to True.

ResetTolerance Method (NCDrillChart Object)

Scope: Xpedition Enterprise

Object: NCDrillChart Object

Prerequisites: None

Resets the positive and negative tolerances to the default values.

Usage

```
NCDrillChart.ResetTolerance()
```

Arguments

None

Description

Method is equivalent to clearing the tolerance fields in the NCDrill dialog. The purpose of resetting these values is to suppress the "All Drills (Unless Specified) / +/- # (unit)" text in the resulting drill charts.

AllSpansOnSingleChart Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the option to automatically assign characters or symbols to drill sizes. Refer to the Drill Chart Options tab, Auto assign drill symbols option.

Usage

```
NCDrillChart.AllSpansOnSingleChart = True | False
```

Arguments

None

Return Values

Boolean. True to automatically assign characters to drill sizes. False to assign symbols.

AssignDrillCharacters Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the option to automatically assign characters or symbols to drill sizes. Refer to the Drill Chart Options tab, Auto assign drill symbols option.

Usage

NCDrillChart.AssignDrillCharacters = True | False

Arguments

None

Return Values

Boolean. True to automatically assign characters to drill sizes. False to assign symbols.

Examples

```
oNCDrillChart.AssignDrillCharacters = FALSE
```

Columns Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Read-Only

Prerequisites: None

Returns the collection of NC Drill chart columns. Refer to the Drill Chart Options tab, Columns option.

Usage

NCDrillChart.Columns

Arguments

None

Return Values

NCDrillChartColumns. Returns the NCDrillChartColumns Collection.

Examples

```
Set oNCDrillChartColumns = oNCDrillChart.Columns

oNCDrillChartColumns.Add eengColumnSymbol
oNCDrillChartColumns.Add eengColumnDiameter
oNCDrillChartColumns.Add eengColumnTolerance
```

DrillHoles Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Read-Only

Prerequisites: None

Returns the collection of Holes.

Usage

```
NCDrillChart.DrillHoles
```

Arguments

None

Return Values

DrillHoles. Returns the DrillHoles Collection.

DrillSymbolsOnSeparateLayers Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the option to put drill symbols on separate layers.

Usage

```
NCDrillChart.DrillSymbolsOnSeparateLayers = True | False
```

Arguments

None

Return Values

Boolean. True to put drill symbols on separate layers. False to put drill symbols on a single layer.

FCFFontName Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the FCF font name for the NC drill chart.

Usage

```
NCDrillChart.FCFFontName = String
```

Arguments

None

Return Values

String. A string that contains the FCF font name.

FontName Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the font name for the NC drill chart.

Usage

NCDrillChart.FontName = String

Arguments

None

Return Values

String. A string that contains the font name.

Examples

```
oNCDrillChart.FontName = "VeriBest Gerber 0"
```

FontSize Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the font size for the NC drill chart.

Usage

NCDrillChart.FontSize ([ByVal eUnit As EEnginesUnit = eengUnitInch]) = Double

Arguments

- eUnit

  EEnginesUnit Enum. (Optional) Sets the units used in processing the font size. If not specified, the unit is inches.

Return Values

Double. A double that specifies the font size to use.

Examples

```
oNCDrillChart.FontSize(eengUnitInch) =  0.05
```

HorizontalLineSeparator Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Adds a horizontal separator if set to True.

Usage

```
NCDrillChart.HorizontalLineSeparator = True | False
```

Arguments

None

Return Values

Boolean. True to add horizontal separator. False, for no separator.

IncludeContours Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the option to include contours in the main drill chart. Refer to the Drill Chart Options tab, Include contours option.

Usage

```
NCDrillChart.IncludeContours = True | False
```

Arguments

None

Return Values

Boolean. True to include contours in the drill chart. False to not include contours.

LineSpacing Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the line spacing for the NC drill chart.

Usage

NCDrillChart.LineSpacing ([ByVal eUnit As EEnginesUnit = eengUnitInch])= Double

Arguments

- eUnit

  EEnginesUnit Enum. (Optional) Sets the units used in processing the line spacing. If not specified, the unit is inches.

Return Values

Double. A double that specifies the line spacing of the chart.

Examples

```
oNCDrillChart.LineSpacing(eengUnitInch) = 0.05
```

NegativeTolerance Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the default negative tolerance.

Usage

NCDrillChart

Arguments

- eUnit

  EEnginesUnit Enum. (Optional) Sets the units used for the tolerance. If not specified, the unit is inches.

Return Values

Double. A double that specifies the default negative tolerance.

Examples

```
oNCDrillChart.NegativeTolerance(eengUnitInch) =  1
```

PenWidth Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the pen width for the NC drill chart.

Usage

NCDrillChart.PenWidth([ByVal eUnit As EEnginesUnit = eengUnitInch]) = Double

Arguments

- eUnit

  EEnginesUnit Enum. (Optional) Sets the units used for the pen width. If not specified, the unit is inches.

Return Values

Double. A double that specifies the pen width.

Examples

```
oNCDrillChart.PenWidth(eengUnitInch) =  0.005
```

PositiveTolerance Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the default positive tolerance.

Usage

NCDrillChart.PositiveTolerance ([ByVal eUnit As EEnginesUnit = eengUnitInch]) = Double

Arguments

- eUnit

  EEnginesUnit Enum. (Optional) Sets the units used for the tolerance. If not specified, the unit is inches.

Return Values

Double. A double that specifies the default positive tolerance.

Examples

```
oNCDrillChart.PositiveTolerance(eengUnitInch) =  1
```

PrecisionLeadingDigits Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the drill chart precision leading digits.

Usage

NCDrillChart.PrecisionLeadingDigits = Long

Arguments

None

Return Values

Long. A long that specifies the precision leading digits.

Examples

```
oNCDrillChart.PrecisionLeadingDigits = 1
```

PrecisionTrailingDigits Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the drill chart precision trailing digits.

Usage

NCDrillChart.PrecisionTrailingDigits = Long

Arguments

None

Return Values

Long. A long that specifies the precision trailing digits.

Examples

```
oNCDrillChart.PrecisionTrailingDigits = 4
```

SpecialNotes Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the drill chart special notes.

Usage

NCDrillChart.SpecialNotes = String

Arguments

None

Return Values

String. A string that contains the special notes.

Examples

```
oNCDrillChart.SpecialNotes = "notes"
```

TextFormatTitleCase Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Uses the title case if set to True. Otherwise, uses upper case.

Usage

```
NCDrillChart.TextFormatTitleCase = True | False
```

Arguments

None

Return Values

Boolean. True to use title case. False to use upper case.

Title Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the drill chart title.

Usage

NCDrillChart.Title = String

Arguments

None

Return Values

String. A string that contains the title of the drill chart.

Examples

```
oNCDrillChart.Title = "title"
```

UnitEx Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Sets the drill chart unit.

Usage

NCDrillChart.UnitEx = EEnginesNCDrillChartUnits

Arguments

None

Return Values

EEnginesNCDrillChartUnits. EEnginesNCDrillChartUnits Enum

Examples

```
Let UnitEx As EEnginesNCDrillChartUnits
```

VerticalLineSeparator Property (NCDrillChart Object)

Object: NCDrillChart Object

Access: Write-Only

Prerequisites: None

Adds a vertical separator if set to True.

Usage

```
NCDrillChart.VerticalLineSeparator = True | False
```

Arguments

None

Return Values

Boolean. True to add vertical separator. False, no separator.

NCDrillChartColumn Object

The NCDrillChartColumn Object sets or displays column information for the numerically-controlled (NC) drill chart.

Table 18. NCDrillChartColumn Object Methods and Properties

Method or Property	Description
Alignment Property (NCDrillChartColumn Object)	Sets or returns the column alignment.
ColumnType Property (NCDrillChartColumn Object)	Returns the drill chart column type.
DisplayName Property (NCDrillChartColumn Object)	Sets or returns the Display name for the NC drill chart column..
Name Property (NCDrillChartColumn Object)	Returns the drill chart column name. This is the column type for pre-defined columns or the name for user-defined columns.
PutUserColumnInput Method (NCDrillChartColumn Object)	Sets the user column input string for the given hole name on this column.
UserColumnInput Property (NCDrillChartColumn Object)	Returns the value of the user column for the specified hole.

Alignment Property (NCDrillChartColumn Object)

Object: NCDrillChartColumn Object

Access: Read/Write

Prerequisites: None

Sets or returns the column alignment.

Usage

```
NCDrillChartColumn.Alignment = EEnginesColumnAlignment
```

Arguments

None

Return Values

EEnginesColumnAlignment. The column alignment as EEnginesColumnAlignment Enum.

ColumnType Property (NCDrillChartColumn Object)

Object: NCDrillChartColumn Object

Access: Read-Only

Prerequisites: None

Returns the drill chart column type.

Usage

```
NCDrillChartColumn.ColumnType
```

Arguments

None

Return Values

EEnginesNCDrillColumn. The column type as EEnginesNCDrillColumn Enum.

DisplayName Property (NCDrillChartColumn Object)

Object: NCDrillChartColumn Object

Access: Read/Write

Prerequisites: None

Sets or returns the Display name for the NC drill chart column..

Usage

```
NCDrillChartColumn.DisplayName = String
```

Arguments

None

Return Values

String. A string that contains the display name.

Name Property (NCDrillChartColumn Object)

Object: NCDrillChartColumn Object

Access: Read-Only

Prerequisites: None

Returns the drill chart column name. This is the column type for pre-defined columns or the name for user-defined columns.

Usage

```
NCDrillChartColumn.Name
```

Arguments

None

Return Values

String. A string that contains the column name.

PutUserColumnInput Method (NCDrillChartColumn Object)

Object: NCDrillChartColumn Object

Prerequisites: None

Sets the user column input string for the given hole name on this column.

Usage

```
NCDrillChartColumn.PutUserColumnInput(
        ByVal holeName As String,
        ByVal strInput As String)
```

Arguments

- holeName

  A string that contains the hole name.
- strInput

  A string that contains the input string.

UserColumnInput Property (NCDrillChartColumn Object)

Object: NCDrillChartColumn Object

Access: Read-Only

Prerequisites: None

Returns the value of the user column for the specified hole.

Usage

```
NCDrillChartColumn.UserColumnInput (ByVal holeName As String)
```

Arguments

- holeName

  A string that contains the hole name.

Return Values

String. A string that contains the values for the specified hole.

NCDrillMachineFormat Object

The NCDrillMachineFormat Object sets NCDrill data for the numerically-controlled (NC) drill machine.

See also NCDrill Object.

Table 19. NCDrillMachineFormat Object Properties

Property	Description
ArcStyle Property (NCDrillMachineFormat Object)	Sets the NC Drill data arc style.
CharacterSet Property (NCDrillMachineFormat Object)	Sets the NC Drill data character set.
Comments Property (NCDrillMachineFormat Object)	Sets the option to include or exclude comments in the output file.
CommentStr Property (NCDrillMachineFormat Object)	Sets the string to use as a comment indicator.
DataFormatLeadingDigits Property (NCDrillMachineFormat Object)	Sets the leading digits of NC Drill data format.
DataFormatTrailingDigits Property (NCDrillMachineFormat Object)	Sets the trailing digits of NC Drill data format.
DataMode Property (NCDrillMachineFormat Object)	Sets the NC Drill data mode.
DataType Property (NCDrillMachineFormat Object)	Sets the NC Drill data type.
Delimiter Property (NCDrillMachineFormat Object)	Sets the NC Drill data delimiter.
FileName Property (NCDrillMachineFormat Object)	Sets the name of the NC Drill machine format file, including path.
RecordLength Property (NCDrillMachineFormat Object)	Sets the NC Drill data record length.
SequenceNumbering Property (NCDrillMachineFormat Object)	Sets the option to enable or disable sequence numbering.
StepMode Property (NCDrillMachineFormat Object)	Sets the NC Drill step mode.
Unit Property (NCDrillMachineFormat Object)	Sets the NC Drill data unit.
ZeroTruncation Property (NCDrillMachineFormat Object)	Sets the Zero Truncation value.

ArcStyle Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the NC Drill data arc style.

Usage

NCDrillMachineFormat.ArcStyle = EEnginesArcStyle

Arguments

None

Return Values

EEnginesArcStyle: EEnginesArcStyle Enum

Examples

```
oNCDrillMF.ArcStyle = eengArcStyleRadius
```

CharacterSet Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the NC Drill data character set.

Usage

NCDrillMachineFormat.CharacterSet = EEnginesCharacterSet

Arguments

None

Return Values

EEnginesCharacterSet: EEnginesCharacterSet Enum

Examples

```
oNCDrillMF.CharacterSet = eengCharacterSetASCII
```

Comments Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the option to include or exclude comments in the output file.

Usage

NCDrillMachineFormat.Comments = True | False

Arguments

None

Return Values

Boolean. True to include comments. False to exclude comments.

Examples

```
oNCDrillMF.Comments = TRUE
```

CommentStr Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the string to use as a comment indicator.

Usage

NCDrillMachineFormat.CommentStr = String

Arguments

None

Return Values

String. A string that contains the comment indicator in the machine format file.

Examples

```
oNCDrillMF.CommentStr = "; "
```

DataFormatLeadingDigits Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the leading digits of NC Drill data format.

Usage

NCDrillMachineFormat.DataFormatLeadingDigits = Long

Arguments

None

Return Values

Long. A long that specifies the leading digits of the NC Drill data format.

Examples

```
oNCDrillMF.DataFormatLeadingDigits = 2
```

DataFormatTrailingDigits Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the trailing digits of NC Drill data format.

Usage

NCDrillMachineFormat.DataFormatTrailingDigits = Long

Arguments

None

Return Values

Long. A long that specifies the trailing digits of the NC Drill data format.

Examples

```
oNCDrillMF.DataFormatTrailingDigits = 4
```

DataMode Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the NC Drill data mode.

Usage

NCDrillMachineFormat.DataMode = True | False

Arguments

None

Return Values

Boolean. True for modal. False for non-modal.

Examples

```
oNCDrillMF.DataMode = TRUE
```

DataType Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the NC Drill data type.

Usage

NCDrillMachineFormat.DataType = EEnginesNCDrillDataType

Arguments

None

Return Values

EEnginesNCDrillDataType: EEnginesNCDrillDataType Enum

Examples

```
oNCDrillMF.DataType = eengExcellon
```

Delimiter Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the NC Drill data delimiter.

Usage

NCDrillMachineFormat.Delimiter = String

Arguments

None

Return Values

String. A string that contains the NC Drill data delimiter.

Examples

```
oNCDrillMF.Delimiter = ""
```

FileName Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the name of the NC Drill machine format file, including path.

Usage

NCDrillMachineFormat.FileName = String

Arguments

None

Return Values

String. A string that specifies the NC Drill machine format file name, including path.

Examples

```
oNCDrillMF.FileName = jobConfigPath + "UserDrillMachineFormat.dff"
```

RecordLength Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the NC Drill data record length.

Usage

NCDrillMachineFormat.RecordLength = Long

Arguments

None

Return Values

Long. A long that specifies the NC Drill data record length.

Examples

```
oNCDrillMF.RecordLength = 0
```

SequenceNumbering Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the option to enable or disable sequence numbering.

Usage

NCDrillMachineFormat.SequenceNumbering = True | False

Arguments

None

Return Values

Boolean. True sets sequence numbering on. False turns sequence numbering off.

Examples

```
oNCDrillMF.SequenceNumbering = FALSE
```

StepMode Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the NC Drill step mode.

Usage

NCDrillMachineFormat.StepMode = EEnginesStepMode

Arguments

None

Return Values

EEnginesStepMode: EEnginesStepMode Enum

Examples

```
oNCDrillMF.StepMode = eengStepAbsolute
```

Unit Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the NC Drill data unit.

Usage

NCDrillMachineFormat.Unit = EEnginesUnit

Arguments

None

Return Values

EEnginesUnit: EEnginesUnit Enum. Sets the NC Drill data unit.

Examples

```
oNCDrillMF.Unit = eengUnitInch
```

ZeroTruncation Property (NCDrillMachineFormat Object)

Object: NCDrillMachineFormat Object

Access: Write-Only

Prerequisites: None

Sets the Zero Truncation value.

Usage

NCDrillMachineFormat.ZeroTruncation = EEnginesZeroTruncation

Arguments

None

Return Values

Value: EEnginesZeroTruncation Enum

Examples

```
oNCDrillMF.ZeroTruncation = eengZeroTruncationTrailing
```

NCDrillParameters Object

The NCDrillParameters Object sets the parameters for numerically-controlled (NC) drill output.

For more information, refer to the Drill Options tab of the NC Drill Generation dialog box by choosing **Output > NC Drill**.

See also Parameters Property (NCDrill Object).

Table 20. NCDrillParameters Object Methods and Properties

Method or Property	Description
AddFileHeader Method (NCDrillParameters Object)	Adds the drill output file header text.
AddFileNotes Method (NCDrillParameters Object)	Adds a drill output file note.
ClearFileHeader Method (NCDrillParameters Object)	Removes all drill output file header text.
ClearFileNotes Method (NCDrillParameters Object)	Removes all drill output file footer notes.
Bandwidth Property (NCDrillParameters Object)	Sets the drill sweep axis bandwidth.
OutputFileExtension Property (NCDrillParameters Object)	Sets the extension to use for the drill output files.
PreDrillHolesLargerThan Property (NCDrillParameters Object)	Sets the minimum drill hole size for a pre-drilled hole.
SweepAxis Property (NCDrillParameters Object)	Sets the drill sweep axis.

AddFileHeader Method (NCDrillParameters Object)

Object: NCDrillParameters Object

Prerequisites: None

Adds the drill output file header text.

Usage

NCDrillParameters.AddFileHeader (ByVal sHeader As String)

Arguments

- sHeader

  A string that contains the drill file header text to add.

Examples

```
oNCDrillParameters.AddFileHeader( "header")
```

AddFileNotes Method (NCDrillParameters Object)

Object: NCDrillParameters Object

Prerequisites: None

Adds a drill output file note.

Usage

NCDrillParameters.AddFileNotes (ByVal sNotes As String)

Arguments

- sNotes

  A string that contains the drill file note text to add.

Examples

```
oNCDrillParameters.AddFileNotes ("notes")
```

ClearFileHeader Method (NCDrillParameters Object)

Object: NCDrillParameters Object

Prerequisites: None

Removes all drill output file header text.

Usage

NCDrillParameters.ClearFileHeader ()

Arguments

None

Examples

```
oNCDrillParameters.ClearFileHeader()
```

ClearFileNotes Method (NCDrillParameters Object)

Object: NCDrillParameters Object

Prerequisites: None

Removes all drill output file footer notes.

Usage

NCDrillParameters.ClearFileNotes ()

Arguments

None

Examples

```
oNCDrillParameters.ClearFileNotes()
```

Bandwidth Property (NCDrillParameters Object)

Object: NCDrillParameters Object

Access: Write-Only

Prerequisites: None

Sets the drill sweep axis bandwidth.

Usage

NCDrillParameters.Bandwidth ([ByVal eUnit As EEnginesUnit = eengUnitInch]) = Double

Arguments

- eUnit

  EEnginesUnit Enum. (Optional) The units to use for the drill sweep axis bandwidth. If nothing is specified, inches is used.

Return Values

Double. A double that specifies the drill sweep axis bandwidth.

Examples

```
oNCDrillParameters.Bandwidth(eengUnitInch) =  0.1
```

OutputFileExtension Property (NCDrillParameters Object)

Object: NCDrillParameters Object

Access: Write-Only

Prerequisites: None

Sets the extension to use for the drill output files.

Usage

NCDrillParameters.OutputFileExtension = String

Arguments

None

Return Values

String. A string that specifies the extension to use for the drill output files.

Examples

```
oNCDrillParameters.OutputFileExtension = ".ncd"
```

PreDrillHolesLargerThan Property (NCDrillParameters Object)

Object: NCDrillParameters Object

Access: Write-Only

Prerequisites: None

Sets the minimum drill hole size for a pre-drilled hole.

Usage

NCDrillParameters.PreDrillHolesLargerThan ([ByVal eUnit As EEnginesUnit = eengUnitInch]) = Double

Arguments

- eUnit

  EEnginesUnit Enum (Optional) The units to use for the minimum drill hole size. If nothing is specified then inches is used.

Return Values

Double. A double that contains the minimum drill hole size for a pre-drilled hole.

Examples

```
oNCDrillParameters.PreDrillHolesLargerThan(eengUnitInch) =  1
```

SweepAxis Property (NCDrillParameters Object)

Object: NCDrillParameters Object

Access: Write-Only

Prerequisites: None

Sets the drill sweep axis.

Usage

NCDrillParameters.SweepAxis = EEnginesNCDrillSweepAxis

Arguments

None

Return Values

EEnginesNCDrillSweepAxis: EEnginesNCDrillSweepAxis Enum

Examples

```
oNCDrillParameters.SweepAxis = eengSweepaxisHorizontal
```

NeutralFileExport Object

The NeutralFileExport Object contains parameters for exporting an XE Neutral File.

For more information, refer to the Create XE Neutral File dialog box by choosing **Output > Neutral File**.

Figure 12. NeutralFileExport Data Model

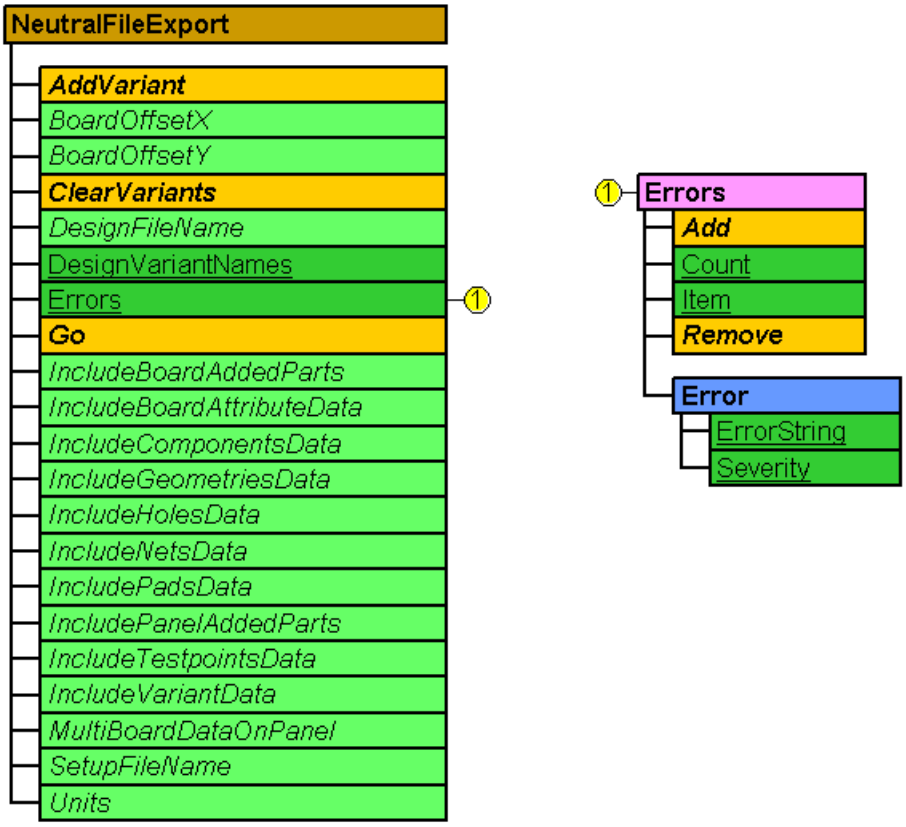


Table 21. NeutralFileExport Object Methods and Properties

Method or Property	Descrption
AddVariant Method (NeutralFileExport Object)	Adds adesign variant to output variant list.
ClearVariants Method (NeutralFileExport Object)	Clears the output variant list.
Go Method (NeutralFileExport Object)	Creates the XE Neutral File.
BoardOffsetX Property (NeutralFileExport Object)	Sets the X board offset.
BoardOffsetY Property (NeutralFileExport Object)	Sets the Y board offset.
DesignFileName Property (NeutralFileExport Object)	Sets the name of the design file, including path.

Table 21. NeutralFileExport Object Methods and Properties (continued)

Method or Property	Descrption
DesignVariantNames Property (NeutralFileExport Object)	Returns the variant names for the current design.
Errors Property (NeutralFileExport Object)	Returns the collection of errors.
IncludeBoardAddedParts Property (NeutralFileExport Object)	Sets the option to include information about fiducial, mounting holes, and mechanical cells.
IncludeBoardAttributeData Property (NeutralFileExport Object)	Sets the option to include board attributes.
IncludeComponentsData Property (NeutralFileExport Object)	Sets the option to include all component data.
IncludeGeometriesData Property (NeutralFileExport Object)	Sets the option to include geometry data.
IncludeHolesData Property (NeutralFileExport Object)	Sets the option to include holes data. This includes drill hole plating status, locations, and drill dimensions.
IncludeNetsData Property (NeutralFileExport Object)	Sets the option to include net data.
IncludePanelAddedParts Property (NeutralFileExport Object)	Sets the option to include information about fiducial, mounting holes, and mechanical cells. Valid only for Panel files.
IncludePadsData Property (NeutralFileExport Object)	Sets the option to include pads data. This includes pin and via padstacks, pin types, drill dimensions, pad shapes, and layers.
IncludeTestpointsData Property (NeutralFileExport Object)	Sets the option to include testpoint data.
IncludeVariantData Property (NeutralFileExport Object)	Sets the option to include variant data.
MultiBoardDataOnPanel Property (NeutralFileExport Object)	Sets the option to transform boards. Only for panel.
SetupFileName Property (NeutralFileExport Object)	Sets the name of the XE Neutral File setup file, including path. This is the neutral.hkp file that the Neutral File Export engine uses to process output data and to load the Create XE Neutral File dialog box.
Units Property (NeutralFileExport Object)	Sets the units when outputting a neutral file.

AddVariant Method (NeutralFileExport Object)

Object: NeutralFileExport Object

Prerequisites: None

Adds adesign variant to output variant list.

Usage

```
NeutralFileExport.AddVariant [ByVal variantName As String = “All”])
```

Arguments

- variantName

  (Optional) A string that contains the variant name. If not specified, the method adds all variant names to the output list.

ClearVariants Method (NeutralFileExport Object)

Object: NeutralFileExport Object

Prerequisites: None

Clears the output variant list.

Usage

```
NeutralFileExport.ClearVariants()
```

Arguments

None

Go Method (NeutralFileExport Object)

Object: NeutralFileExport Object

Prerequisites: None

Creates the XE Neutral File.

Usage

NeutralFileExport.Go()

Arguments

None

Examples

```
 oNeutralFileExportEngine.DesignFileName = doc.FullName
' Set the output options
 oNeutralFileExportEngine.BoardOffsetX = 5
 oNeutralFileExportEngine.BoardOffsetY = 5
 oNeutralFileExportEngine.IncludeBoardAttributeData = True
 oNeutralFileExportEngine.IncludeComponentsData = False
 oNeutralFileExportEngine.IncludeGeometriesData = True
 oNeutralFileExportEngine.IncludeHolesData = True
 oNeutralFileExportEngine.IncludeBoardAddedParts = True
 oNeutralFileExportEngine.IncludeNetsData = True
 oNeutralFileExportEngine.IncludePadsData = True
 oNeutralFileExportEngine.IncludeTestpointsData = True
 oNeutralFileExportEngine.IncludeVariantData = True
' Run the engine
 oNeutralFileExportEngine.Go
```

BoardOffsetX Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the X board offset.

Usage

NeutralFileExport.BoardOffsetX = Long

Arguments

None

Return Values

Long. A long that contains the X board offset value.

Examples

```
oNeutralFileExportEngine.BoardOffsetX =0
oNeutralFileExportEngine.BoardOffsetY = 0
```

BoardOffsetY Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the Y board offset.

Usage

NeutralFileExport.BoardOffsetY = Long

Arguments

None

Return Values

Long. A long that contains the Y board offset value.

Examples

```
oNeutralFileExportEngine.BoardOffsetX =0
oNeutralFileExportEngine.BoardOffsetY = 0
```

DesignFileName Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

NeutralFileExport.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the name of the design file.

Examples

```
oNeutralFileExportEngine.DesignFileName = jobName
```

DesignVariantNames Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Read-Only

Prerequisites: None

Returns the variant names for the current design.

Usage

```
NeutralFileExport.DesignVariantNames
```

Arguments

None

Return Values

Array. A array of variant names.

Errors Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

NeutralFileExport.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oNeutralFileExportEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

IncludeBoardAddedParts Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include information about fiducial, mounting holes, and mechanical cells.

Usage

NeutralFileExport.IncludeBoardAddedParts = True | False

Arguments

None

Return Values

Boolean. If True, include the part information. If False, do not include the part information.

Examples

```
oNeutralFileExportEngine.IncludeBoardAddedParts = True
```

IncludeBoardAttributeData Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include board attributes.

Usage

NeutralFileExport.IncludeBoardAttributeData = True | False

Arguments

None

Return Values

Boolean. If True, include board attributes. If False, do not include board attributes.

Examples

```
oNeutralFileExportEngine.IncludeBoardAttributeData = True
```

IncludeComponentsData Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include all component data.

Usage

NeutralFileExport.IncludeComponentsData = True | False

Arguments

None

Return Values

Boolean. If True, include all component data. If False, do not include component data.

Examples

```
oNeutralFileExportEngine.IncludeComponentsData = True
```

IncludeGeometriesData Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include geometry data.

Usage

NeutralFileExport.IncludeGeometriesData = True | False

Arguments

None

Return Values

Boolean. If True, include geometry data. If False, do not include geometry data.

Examples

```
oNeutralFileExportEngine.IncludeGeometriesData = True
```

IncludeHolesData Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include holes data. This includes drill hole plating status, locations, and drill dimensions.

Usage

NeutralFileExport.IncludeHolesData = True | False

Arguments

None

Return Values

Boolean. If True, include the holes data. If False, do not include the holes data.

Examples

```
oNeutralFileExportEngine.IncludeHolesData = True
```

IncludeNetsData Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include net data.

Usage

NeutralFileExport.IncludeNetsData = True | False

Arguments

None

Return Values

Boolean. If True, include net data. If False, do not include net data.

Examples

```
oNeutralFileExportEngine.IncludeNetsData = True
```

IncludePanelAddedParts Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include information about fiducial, mounting holes, and mechanical cells. Valid only for Panel files.

Usage

NeutralFileExport.IncludePanelAddedParts = True | False

Arguments

None

Return Values

Boolean. If True, include the part information. If False, do not include the part information.

Examples

```
oNeutralFileExportEngine.IncludePanelAddedParts = True
```

IncludePadsData Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include pads data. This includes pin and via padstacks, pin types, drill dimensions, pad shapes, and layers.

Usage

NeutralFileExport.IncludePadsData = True | False

Arguments

None

Return Values

Boolean. If True, include pads data. If False, do not include pads data.

Examples

```
oNeutralFileExportEngine.IncludePadsData = True
```

IncludeTestpointsData Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include testpoint data.

Usage

NeutralFileExport.IncludeTestpointsData = True | False

Arguments

None

Return Values

Boolean. If True, include testpoint data. If False, do not include testpoint data.

Examples

```
oNeutralFileExportEngine.IncludeTestpointsData = True
```

IncludeVariantData Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to include variant data.

Usage

NeutralFileExport.IncludeVariantData = True | False

Arguments

None

Return Values

Boolean. If True, include variant data. If False, do not include variant data.

Examples

```
oNeutralFileExportEngine.IncludeVariantData = True
```

MultiBoardDataOnPanel Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the option to transform boards. Only for panel.

Usage

NeutralFileExport.MultiBoardDataOnPanel = True | False

Arguments

None

Return Values

Boolean. If True, transform boards. If False, do not transform boards.

Examples

```
oNeutralFileExportEngine.MultiBoardDataOnPanel = True
```

SetupFileName Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the name of the XE Neutral File setup file, including path. This is the *neutral.hkp* file that the Neutral File Export engine uses to process output data and to load the Create XE Neutral File dialog box.

Usage

NeutralFileExport.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the name of the setup file.

Examples

```
oNeutralFileExportEngine.SetupFileName = _
   “mydesign/pcb/config/neutral.hkp”
```

Units Property (NeutralFileExport Object)

Object: NeutralFileExport Object

Access: Write-Only

Prerequisites: None

Sets the units when outputting a neutral file.

Usage

```
NeutralFileExport.Units = EEnginesUnit
```

Arguments

None

Return Values

EEnginesUnit. The units as “EEnginesUnit.eengUnitInch” on page 579.

ODBPPOutputEngine Object

The ODBPPOutputEngine Object contains parameters for ODB++ output.

For more information, refer to the ODB++ Output dialog box by choosing **Output > ODB++**.

Figure 13. ODBPPOutputEngine Data Model

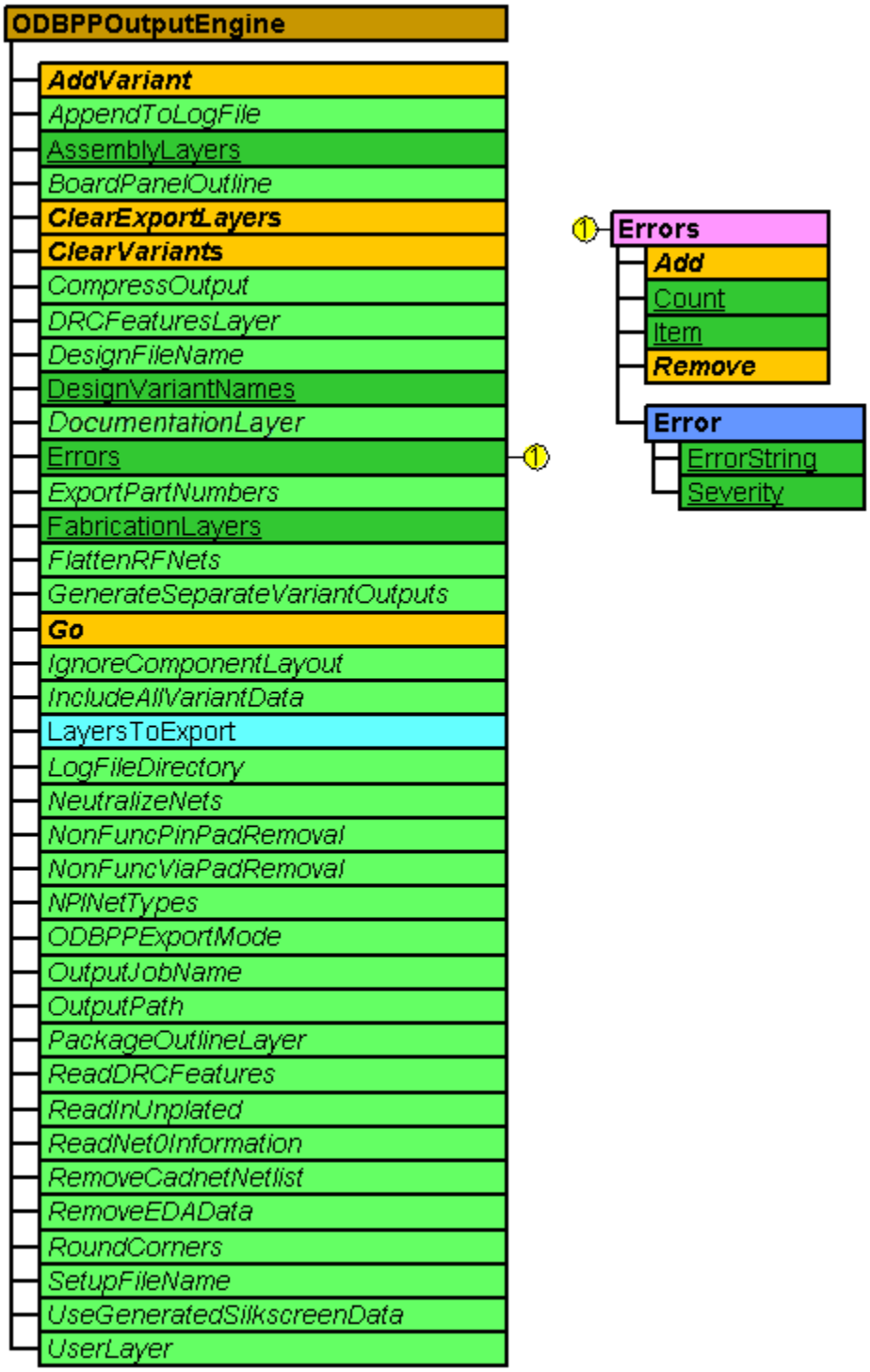


Table 22. ODBPPOutputEngine Object Methods and Properties

Method or Property	Description
AddVariant Method (ODBPPOutputEngine Object)	Adds a design variant to the output variant list.
ClearAdvancedPackagingLayers Method (ODBPPOutputEngine Object)	Clears the list of Advanced Packaging layers from the export list.
ClearExportLayers Method (ODBPPOutputEngine Object)	Clears all export layers.
ClearVariants Method (ODBPPOutputEngine Object)	Clears the output variant list.
Go Method (ODBPPOutputEngine Object)	Creates data in ODB++ format.
AdvancedPackagingData Property (ODBPPOutputEngine Object)	Sets the option to output Advanced Packaging data.
AdvancedPackagingLayers Property (ODBPPOutputEngine Object)	Returns the Advanced Packaging layer names.
AdvancedPackagingLayersToExport Property (ODBPPOutputEngine Object)	Sets or returns the list of Advanced Packaging layers to export. This appends to the current list of layers.
AppendToLogFile Property (ODBPPOutputEngine Object)	Sets the option to append to the most recent log file generated.
AssemblyLayers Property (ODBPPOutputEngine Object)	Returns the assembly layer names.
BoardPanelOutline Property (ODBPPOutputEngine Object)	Sets the option to output board or panel outline on all layers containing data.
CompressOutput Property (ODBPPOutputEngine Object)	Sets the option to create a compressed format. Compression creates a single tar'd and zipped file.
DesignFileName Property (ODBPPOutputEngine Object)	Sets the name of the design file, including path.
DesignVariantNames Property (ODBPPOutputEngine Object)	Returns the variant names for the current design.
DocumentationLayer Property (ODBPPOutputEngine Object)	Exports layers to an ODB++ documentation layer. If set to True, a documentation layer must be specified. Refer to the Define user layer mapping option in the ODB++ Output dialog box.
DRCFeaturesLayer Property (ODBPPOutputEngine Object)	Sets the DRC features layers name and enables the usage of the layer. Refer to the ODB++ DRC feature dialog box by choosing the Read DRC feature option on the ODB++ Output dialog box.
Errors Property (ODBPPOutputEngine Object)	Returns the collection of errors.
ExportEmbeddedPassiveLayers Property (ODBPPOutputEngine Object)	Sets the option to export Embedded Passive layers.
ExportPartNumbers Property (ODBPPOutputEngine Object)	Sets the option to export part numbers.
FabricationLayers Property (ODBPPOutputEngine Object)	Returns the fabrication layers names.
GenerateSeparateVariantOutputs Property (ODBPPOutputEngine Object)	Generates separate files for all variants if set to True. If no variants is specified, the property outputs the Master design.
IgnoreComponentLayout Property (ODBPPOutputEngine Object)	Sets the option to ignore the Component_Layout_Type property.

Table 22. ODBPPOutputEngine Object Methods and Properties (continued)

Method or Property	Description
IncludeAllVariantData Property (ODBPPOutputEngine Object)	Includes all variant data in single output file if set to True.
LayersToExport Property (ODBPPOutputEngine Object)	Sets or returns the list layers to export. This appends to the current list of layers.
LogFileDirectory Property (ODBPPOutputEngine Object)	Sets the directory to use when outputting the log file.
NeutralizeNets Property (ODBPPOutputEngine Object)	Sets the option to assign random names to nets.
NonFuncPinPadRemoval Property (ODBPPOutputEngine Object)	Sets the options for non-functional pad removal for pins.
NonFuncViaPadRemoval Property (ODBPPOutputEngine Object)	Sets the options for non-functional pad removal for vias. Only “unused” and “none” are valid options.
NPINetTypes Property (ODBPPOutputEngine Object)	Sets the option to generate NPI net types.
ODBPPExportMode Property (ODBPPOutputEngine Object)	Sets the type of data to export.
ODBPPVersion Property (ODBPPOutputEngine Object)	Sets the ODBpp Output Version.
OutputJobName Property (ODBPPOutputEngine Object)	Sets the ODB++ output job name. This does not include the path.
OutputPath Property (ODBPPOutputEngine Object)	Sets the ODB++ output path. This does not include the job name.
PackageOutlineLayer Property (ODBPPOutputEngine Object)	Sets the package outline layer. A User Layer requires a name. Panel Layer and Panel Text are only valid for a panel design.
ReadDRCFeatures Property (ODBPPOutputEngine Object)	Sets the option to export DRC data.
RemoveCadnetNetlist Property (ODBPPOutputEngine Object)	Sets the option to remove the cadnet netlist.
RemoveEDAData Property (ODBPPOutputEngine Object)	Sets the option to not export nets, components, or packages information.
RoundCorners Property (ODBPPOutputEngine Object)	Sets the option to create round corner surfaces.
SetupFileName Property (ODBPPOutputEngine Object)	Sets the name of the ODB++ output setup file, including path. This is the .ocf file that the ODBPPOutput engine uses to process output data and to load the ODB++ Output dialog box when the file is selected.
Units Property (ODBPPOutputEngine Object)	Sets the output units for outputting ODB++ (Metric or English).
UseGeneratedSilkscreenData Property (ODBPPOutputEngine Object)	Sets the option to export Silkscreen Generated data. Otherwise, regular silkscreen data is exported.
UserLayerEx Property (ODBPPOutputEngine Object)	Exports the user layers to an ODB++ job. Refer to the Define user layer mapping option in the ODB++ Output dialog box.

AddVariant Method (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Prerequisites: None

Adds a design variant to the output variant list.

Usage

ODBPPOutputEngine.AddVariant [ByVal variantName As String = “All”])

Arguments

- variantName

  (Optional) A string that contains the variant name. If not specified, the method adds all variant names to the output list.

ClearAdvancedPackagingLayers Method (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Prerequisites: None

Clears the list of Advanced Packaging layers from the export list.

Usage

ODBPPOutputEngine.ClearAdvancedPackagingLayers()

Arguments

None

Examples

```
ODBPPEngine.ClearAdvancedPackagingLayers
```

ClearExportLayers Method (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Prerequisites: None

Clears all export layers.

Usage

```
ODBPPOutputEngine.ClearExportLayers()
```

Arguments

None

ClearVariants Method (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Prerequisites: None

Clears the output variant list.

Usage

ODBPPOutputEngine.ClearVariants ()

Arguments

None

Go Method (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Prerequisites: None

Creates data in ODB++ format.

Usage

ODBPPOutputEngine.Go()

Arguments

None

Examples

```
oODBPPOutputEngine.Go()
```

AdvancedPackagingData Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to output Advanced Packaging data.

Usage

ODBPPOutputEngine.AdvancedPackagingData = True | False

Arguments

None

Return Values

Boolean. If True, output Advanced Packaging data. If False, do not output Advanced Packaging data.

Examples

```
oODBPPOutputEngine.AdvancedPackagingData = False
```

AdvancedPackagingLayers Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Read-Only

Prerequisites: None

Returns the Advanced Packaging layer names.

Usage

ODBPPOutputEngine.AdvancedPackagingLayers

Arguments

None

Return Values

Array. An array of Advanced Package layer names as strings.

Examples

```
Dim layerarray
layerarray = ODBPPEngine.AdvancedPackagingLayers
```

AdvancedPackagingLayersToExport Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Read/Write

Prerequisites: None

Sets or returns the list of Advanced Packaging layers to export. This appends to the current list of layers.

Usage

ODBPPOutputEngine.AdvancedPackagingLayersToExport = Array

Arguments

None

Return Values

Array. An array of layer names as strings.

Examples

```
Dim layerarray
layerarray = ODBPPEngine.AdvancedPackagingLayersToExport
ODBPPEngine.AdvancedPackagingLayersToExport = layerarray
```

AppendToLogFile Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to append to the most recent log file generated.

Usage

ODBPPOutputEngine.AppendToLogFile = True | False

Arguments

None

Return Values

Boolean. If True, append to the log file. If False, copy over the existing log file.

Examples

```
oODBPPOutputEngine.AppendToLogFile = False
```

AssemblyLayers Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Read-Only

Prerequisites: None

Returns the assembly layer names.

Usage

```
ODBPPOutputEngine.AssemblyLayers
```

Arguments

None

Return Values

Array. A array of layer names.

BoardPanelOutline Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to output board or panel outline on all layers containing data.

Usage

```
ODBPPOutputEngine.BoardPanelOutline = True | False
```

Arguments

None

Return Values

Boolean. If True, output board or panel outline on all layers containing data. If False, do not output the outline.

CompressOutput Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to create a compressed format. Compression creates a single tar'd and zipped file.

Usage

ODBPPOutputEngine.CompressOutput = True | False

Arguments

None

Return Values

Boolean. If True, compress output. If False, do not compress output.

Examples

```
oODBPPOutputEngine.CompressOutput = False
```

DesignFileName Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

ODBPPOutputEngine.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the name of the design file.

Examples

```
oODBPPOutputEngine.DesignFileName = jobName
```

DesignVariantNames Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Read-Only

Prerequisites: None

Returns the variant names for the current design.

Usage

```
ODBPPOutputEngine.DesignVariantNames
```

Arguments

None

Return Values

Array. A array of variant names.

DocumentationLayer Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Exports layers to an ODB++ documentation layer. If set to True, a documentation layer must be specified. Refer to the Define user layer mapping option in the ODB++ Output dialog box.

Usage

ODBPPOutputEngine.DocumentationLayer ([ByVal bIncludeDocumentationLayer As Boolean = False], [ByVal ODBDocumentationLayerName As String]) = Value

Arguments

- bIncludeDocumentationLayer

  (Optional) If True, export layers to an ODB++ documentation layer.
- ODBDocumentationLayerName

  (Optional) A string that contains the name of the documentation layer.

Return Values

Value. Can be either of the following:

- Layer names as strings, limited to user defined layers in the Setup Parameters dialog box.
- Mapped layers (using 1-256) for Layer1-Layer256 and “EEnginesMappedUserLayer” on page 559 for fabrication layers.

Examples

```
'Define User Layer Mapping
ReDim layName(0)
layName(0) = "SILKSCREEN_TOP"
odbEng.UserLayerEx(True, "SILKSCREEN_TOP") = layName
layName(0) = "SILKSCREEN_BOTTOM"
odbEng.UserLayerEx(True, "SILKSCREEN_BOTTOM") = layName

layName(0) = "SILKSCREEN_BOTTOM"
odbEng.DocumentationLayer(True, "usr_ssb") = layName

layName(0) = "SILKSCREEN_TOP"
odbEng.DocumentationLayer(True, "usr_sst") = layName
```

DRCFeaturesLayer Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the DRC features layers name and enables the usage of the layer. Refer to the ODB++ DRC feature dialog box by choosing the Read DRC feature option on the ODB++ Output dialog box.

Usage

```
ODBPPOutputEngine.DRCFeaturesLayer (ByVal bIncludeFeatureLayer As Boolean, ByVal
        eDRCFeature As EEnginesDRCFeaturesLayer) = String
```

Arguments

- bIncludeFeatureLayer

  If True, enables the usage of the feature layer.
- eDRCFeature

  The features layer type ().EEnginesDRCFeaturesLayer

Return Values

String. A string that contains the DRC features layer name.

Errors Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

ODBPPOutputEngine.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oODBPPOutputEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

ExportEmbeddedPassiveLayers Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to export Embedded Passive layers.

Usage

ODBPPOutputEngine.ExportEmbeddedPassiveLayers = True | False

Arguments

None

Return Values

Boolean. If True, export Embedded Passive layers. If False, do not export Embedded Passive layers.

Examples

```
oODBPPOutputEngine.ExportEmbeddedPassiveLayers = False
```

ExportPartNumbers Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to export part numbers.

Usage

ODBPPOutputEngine.ExportPartNumbers = True | False

Arguments

None

Return Values

Boolean. If True, export part numbers. If False, do not export part numbers.

Examples

```
oODBPPOutputEngine.ExportPartNumbers = False
```

FabricationLayers Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Read-Only

Prerequisites: None

Returns the fabrication layers names.

Usage

```
ODBPPOutputEngine.FabricationLayers
```

Arguments

None

Return Values

Array. A array of layer names.

GenerateSeparateVariantOutputs Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Generates separate files for all variants if set to True. If no variants is specified, the property outputs the Master design.

Usage

```
ODBPPOutputEngine.GenerateSeparateVariantOutputs = True | False
```

Arguments

None

Return Values

Boolean. If True, generate separate files for all variants. If False, generate output for the Master design.

IgnoreComponentLayout Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to ignore the Component_Layout_Type property.

Usage

ODBPPOutputEngine.IgnoreComponentLayout = True | False

Arguments

None

Return Values

Boolean. If True, ignore component layout. If False, do not ignore component layout.

Examples

```
oODBPPOutputEngine.IgnoreComponentLayout = False
```

IncludeAllVariantData Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Includes all variant data in single output file if set to True.

Usage

```
ODBPPOutputEngine.IncludeAllVariantData = True | False
```

Arguments

None

Return Values

Boolean. If True, output all variant data in a single file.

LayersToExport Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Read/Write

Prerequisites: None

Sets or returns the list layers to export. This appends to the current list of layers.

Usage

```
ODBPPOutputEngine.LayersToExport =  Array
```

Arguments

None

Return Values

Array. An array of layer names.

LogFileDirectory Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the directory to use when outputting the log file.

Usage

ODBPPOutputEngine.LogFileDirectory = String

Arguments

None

Return Values

String. A string that contains the log file directory.

Examples

```
oODBPPOutputEngine.LogFileDirectory = logDir
```

NeutralizeNets Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to assign random names to nets.

Usage

ODBPPOutputEngine.NeutralizeNets = True | False

Arguments

None

Return Values

Boolean. If True, the neutralize nets option is on and random names are assigned to nets. If False, the option is off.

Examples

```
oODBPPOutputEngine.NeutralizeNets = True
```

NonFuncPinPadRemoval Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the options for non-functional pad removal for pins.

Usage

ODBPPOutputEngine.NonFuncPinPadRemoval = EEnginesNonFuncPad

Arguments

None

Return Values

EEnginesNonFuncPad. The non-functional pad removal types (“EEnginesNonFuncPad” on page 567)

Examples

```
oODBPPOutputEngine.NonFuncPinPadRemoval = eengNonFuncRemoveNone
```

NonFuncViaPadRemoval Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the options for non-functional pad removal for vias. Only “unused” and “none” are valid options.

Usage

ODBPPOutputEngine.NonFuncViaPadRemoval = EEnginesNonFuncPad

Arguments

None

Return Values

EEnginesNonFuncPad. The non-functional pad removal types (“EEnginesNonFuncPad” on page 567)

Examples

```
oODBPPOutputEngine.NonFuncViaPadRemoval = eengNonFuncRemoveNone
```

NPINetTypes Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to generate NPI net types.

Usage

ODBPPOutputEngine.NPINetTypes = True | False

Arguments

None

Return Values

Boolean. If True, generate NPI net types. If False, do not generate NPI net types.

Examples

```
oODBPPOutputEngine.NPINetTypes = True
```

ODBPPExportMode Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the type of data to export.

Usage

ODBPPOutputEngine.ODBPPExportMode = EEnginesODBPPExportMode

Arguments

None

Return Values

EEnginesODBPPExportMode. The type of data to export ().EEnginesODBPPExportMode

Examples

```
oODBPPOutputEngine.ODBPPExportMode = eengFullExport
```

ODBPPVersion Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the ODBpp Output Version.

Usage

```
ODBPPOutputEngine.ODBPPVersion As Double
```

Arguments

None

Return Values

Double. The ODBpp version.

OutputJobName Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the ODB++ output job name. This does not include the path.

Usage

ODBPPOutputEngine.OutputJobName = String

Arguments

None

Return Values

String. A string that defines the output job name.

Examples

```
oODBPPOutputEngine.OutputJobName = outputFile
```

OutputPath Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the ODB++ output path. This does not include the job name.

Usage

ODBPPOutputEngine.OutputPath = String

Arguments

None

Return Values

String. A string that contains the output path.

Examples

```
oODBPPOutputEngine.OutputPath = outputPath
```

PackageOutlineLayer Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the package outline layer. A User Layer requires a name. Panel Layer and Panel Text are only valid for a panel design.

Usage

```
ODBPPOutputEngine.PackageOutlineLayer ([ByVal ePackageOutlineLayer As
        EEnginesPackageOutlineLayer = eengPackageOutlinePlacementOutline]) = String
```

Arguments

- ePackageOutlineLayer

  (Optional) The package outline layer type ().EEnginesPackageOutlineLayer If not specified the Placement Outline is used.

Return Values

String. A string that contains the package outline layer name.

Examples

```
oODBPPOutputEngine.PackageOutlineLayer = eengPackageOutlinePanelText
```

ReadDRCFeatures Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to export DRC data.

Usage

ODBPPOutputEngine.ReadDRCFeatures = True | False

Arguments

None

Return Values

Boolean. If True, export DRC data. If False, do not export DRC data.

Examples

```
oODBPPOutputEngine.ReadDRCFeatures = True
```

RemoveCadnetNetlist Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to remove the cadnet netlist.

Usage

```
ODBPPOutputEngine.RemoveCadnetNetlist = True | False
```

Arguments

None

Return Values

Boolean. If True, remove cadnet netlist. If False, do not remove cadnet netlist.

RemoveEDAData Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to not export nets, components, or packages information.

Usage

ODBPPOutputEngine.RemoveEDAData = True | False

Arguments

None

Return Values

Boolean. If True, do not export nets, components and packages information. If False, export the EDA data.

Examples

```
oODBPPOutputEngine.RemoveEDAData = False
```

RoundCorners Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to create round corner surfaces.

Usage

ODBPPOutputEngine.RoundCorners = True | False

Arguments

None

Return Values

Boolean. If True, create round corner surfaces. If False, do not create corner surfaces.

Examples

```
oODBPPOutputEngine.RoundCorners = True
```

SetupFileName Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the name of the ODB++ output setup file, including path. This is the *.ocf* file that the ODBPPOutput engine uses to process output data and to load the ODB++ Output dialog box when the file is selected.

Usage

ODBPPOutputEngine.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the name of the setup file.

Examples

```
oODBPPOutputEngine.SetupFileName = “mydesign/pcb/config/ODBSetup.ocf”
```

Units Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the output units for outputting ODB++ (Metric or English).

Usage

```
ODBPPOutputEngine.Units = EEnginesOutputUnit
```

Arguments

None

Return Values

EEnginesOutputUnit. The units as EEnginesOutputUnits Enum.

UseGeneratedSilkscreenData Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Sets the option to export Silkscreen Generated data. Otherwise, regular silkscreen data is exported.

Usage

ODBPPOutputEngine.UseGeneratedSilkscreenData = True | False

Arguments

None

Return Values

Boolean. If True, export silkscreen data from Generated Silkscreen. If False, export regular silkscreen data.

Examples

```
oODBPPOutputEngine.UseGeneratedSilkscreenData = True
```

UserLayerEx Property (ODBPPOutputEngine Object)

Object: ODBPPOutputEngine Object

Access: Write-Only

Prerequisites: None

Exports the user layers to an ODB++ job. Refer to the Define user layer mapping option in the ODB++ Output dialog box.

Usage

ODBPPOutputEngine.UserLayerEx = ([ByVal bIncludeUserLayer As Boolean = False], [ByVal UserLayerName As String]) = Value

Arguments

- bIncludeUserLayer

  (Optional) If True, export the user layer.
- UserLayerName

  (Optional) A string that contains the name of the user layer.

Return Values

Value. Can be either of the following:

- Layer names as strings, limited to user defined layers in the Setup Parameters dialog box.
- Mapped layers (using 1-256) for Layer1-Layer256 and “EEnginesMappedUserLayer” on page 559 for fabrication layers.

Examples

```
Dim layerArray(2)

layerArray(0) = 1                            'routing layer 1
layerArray(1) = eengMappedTopSolderPaste     'Top solder paste layer
layerArray(2) = "Fabrication Notes"          'user layer defined in Setup
                                                                Parameters

oODBPPOutputEngine.UserLayerEx (True, "Default User Layer" ) = layerArray
```

PDFConductiveLayer Object

The PDFConductiveLayer Object contains information on conductive items.

Note:
Portable Document Format (PDF) is a file format in which documents may be viewed.

See also PDFConductiveLayers Collection.

Table 23. PDFConductiveLayer Properties

Property	Description
ConductiveItems Property (PDFConductiveLayer Object)	Returns the collection of conductive items.

ConductiveItems Property (PDFConductiveLayer Object)

Object: PDFConductiveLayer Object

Access: Read-Only

Prerequisites: None

Returns the collection of conductive items.

Usage

PDFConductiveLayer.ConductiveItems

Arguments

None

Return Values

ConductiveItems. Returns a ConductiveItems Collection.

Examples

```
set conductiveColl = conductiveLayerObj.ConductiveItems
```

PDFLayerOrder Object

The PDFLayerOrder Object contains information on layers.

Note:
Portable Document Format (PDF) is a file format in which documents may be viewed.

For more information, refer to the **Layer Order** tab and to the **Layer Order** field on the **Sheets Setup** tab of the Extended Print dialog box (**File > Extended Print**).

Table 24. PDFLayerOrder Object Properties

Property	Description
Count Property (PDFLayerOrder Object)	Returns the number of layers in the PDFLayerOrder object.
Layers Property (PDFLayerOrder Object)	Sets or returns the list layers of the PDFLayerOrder. This appends to the current list of layers.

Count Property (PDFLayerOrder Object)

Object:PDFLayerOrder Object

Access: Read-Only

Prerequisites: None

Returns the number of layers in the PDFLayerOrder object.

Usage

```
PDFLayerOrder.Count = Long
```

Arguments

None

Return Values

Long. A long that contains the number of layers.

Layers Property (PDFLayerOrder Object)

Object:PDFLayerOrder Object

Access: Read/Write

Prerequisites: None

Sets or returns the list layers of the PDFLayerOrder. This appends to the current list of layers.

Usage

```
PDFLayerOrder.Layers =  Array
```

Arguments

None

Return Values

Array. An array of layer names.

PDFOutput Object

The PDFOutput Object is used by the ExtendedPrint Object to define the PDF output.

Note:
Portable Document Format (PDF) is a file format in which documents may be viewed.

Refer to ExtendedPrint Object for more information.

Figure 14. PDFOutput Data Model

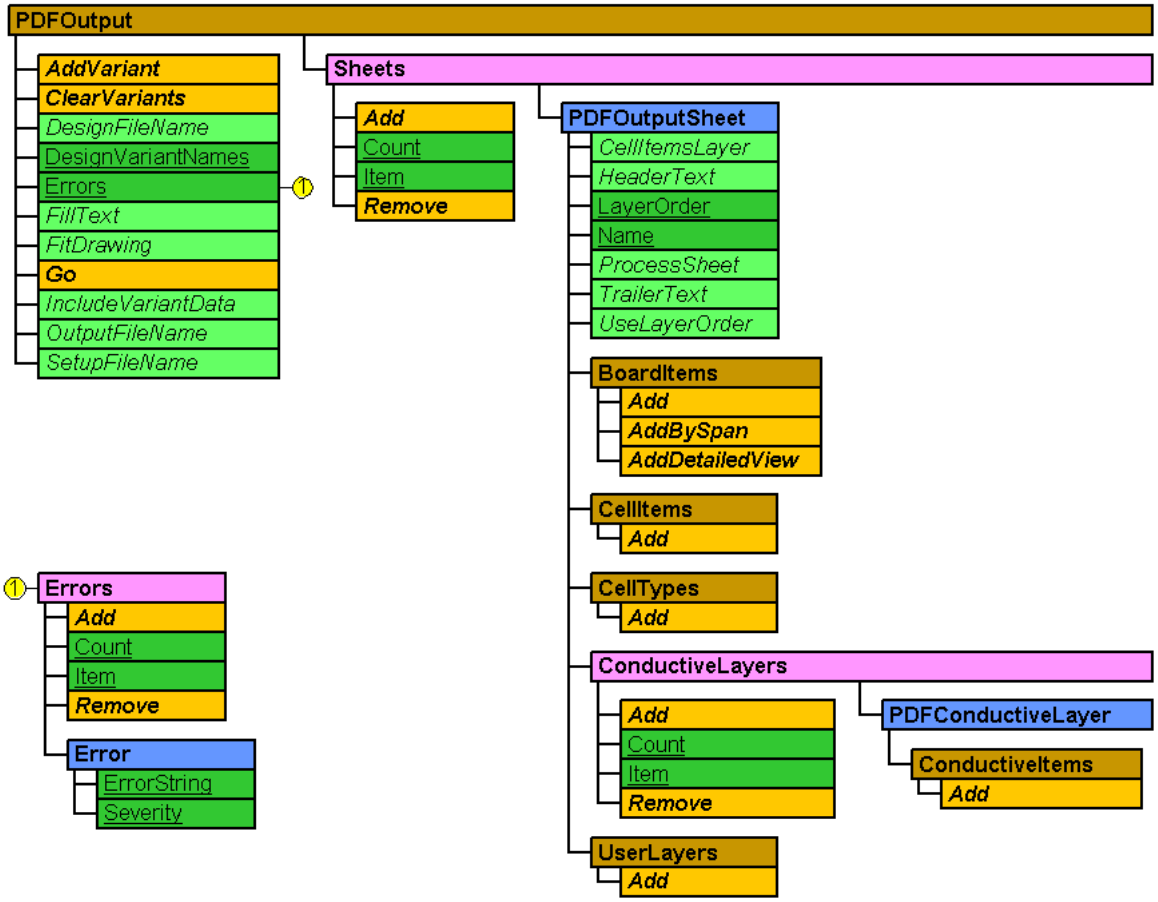


Table 25. PDFOutput Object Methods and Properties

Method or Property	Description
AddVariant Method (PDFOutput Object)	Adds a design variant to the output variant list.
ClearVariants Method (PDFOutput Object)	Clears the output variant list.
Go Method (PDFOutput Object)	Generates the PDF data.
DesignFileName Property (PDFOutput Object)	Sets the name of the design file, including path.

Table 25. PDFOutput Object Methods and Properties (continued)

Method or Property	Description
DesignVariantNames Property (PDFOutput Object)	Returns the variant names for the current design.
Errors Property (PDFOutput Object)	Returns the collection of errors.
FillText Property (PDFOutput Object)	Sets the option for Fill Text, which allows fonts to be filled.
FitDrawing Property (PDFOutput Object)	Sets the option for Fit Drawing.
IncludeVariantData Property (PDFOutput Object)	Includes variant data (if it exists) if set to True. If no variant is specified, the property outputs the Master design.
OutputFileName Property (PDFOutput Object)	Sets the name of the PDF output file, including path.
SetupFileName Property (PDFOutput Object)	Sets the name of the PDF setup file, including path. This is the .pcf file that the PDF Output engine uses to process output data and to load the dialog box.
Sheets Property (PDFOutput Object)	Returns the collection of PDF output sheets.

AddVariant Method (PDFOutput Object)

Object: PDFOutput Object

Prerequisites: None

Adds a design variant to the output variant list.

Usage

```
PDFOutput.AddVariant
        [ByVal variantName As String = “All”])
```

Arguments

- variantName

  (Optional) A string that contains the variant name. If not specified, the method adds all variant names to the output list.

ClearVariants Method (PDFOutput Object)

Object: PDFOutput Object

Prerequisites: None

Clears the output variant list.

Usage

```
PDFOutput.ClearVariants()
```

Arguments

None

Go Method (PDFOutput Object)

Object: PDFOutput Object

Prerequisites: None

Generates the PDF data.

Usage

PDFOutput.Go ()

Arguments

None

Examples

```
' Run the PDF Engine
oPDFEngine.Go
```

DesignFileName Property (PDFOutput Object)

Object: PDFOutput Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

PDFOutput.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the design file name.

Examples

```
oPDFEngine.DesignFileName = jobName
```

DesignVariantNames Property (PDFOutput Object)

Object: PDFOutput Object

Access: Read-Only

Prerequisites: None

Returns the variant names for the current design.

Usage

```
PDFOutput.DesignVariantNames
```

Arguments

None

Return Values

Array. An array of variant names.

Errors Property (PDFOutput Object)

Object: PDFOutput Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

PDFOutput.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oPDFEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

FillText Property (PDFOutput Object)

Object: PDFOutput Object

Access: Write-Only

Prerequisites: None

Sets the option for Fill Text, which allows fonts to be filled.

Usage

```
PDFOutput.FillText = True | False
```

Arguments

None

Return Values

Boolean. If True, the fonts are filled. If False, the fonts are not filled.

FitDrawing Property (PDFOutput Object)

Object: PDFOutput Object

Access: Write-Only

Prerequisites: None

Sets the option for Fit Drawing.

Usage

```
PDFOutput.FitDrawing = True | False
```

Arguments

None

Return Values

Boolean. If True, fit and output the whole drawing sheet. If False, output the current window view.

IncludeVariantData Property (PDFOutput Object)

Object: PDFOutput Object

Access: Write-Only

Prerequisites: None

Includes variant data (if it exists) if set to True. If no variant is specified, the property outputs the Master design.

Usage

```
PDFOutput.IncludeVariantData = True | False
```

Arguments

None

Return Values

Boolean. If True, include variant data. If False, do not include variant data.

OutputFileName Property (PDFOutput Object)

Object: PDFOutput Object

Access: Write-Only

Prerequisites: None

Sets the name of the PDF output file, including path.

Usage

PDFOutput.OutputFileName = String

Arguments

None

Return Values

String. A string that contains the PDF output file name.

Examples

```
oPDFEngine.OutputFileName = docObj.Path & "Output\PDF\" & docObj.Name &
 ".pdf" ' This is the name and location of the output pdf
```

SetupFileName Property (PDFOutput Object)

Object: PDFOutput Object

Access: Write-Only

Prerequisites: None

Sets the name of the PDF setup file, including path. This is the *.pcf* file that the PDF Output engine uses to process output data and to load the dialog box.

Usage

PDFOutput.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the PDF setup file name.

Examples

```
oPDFEngine.SetupFileName = jobConfigPath + "PDFSetup.pcf"
```

Sheets Property (PDFOutput Object)

Object: PDFOutput Object

Access: Read-Only

Prerequisites: None

Returns the collection of PDF output sheets.

Usage

PDFOutput.Sheets

Arguments

None

Return Values

Type. Return a PDFOutputSheets Collection.

Examples

```
Dim numsh 'As Integer
Dim sheet 'As PDFOutputSheet Object
Dim oPDFOutputSheets 'As PDFOutputSheets Collection
Set oPDFOutputSheets = oPDFEngine.Sheets

'Add all the copper layers
For numsh = 1 To docObj.LayerCount
 oPDFOutputSheets.Add "Layer" & numsh
Next
```

PDFOutputSheet Object

The PDFOutputSheet Object contains information on the PDF output sheet.

Note:
Portable Document Format (PDF) is a file format in which documents may be viewed.

See also PDFOutputSheets Collection.

Table 26. PDFOutputSheet Object Properties

Property	Description
BoardItems Property (PDFOutputSheet Object)	Returns the collection of board items.
CellItems Property (PDFOutputSheet Object)	Returns the collection of cell items.
CellItemsLayer Property (PDFOutputSheet Object)	Sets the layer of the board for cell fabrication data items.
CellTypes Property (PDFOutputSheet Object)	Returns the collection of cell types.
ConductiveLayers Property (PDFOutputSheet Object)	Returns the collection of conductive layers.
HeaderText Property (PDFOutputSheet Object)	Sets the header text to add to the PDF output sheet.
LayerOrder Property (PDFOutputSheet Object)	Returns the LayerOrder object that contains the layer order of the sheet.
Mirror Property (PDFOutputSheet Object)	Sets or returns the mirror value for the sheet to specify whether the sheet should be mirrored.
Name Property (PDFOutputSheet Object)	Sets or returns the PDF output sheet name. In Xpedition FabLink Drawing Editor, the Name is read-only.
ProcessSheet Property (PDFOutputSheet Object)	Sets the option to process sheet on the PDF output.
TrailerText Property (PDFOutputSheet Object)	Sets the trailer text to add to the PDF output sheet.
UserLayerOrder Property (PDFOutputSheet Object)	Returns the user layer order in the output.
UserLayers Property (PDFOutputSheet Object)	Returns the collection of user layers.

BoardItems Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Read-Only

Prerequisites: None

Returns the collection of board items.

Usage

PDFOutputSheet.BoardItems

Arguments

None

Return Values

BoardItems. Returns a BoardItems Collection.

Examples

```
' Add all board items individually
Set oBoardItems = sheet.BoardItems
oBoardItems.Add eengBoardItemBoardObstruct
oBoardItems.Add eengBoardItemBoardOrigin
oBoardItems.Add eengBoardItemBoardOutline
```

CellItems Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Read-Only

Prerequisites: None

Returns the collection of cell items.

Usage

PDFOutputSheet.CellItems

Arguments

None

Return Values

CellItems. Returns a CellItems Collection.

Examples

```
' Add all cell items individually
sheet.CellItems.Add eengCellItemAssemblyOutline
sheet.CellItems.Add eengCellItemAssemblyPartNumber
sheet.CellItems.Add eengCellItemAssemblyReferenceDesignator
```

CellItemsLayer Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Write-Only

Prerequisites: None

Sets the layer of the board for cell fabrication data items.

Usage

PDFOutputSheet.CellItemsLayer = Long

Arguments

None

Return Values

Long. A long that specifies the layer of the board for cell fabrication data items.

Examples

```
sheet.CellItemsLayer = 1
```

CellTypes Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Read-Only

Prerequisites: None

Returns the collection of cell types.

Usage

PDFOutputSheet.CellTypes

Arguments

None

Return Values

CellTypes. Returns a CellTypes Collection.

Examples

```
' Add all cell types individually
sheet.CellTypes.Add eengCellTypeBadBoardIdentifier
sheet.CellTypes.Add eengCellTypeBreakAwayTab
sheet.CellTypes.Add eengCellTypeBuried
```

ConductiveLayers Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Read-Only

Prerequisites: None

Returns the collection of conductive layers.

Usage

PDFOutputSheet.ConductiveLayers

Arguments

None

Return Values

PDFConductiveLayers. Returns a PDFConductiveLayers Collection.

Examples

```
Set oCondLayer = sheet.ConductiveLayers.Add(i)
' Add all conductive items individually
oCondLayer.ConductiveItems.Add eengCondItemActualPlaneShapes
oCondLayer.ConductiveItems.Add eengCondItemCopperBalancing
oCondLayer.ConductiveItems.Add eengCondItemEtchedText
```

HeaderText Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Write-Only

Prerequisites: None

Sets the header text to add to the PDF output sheet.

Usage

PDFOutputSheet.HeaderText = String

Arguments

None

Return Values

String. The header text to add to the PDF output sheet.

Examples

```
Dim jn
jn = (jobName, "\", "/")
sheet.HeaderText = jn
```

LayerOrder Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Read-Only

Prerequisites: None

Returns the LayerOrder object that contains the layer order of the sheet.

Usage

```
PDFOutputSheet.LayerOrder
```

Arguments

None

Return Values

PDFLayerOrder. Returns the PDFLayerOrder Object of the sheet.

Mirror Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Read/Write

Prerequisites: None

Sets or returns the mirror value for the sheet to specify whether the sheet should be mirrored.

Usage

```
PDFOutputSheet.Mirror = True | False
```

Arguments

None

Return Values

Boolean. If True, mirror the sheet; otherwise, do not mirror the sheet.

Name Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Read/Write

Prerequisites: None

Sets or returns the PDF output sheet name. In Xpedition FabLink Drawing Editor, the Name is read-only.

Usage

PDFOutputSheet.Name = String

Arguments

None

Return Values

String. Specifies the PDF output sheet name.

Examples

```
sheet.Name = “Sheet3”
```

ProcessSheet Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Read/Write

Prerequisites: None

Sets the option to process sheet on the PDF output.

Usage

```
PDFOutputSheet.ProcessSheet = True | False
```

Arguments

None

Return Values

Boolean. If True, process sheet. If False, do not process sheet.

TrailerText Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Write-Only

Prerequisites: None

Sets the trailer text to add to the PDF output sheet.

Usage

PDFOutputSheet.TrailerText = String

Arguments

None

Return Values

String. Specifies the trailer text to add to the PDF output sheet.

Examples

```
sheet.TrailerText = “End of Page”
```

UserLayerOrder Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Write-Only

Prerequisites: None

Returns the user layer order in the output.

Usage

```
PDFOutputSheet.UserLayerOrder = True | False
```

Arguments

None

Return Values

Boolean. If True, include user layer data. If False, do not include user layer data.

UserLayers Property (PDFOutputSheet Object)

Object: PDFOutputSheet Object

Access: Read-Only

Prerequisites: None

Returns the collection of user layers.

Usage

PDFOutputSheet.UserLayers

Arguments

None

Return Values

UserLayers. Returns a UserLayers Collection.

Examples

```
Set UserLayersColl = sheet.UserLayers
```

SilkscreenGen Object

The SilkscreenGen Object contains parameters for the silkscreen generator file.

For more information refer to the Silkscreen Generator dialog box by choosing **Output > Silkscreen Generator**.

Figure 15. SilkscreenGen Data Model

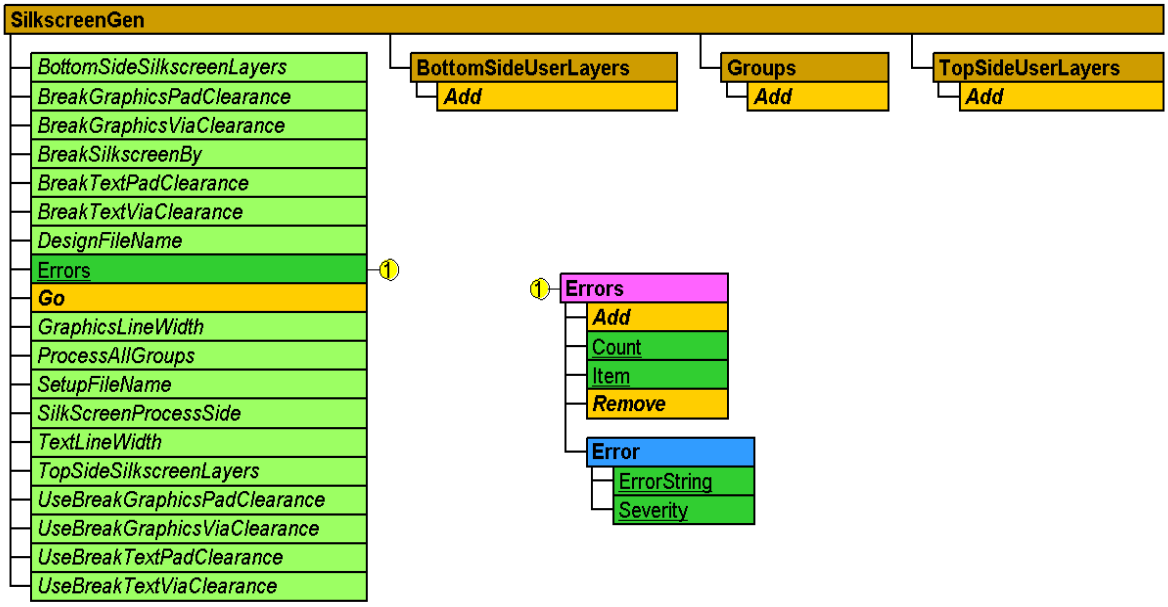


Table 27. SilkscreenGen Object Methods and Properties

Method or Property	Description
Go Method (SilkscreenGen Object)	Generates the silkscreen data.
BottomSideSilkscreenLayers Property (SilkscreenGen Object)	Sets the bottom-side silkscreen layers for the selected package groups.
BottomSideUserLayers Property (SilkscreenGen Object)	Returns a collection of user layers for the bottom side.
BreakGraphicsPadClearance Property (SilkscreenGen Object)	Sets the option to create outline information with clearance for all non-via pads. Using this property automatically sets the UseBreakGraphicsPadClearance property to True.
BreakGraphicsViaClearance Property (SilkscreenGen Object)	Sets the option to create outline information with clearance for all via pads. Using this property automatically sets the UseBreakGraphicsViaClearance property to True.
BreakSilkscreenBy Property (SilkscreenGen Object)	Sets the option to break silkscreen by either Conductive Pads or Soldermask Pads.
BreakTextPadClearance Property (SilkscreenGen Object)	Sets the option to create text with clearance for all non-via pads. Using this property automatically sets the UseBreakTextPadClearance property to True.

Table 27. SilkscreenGen Object Methods and Properties (continued)

Method or Property	Description
BreakTextViaClearance Property (SilkscreenGen Object)	Sets the option to create text with clearance for all via pads. Using this property automatically sets the UseBreakTextViaClearance property to True.
DesignFileName Property (SilkscreenGen Object)	Sets the name of the design file, including path.
Errors Property (SilkscreenGen Object)	Returns the collection of errors.
GraphicsLineWidth Property (SilkscreenGen Object)	Sets the width to use for zero-width graphics when generating silkscreen.
Groups Property (SilkscreenGen Object)	Returns the collection of package or panel groups to process.
ProcessAllGroups Property (SilkscreenGen Object)	Sets the option to process all package groups or selected package groups.
SetupFileName Property (SilkscreenGen Object)	Sets the name of the silkscreen generator file, including path. This is the silkGenDlg.hkp file that the Silkscreen generator uses to process output data and to load the Silkscreen Generator dialog box.
SilkScreenProcessSide Property (SilkscreenGen Object)	Sets the side of the board on which to generate silkscreen information.
TextLineWidth Property (SilkscreenGen Object)	Sets the width to use for zero-width text when generating the silkscreen.
TopSideSilkscreenLayers Property (SilkscreenGen Object)	Sets the top-side silkscreen layers for selected package groups.
TopSideUserLayers Property (SilkscreenGen Object)	Returns a collection of user layers for the top side.
UseBreakGraphicsPadClearance Property (SilkscreenGen Object)	Sets the option to enable or disable Break graphics Pad clearance.
UseBreakGraphicsViaClearance Property (SilkscreenGen Object)	Sets the option to enable or disable Break graphics Via clearance.
UseBreakTextPadClearance Property (SilkscreenGen Object)	Sets the option to enable or disable Break text Pad clearance.
UseBreakTextViaClearance Property (SilkscreenGen Object)	Sets the option to enable or disable Break text Via clearance.

Go Method (SilkscreenGen Object)

Object: SilkscreenGen Object

Prerequisites: None

Generates the silkscreen data.

Usage

SilkscreenGen.Go()

Arguments

None

Examples

```
' Run the engine
oSilkscreenGenEngine.Go()
```

BottomSideSilkscreenLayers Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the bottom-side silkscreen layers for the selected package groups.

Usage

SilkscreenGen.BottomSideSilkscreenLayers = Array

Arguments

None

Return Values

Array. An array of “EEnginesSilkscreenDesignLayer” on page 577.

Examples

```
botArray = Array (eengSilkscreenReferenceDesignators,
 eengSilkscreenOutline)
oSilkscreenGenEngine.BottomSideSilkscreenLayers = botArray
```

BottomSideUserLayers Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Read-Only

Prerequisites: None

Returns a collection of user layers for the bottom side.

Usage

SilkscreenGen.BottomSideUserLayers

Arguments

None

Return Values

IUserLayers. A collection of user layers (“UserLayers” on page 490)

Examples

```
oUserLayers = oSilkscreenGenEngine.BottomSideUserLayers
oUserLayers.Add “USER_LVL_60”
oUserLayers.Add “TP Graphics”
```

BreakGraphicsPadClearance Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to create outline information with clearance for all non-via pads. Using this property automatically sets the UseBreakGraphicsPadClearance property to True.

Usage

SilkscreenGen.BreakGraphicsPadClearance ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  (Optional) The units to use for the pad clearance. If not specified, the current unit is used.

Return Values

Double. A double that contains the value for the pad clearance option.

Examples

```
oSilkscreenGenEngine.BreakGraphicsPadClearance = 10
oSilkscreenGenEngine.UseBreakGraphicsPadClearance = True
```

BreakGraphicsViaClearance Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to create outline information with clearance for all via pads. Using this property automatically sets the UseBreakGraphicsViaClearance property to True.

Usage

SilkscreenGen.BreakGraphicsViaClearance ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  (Optional) The units to use for the via clearance. If not specified, the current unit is used.

Return Values

Double. A double that contains the value for via clearance option.

Examples

```
oSilkscreenGenEngine.BreakGraphicsViaClearance = 10
oSilkscreenGenEngine.UseBreakGraphicsViaClearance = True
```

BreakSilkscreenBy Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to break silkscreen by either Conductive Pads or Soldermask Pads.

Usage

SilkscreenGen.BreakSilkscreenBy = EEnginesSilkScreenBreakType

Arguments

None

Return Values

EEnginesSilkScreenBreakType. The break silkscreen type (“EEnginesSilkScreenBreakType” on page 576)

Examples

```
oSilkscreenGenEngine.BreakSilkscreenBy = eengSolderMaskPads
```

BreakTextPadClearance Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to create text with clearance for all non-via pads. Using this property automatically sets the UseBreakTextPadClearance property to True.

Usage

SilkscreenGen.BreakTextPadClearance ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  (Optional) The unit to use for the pad clearance. If not specified, the current unit is used.

Return Values

Double. A double that contains the value for the pad clearance option.

Examples

```
oSilkscreenGenEngine.BreakTextPadClearance = 10
oSilkscreenGenEngine.UseBreakTextPadClearance = True
```

BreakTextViaClearance Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to create text with clearance for all via pads. Using this property automatically sets the UseBreakTextViaClearance property to True.

Usage

SilkscreenGen.BreakTextViaClearance ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  (Optional) The unit to use for the via clearance. If not specified, the current unit is used.

Return Values

Double. A double that contains the value for the via clearance option.

Examples

```
oSilkscreenGenEngine.BreakTextViaClearance = 10
oSilkscreenGenEngine.UseBreakTextViaClearance = True
```

DesignFileName Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the name of the design file, including path.

Usage

SilkscreenGen.DesignFileName = String

Arguments

None

Return Values

String. A string that contains the design file name.

Examples

```
oSilkscreenGenEngine.DesignFileName = jobName
```

Errors Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Read-Only

Prerequisites: None

Returns the collection of errors.

Usage

SilkscreenGen.Errors

Arguments

None

Return Values

Errors. Returns the Errors Collection.

Examples

```
' Check errors
Dim oErrors, oErrCnt, oErr
oErrors = oSilkscreenGenEngine.Errors
Set oErrCnt = oErrors.Count
For Each oErr in oErrors
 If oErr.Severity = eengError Then
 MsgBox "Error occurred " & oErr.ErrorString
 End If
Next
```

GraphicsLineWidth Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the width to use for zero-width graphics when generating silkscreen.

Usage

SilkscreenGen.GraphicsLineWidth ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  (Optional) The unit to use for the line width. If not specified, the current unit is used.

Return Values

Double. A double that defines the width to use for zero width graphics.

Examples

```
oSilkscreenGenEngine.GraphicsLineWidth = 8
```

Groups Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Read-Only

Prerequisites: None

Returns the collection of package or panel groups to process.

Usage

SilkscreenGen.Groups

Arguments

None

Return Values

ICellTypes. A collection of package or panel groups ().CellTypes

Examples

```
' Set the selected package groups
oSilkscreenGenEngine.Groups.Add eengCellTypeDiscreteChip
oSilkscreenGenEngine.Groups.Add eengCellTypeDiscreteOther
```

ProcessAllGroups Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to process all package groups or selected package groups.

Usage

SilkscreenGen.ProcessAllGroups = True | False

Arguments

None

Return Values

Boolean. If True, process all package groups. If False, process selected package groups.

Examples

```
oSilkscreenGenEngine.ProcessAllGroups = True
```

SetupFileName Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the name of the silkscreen generator file, including path. This is the *silkGenDlg.hkp* file that the Silkscreen generator uses to process output data and to load the Silkscreen Generator dialog box.

Usage

SilkscreenGen.SetupFileName = String

Arguments

None

Return Values

String. A string that contains the setup file name.

Examples

```
oSilkscreenGenEngine.SetupFileName = “mydesign/pcb/config/silkGenDlg.hkp”
```

SilkScreenProcessSide Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the side of the board on which to generate silkscreen information.

Usage

SilkscreenGen.SilkScreenProcessSide = EEnginesCellSide

Arguments

None

Return Values

EEnginesCellSide. Defines the side to process. EEnginesCellSideBoth, Top or Bottom are valid options.

Examples

```
oSilkscreenGenEngine.SilkScreenProcessSide = eengCellSideBoth
```

TextLineWidth Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the width to use for zero-width text when generating the silkscreen.

Usage

SilkscreenGen.TextLineWidth ([ByVal eUnit As EEnginesUnit = eengUnitCurrent]) = Double

Arguments

- eUnit

  (Optional) The unit to use for the line width. If not specified, the current unit is used.

Return Values

Double. A double that defines the width to use for zero-width text.

Examples

```
oSilkscreenGenEngine.TextLineWidth = 8
```

TopSideSilkscreenLayers Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the top-side silkscreen layers for selected package groups.

Usage

SilkscreenGen.TopSideSilkscreenLayers = Array

Arguments

None

Return Values

Array. An array of “EEnginesSilkscreenDesignLayer” on page 577.

Examples

```
topArray = Array (eengSilkscreenOutline)
oSilkscreenGenEngine.TopSideSilkscreenLayers = topArray
```

TopSideUserLayers Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Read-Only

Prerequisites: None

Returns a collection of user layers for the top side.

Usage

SilkscreenGen.TopSideUserLayers

Arguments

None

Return Values

IUserLayers. A collection of user layers (“UserLayers” on page 490)

Examples

```
oUserLayers = oSilkscreenGenEngine.TopSideUserLayers
oUserLayers.Add “USER_LVL_60”
oUserLayers.Add “TP Graphics”
```

UseBreakGraphicsPadClearance Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to enable or disable Break graphics Pad clearance.

Usage

SilkscreenGen.UseBreakGraphicsPadClearance = True | False

Arguments

None

Return Values

Boolean. If True, the Break graphics Pad clearance option is set. If False, the option is not set.

Examples

```
oSilkscreenGenEngine.BreakGraphicsPadClearance = 10
oSilkscreenGenEngine.UseBreakGraphicsPadClearance = True
```

UseBreakGraphicsViaClearance Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to enable or disable Break graphics Via clearance.

Usage

SilkscreenGen.UseBreakGraphicsViaClearance = True | False

Arguments

None

Return Values

Boolean. If True, the Break graphics Via clearance option is set. If False, the option is not set.

Examples

```
oSilkscreenGenEngine.BreakGraphicsViaClearance = 10
oSilkscreenGenEngine.UseBreakGraphicsViaClearance = True
```

UseBreakTextPadClearance Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to enable or disable Break text Pad clearance.

Usage

SilkscreenGen.UseBreakTextPadClearance = True | False

Arguments

None

Return Values

Boolean. If True, the Break text Pad clearance option is set. If False, the option is not set.

Examples

```
oSilkscreenGenEngine.BreakTextPadClearance = 10
oSilkscreenGenEngine.UseBreakTextPadClearance = True
```

UseBreakTextViaClearance Property (SilkscreenGen Object)

Object: SilkscreenGen Object

Access: Write-Only

Prerequisites: None

Sets the option to enable or disable Break text Via clearance.

Usage

SilkscreenGen.UseBreakTextViaClearance = True | False

Arguments

None

Return Values

Boolean. If True, the Break text Via clearance option is set. If False, the option is not set.

Examples

```
oSilkscreenGenEngine.BreakTextViaClearance = 10
oSilkscreenGenEngine.UseBreakTextViaClearance = True
```

UserInformation Object

The UserInformation Object contains information about the user who creates the manufacturing outputs.

See also UserInformation Property (ManufacturingOutputValidation Object).

Table 28. UserInformation Object Properties

Property	Description
Address Property (UserInformation Object)	Set the address of the user who creates the manufacturing outputs.
CompanyName Property (UserInformation Object)	Sets the company name of the user who creates the manufacturing outputs.
EmailAddress Property (UserInformation Object)	Sets the email address of the user who creates the manufacturing outputs.
FaxNumber Property (UserInformation Object)	Sets the fax number of the user who creates the manufacturing outputs.
FileName Property (UserInformation Object)	Sets the name of the MOV user information file, including path.
PhoneNumber Property (UserInformation Object)	Sets the phone number of the user who creates the manufacturing outputs.
UserName Property (UserInformation Object)	Sets the name of the user who creates the manufacturing outputs.

Address Property (UserInformation Object)

Object: UserInformation Object

Access: Write-Only

Prerequisites: None

Set the address of the user who creates the manufacturing outputs.

Usage

UserInformation.Address = String

Arguments

None

Return Values

String. A string that specifies the address of the user who creates the manufacturing outputs.

Examples

```
oUserInfo.Address = "5030 Bradford Drive"
oUserInfo.Address = "Huntsville, AL 35805"
```

CompanyName Property (UserInformation Object)

Object: UserInformation Object

Access: Write-Only

Prerequisites: None

Sets the company name of the user who creates the manufacturing outputs.

Usage

UserInformation.CompanyName = String

Arguments

None

Return Values

String. A string that specifies the company name of the user who creates the manufacturing outputs.

Examples

```
oUserInfo.CompanyName = "Mentor Graphics"
```

EmailAddress Property (UserInformation Object)

Object: UserInformation Object

Access: Write-Only

Prerequisites: None

Sets the email address of the user who creates the manufacturing outputs.

Usage

UserInformation.EmailAddress = String

Arguments

None

Return Values

String. A string that specifies the email address of the user who creates the manufacturing outputs.

Examples

```
oUserInfo.EmailAddress = "mov_user@mentor.com"
```

FaxNumber Property (UserInformation Object)

Object: UserInformation Object

Access: Write-Only

Prerequisites: None

Sets the fax number of the user who creates the manufacturing outputs.

Usage

UserInformation.FaxNumber = String

Arguments

None

Return Values

String. A string that specifies the fax number of the user who creates the manufacturing outputs.

Examples

```
oUserInfo.FaxNumber = "(256)864-0001"
```

FileName Property (UserInformation Object)

Object: UserInformation Object

Access: Write-Only

Prerequisites: None

Sets the name of the MOV user information file, including path.

Usage

UserInformation.FileName = String

Arguments

None

Return Values

String. A string that contains the MOV user information file name, including path.

Examples

```
oUserInfo.FileName = jobConfigPath + "MovUser.HKP"
```

PhoneNumber Property (UserInformation Object)

Object: UserInformation Object

Access: Write-Only

Prerequisites: None

Sets the phone number of the user who creates the manufacturing outputs.

Usage

UserInformation.PhoneNumber = String

Arguments

None

Return Values

String. A string that specifies the phone number of the user who creates the manufacturing outputs.

Examples

```
oUserInfo.PhoneNumber = "(256)864-0000"
```

UserName Property (UserInformation Object)

Object: UserInformation Object

Access: Write-Only

Prerequisites: None

Sets the name of the user who creates the manufacturing outputs.

Usage

UserInformation.UserName = String

Arguments

None

Return Values

String. A string that specifies the name of the user who creates the manufacturing outputs.

Examples

```
oUserInfo.UserName = "MOV User"
```

Chapter 3
Batch Engines Collections

The Automation Pro Batch Engines collections enables you to work with multiple objects of the same type.

Batch Engines Collection Descriptions

Batch Engines Collection Descriptions

The Automation Pro batch engines collections are organized by object type.

Refer to a specific collection for information about the methods and properties available to that collection.

Table 29. Batch Engine Collections

Collection	Description
“BoardItems Collection” on page 449	This is a collection of EEnginesBoardItem Enums.
“CellItems Collection” on page 455	This is a collection of EEnginesCellItem Enums.
“CellTypes Collection” on page 457	This is a collection of EEnginesCellType Enums.
“ConductiveItems Collection” on page 459	This is a collection of EEnginesConductiveItem Enums.
“DrillHoles Collection” on page 461	This is a collection of DrillHole Objects.
“Errors Collection” on page 465	This is a collection of Error Objects.
“GerberOutputFiles Collection” on page 470	This is a collection of GerberOutputFile Objects.
“NCDrillChartColumns Collection” on page 475	This is a collection of EEnginesNCDrillColumn Enums.
“PDFConductiveLayers Collection” on page 480	This is a collection of PDFConductiveLayer Objects.
“PDFOutputSheets Collection” on page 485	This is a collection of PDFOutputSheet on page 387 objects.
“UserLayers Collection” on page 490	This is a collection of layers.

BoardItems Collection

This section defines the methods and properties of the BoardItems collection. This is a collection of BoardItem enums.

See also BoardItems Property (GerberOutputFile Object) and BoardItems Property (PDFOutputSheet Object).

Table 30. BoardItems Collection Methods

Method	Description
Add Method (BoardItems Collection)	Adds a BoardItem to the BoardItems collection.
AddBySpan Method (BoardItems Collection)	Adds a contour or cavity item and layer spans. Specify 0 for fromLayer and toLayer to create a Through Board contour.
AddDetailedView Method (BoardItems Collection)	Adds a detailed view by name.
AddFlexLayer Method (BoardItems Collection)	Adds flex layer to the output.
AddFlexPadLayer Method (BoardItems Collection)	Adds a flex pad layer to the output.

Add Method (BoardItems Collection)

Collection: BoardItems Collection

Prerequisites: None

Adds a BoardItem to the BoardItems collection.

Usage

BoardItems.Add (By Val eBoardItem As EEnginesBoardItem)

Arguments

- eBoardItem

  The EEnginesBoardItem Enum to add to the collection.

Examples

```
oBoardItems.Add eengBoardItemGeneratedSilkscreenTop
oBoardItems.Add eengBoardItemBoardOrigin
oBoardItems.Add eengBoardItemBoardOutline
```

AddBySpan Method (BoardItems Collection)

Collection: BoardItems Collection

Prerequisites: None

Adds a contour or cavity item and layer spans. Specify 0 for fromLayer and toLayer to create a Through Board contour.

Usage

```
BoardItems.AddBySpan (
        ByVal eBoardItem As EEnginesBoardItem,
        ByVal fromLayer As Long,
        ByVal toLayer As Long)
```

Arguments

- eBoardItem

  The EEnginesBoardItem Enum to add to the collection.
- fromLayer

  A long that contains the start layer.
- toLayer

  A long that contains the end layer.

AddDetailedView Method (BoardItems Collection)

Collection: BoardItems Collection

Prerequisites: None

Adds a detailed view by name.

Usage

```
BoardItems.AddDetailedView (
        ByVal sDetailedViewName As String)
        Description
```

Arguments

- eDetailedViewName

  A string that contains the detailed view name.

AddFlexLayer Method (BoardItems Collection)

Prerequisites: None

Object: BoardItems Collection

Adds flex layer to the output.

Usage

```
BoardItems.AddFlexLayer(ByVal sFlexLayer As String) As Nothing
```

Arguments

- sFlexLayer

  A string that defines the name of the flex layer.

AddFlexPadLayer Method (BoardItems Collection)

Prerequisites: None

Object: BoardItems Collection

Adds a flex pad layer to the output.

Usage

```
BoardItems.AddFlexPadLayer(ByVal sFlexLayer As String) As Nothing
```

Arguments

- sFlexLayer

  A string that defines the name of the flex pad layer.

CellItems Collection

Defines the methods and properties CellItems collection.

This is a collection of EEnginesCellItem Enums.

See also CellItems Property (GerberOutputFile Object) and CellItems Property (PDFOutputSheet Object).

Table 31. CellItems Collection Methods

Method	Description
Add Method (CellItems Collection)	Adds a CellItem to the CellItems collection.

Add Method (CellItems Collection)

Collection: CellItems Collection

Prerequisites: None

Adds a CellItem to the CellItems collection.

Usage

CellItems.Add (ByVal eCellItem As EEnginesCellItem)

Arguments

- eCellItem

  The (EEnginesCellItem Enum) to add to the collection.

Examples

```
' Add all cell items individually
sheet.CellItems.Add eengCellItemAssemblyOutline
sheet.CellItems.Add eengCellItemAssemblyPartNumber
sheet.CellItems.Add eengCellItemAssemblyReferenceDesignator
```

CellTypes Collection

Defines the methods and properties of a CellTypes collection.

This is a collection of EEnginesCellType Enums.

See also CellTypes Property (GerberOutputFile Object), CellTypes Property (PDFOutputSheet Object) and Groups Property (SilkscreenGen Object).

Table 32. CellTypes Collection Methods

Method	Description
Add Method (CellTypes Collection)	Adds a CellType to the CellTypes collection.

Add Method (CellTypes Collection)

Collection: CellTypes Collection

Prerequisites: None

Adds a CellType to the CellTypes collection.

Usage

CellTypes.Add (ByVal eCellType As EEnginesCellType)

Arguments

- eCellType

  The (EEnginesCellType Enum) to add to the collection.

Examples

```
sheet.CellTypes.Add eengCellTypeBuried
sheet.CellTypes.Add eengCellTypeConnector
sheet.CellTypes.Add eengCellTypeDiscreteAxial
```

ConductiveItems Collection

Defines the methods and properties of a ConductiveItems collection.

This is a collection of EEnginesConductiveItem Enums.

See also ConductiveItems Property (PDFConductiveLayer Object).

Table 33. ConductiveItems Collection Methods

Method	Description
Add Method (ConductiveItems Collection)	Adds a ConductiveItem type to the ConductiveItems collection.

Add Method (ConductiveItems Collection)

Collection: ConductiveItems Collection

Prerequisites: None

Adds a ConductiveItem type to the ConductiveItems collection.

Usage

ConductiveItems.Add (ByVal eConductiveItem As EEnginesConductiveItem)

Arguments

- eConductiveItem

  The (EEnginesConductiveItem Enum) to add to the collection.

Examples

```
oCondLayer.ConductiveItems.Add eengCondItemActualPlaneShapes
 oCondLayer.ConductiveItems.Add eengCondItemCopperBalancing
 oCondLayer.ConductiveItems.Add eengCondItemEtchedText
```

DrillHoles Collection

Defines the methods and properties of a DrillHoles collection.

This is a collection of DrillHole Objects.

See also DrillHoles Property (NCDrillChart Object).

Table 34. DrillHoles Collection Methods and Properties

Method or Property	Description
Remove Method (DrillHoles Collection)	Removes a DrillHole object from the DrillHoles collection.
Count Property (DrillHoles Collection)	Returns the number of DrillHole objects contained in the DrillHoles collection.
Item Property (DrillHoles Collection)	Returns a DrillHole object contained in the DrillHoles collection.

Remove Method (DrillHoles Collection)

Collection: DrillHoles Collection

Prerequisites: None

Removes a DrillHole object from the DrillHoles collection.

Usage

```
DrillHoles.Remove (ByVal vIndex As Variant)
```

Arguments

- vIndex

  The index of the DrillHole Object to remove from the collection.

Count Property (DrillHoles Collection)

Collection: DrillHoles Collection

Access: Read-Only

Prerequisites: None

Returns the number of DrillHole objects contained in the DrillHoles collection.

Usage

```
DrillHoles.Count
```

Arguments

None

Return Values

Long. The number of DrillHole Objects the collection contains.

Item Property (DrillHoles Collection)

Collection: DrillHoles Collection

Access: Read-Only

Prerequisites: None

Returns a DrillHole object contained in the DrillHoles collection.

Usage

```
DrillHoles.Item (ByVal vIndex As Variant)
```

Arguments

- vIndex

  Index of the object to retrieve from the collection.

Return Values

DrillHole. The DrillHole Object located at position vIndex.

Errors Collection

Defines the methods and properties of an Errors collection.

This is a collection of Error Objects.

See also:

- Errors Property (DFFDRC Object)
- Errors Property (DRC Object)
- Errors Property (Gerber Object)
- Errors Property (ManufacturingOutputValidation Object)
- Errors Property (NCDrill Object)
- Errors Property (PDFOutput Object)
- Errors Property (SilkscreenGen Object)

Table 35. Errors Collection Methods and Properties

Method or Property	Description
Add Method (Errors Collection)	Adds an Error object to the Errors collection.
Remove Method (Errors Collection)	Removes an Error object from the Errors collection.
Count Property (Errors Collection)	Returns the number of Error objects contained in the Errors collection.
Item Property (Errors Collection)	Returns an Error object contained in the Errors collection.

Add Method (Errors Collection)

Collection: Errors Collection

Prerequisites: None

Adds an Error object to the Errors collection.

Usage

Errors.Add (ByVal pError As IError)

Arguments

- pError

  The (Error Object) to add to the collection.

Examples

```
ErrorsColl..Add errorObj
```

Remove Method (Errors Collection)

Collection: Errors Collection

Prerequisites: None

Removes an Error object from the Errors collection.

Usage

Errors.Remove (ByVal vIndex As Variant)

Arguments

- vIndex

  The index of the (Error Object) to remove from the collection.

Examples

```
ErrorsColl.Remove(i)
```

Count Property (Errors Collection)

Collection: Errors Collection

Access: Read-Only

Prerequisites: None

Returns the number of Error objects contained in the Errors collection.

Usage

Errors.Count

Arguments

None

Return Values

Long. The number of (Error Object) the collection contains.

Examples

```
errCnt = ErrorsColl.Count
```

Item Property (Errors Collection)

Collection: Errors Collection

Access: Read-Only

Prerequisites: None

Returns an Error object contained in the Errors collection.

Usage

Errors.Item (ByVal vIndex As Variant)

Arguments

- vIndex

  Index of the object to retrieve from the collection.

Return Values

Error. The (Error Object) located at position vIndex.

Examples

```
Set errorObj = errorsColl.Item(i)
```

GerberOutputFiles Collection

Defines the methods and properties of a GerberOutputFiles collection.

This is a collection of GerberOutputFile Objects. Refer to the Gerber Output dialog box, Contents tab, Output files, File Name column.

See also OutputFiles Property (Gerber Object).

Table 36. GerberOutputFiles Collection Methods and Properties

Method or Property	Description
Add Method (GerberOutputFiles Collection)	Adds a GerberOutputFile object to the GerberOutputFiles collection.
Remove Method (GerberOutputFiles Collection)	Removes a GerberOutputFile object from the GerberOutputFiles collection.
Count Property (GerberOutputFiles Collection)	Returns the number of GerberOutputFile objects contained in the GerberOutputFiles collection.
Item Property (GerberOutputFiles Collection)	Returns a GerberOutputFile object contained in the GerberOutputFiles collection.

Add Method (GerberOutputFiles Collection)

Collection: GerberOutputFiles Collection

Prerequisites: None

Adds a GerberOutputFile object to the GerberOutputFiles collection.

Usage

GerberOutputFiles.Add (ByVal bFileName As String) As IGerberOutputFile

Arguments

- bFileName

  The (GerberOutputFile Object) to add to the collection.

Examples

```
oGOFAddEach = oGerberOutputFiles.Add("EachElements.gdo")
```

Remove Method (GerberOutputFiles Collection)

Collection: GerberOutputFiles Collection

Prerequisites: None

Removes a GerberOutputFile object from the GerberOutputFiles collection.

Usage

GerberOutputFiles.Remove (ByVal vIndex As Variant)

Arguments

- vIndex

  The index of the (GerberOutputFile Object) to remove from the collection.

Examples

```
OutputFilesColl.Remove(i)
```

Count Property (GerberOutputFiles Collection)

Collection: GerberOutputFiles Collection

Access: Read-Only

Prerequisites: None

Returns the number of GerberOutputFile objects contained in the GerberOutputFiles collection.

Usage

GerberOutputFiles.Count

Arguments

None

Return Values

Long. The number of (GerberOutputFile Objects) the collection contains.

Examples

```
OutputFileCnt = OutputFileColl.Count
```

Item Property (GerberOutputFiles Collection)

Collection: GerberOutputFiles Collection

Access: Read-Only

Prerequisites: None

Returns a GerberOutputFile object contained in the GerberOutputFiles collection.

Usage

GerberOutputFiles.Item (ByVal vIndex As Variant)

Arguments

- vIndex

  Index of the object to retrieve from the collection.

Return Values

GerberOutputFile. The (GerberOutputFile Object) located at position vIndex.

Examples

```
Set outputFileObj = OutputFilesColl.Item(i)
```

NCDrillChartColumns Collection

Defines the methods and properties of an NCDrillChartColumns collection.

This is a collection of EEnginesNCDrillColumn Enums. For more information, refer to the Drill Chart Options tab, Columns option, of the NC Drill Generation dialog box.

See also Columns Property (NCDrillChart Object).

Table 37. NCDrillChartColumns Collection Methods and Properties

Method or Property	Description
Add Method (NCDrillChartColumns Collection)	Adds a new column to the NCDrillChartColumns collection. Refer to the Drill Chart Options tab, Columns option, of the NC Drill Generation dialog box.
Remove Method (NCDrillChartColumns Collection)	Removes an NCDrillChartColumn object from the NCDrillChartColumns collection.
Count Property (NCDrillChartColumns Collection)	Returns the number of NCDrillChartColumn objects contained in the NCDrillChartColumns collection.
Item Property (NCDrillChartColumns Collection)	Returns an NCDrillChartColumn object contained in the NCDrillChartColumns collection.

Add Method (NCDrillChartColumns Collection)

Collection: NCDrillChartColumns Collection

Prerequisites: None

Adds a new column to the NCDrillChartColumns collection. Refer to the Drill Chart Options tab, Columns option, of the NC Drill Generation dialog box.

Usage

NCDrillChartColumns.Add (ByVal eColumn As EEnginesNCDrillColumn)

Arguments

- eColumn

  The (EEnginesNCDrillColumn Enum) to add to the collection.

Examples

```
oNCDrillChartColumns.Add eengColumnSymbol
oNCDrillChartColumns.Add eengColumnDiameter
oNCDrillChartColumns.Add eengColumnTolerance
```

Remove Method (NCDrillChartColumns Collection)

Collection: NCDrillChartColumns Collection

Prerequisites: None

Removes an NCDrillChartColumn object from the NCDrillChartColumns collection.

Usage

```
NCDrillChartColumns.Remove (ByVal vIndex As Variant)
```

Arguments

- vIndex

  The index of the (NCDrillChartColumn Object) to remove from the collection.

Count Property (NCDrillChartColumns Collection)

Collection: NCDrillChartColumns Collection

Access: Read-Only

Prerequisites: None

Returns the number of NCDrillChartColumn objects contained in the NCDrillChartColumns collection.

Usage

```
NCDrillChartColumns.Count
```

Arguments

None

Return Values

Long. The number of (NCDrillChartColumn Objects) the collection contains.

Item Property (NCDrillChartColumns Collection)

Collection: NCDrillChartColumns Collection

Access: Read-Only

Prerequisites: None

Returns an NCDrillChartColumn object contained in the NCDrillChartColumns collection.

Usage

```
NCDrillChartColumns.Item (ByVal vIndex As Variant)
```

Arguments

- vIndex

  Index of the object to retrieve from the collection.

Return Values

NCDrillChartColumn. The object (NCDrillChartColumn Object) located at position vIndex.

PDFConductiveLayers Collection

Defines the methods and properties of a PDFConductiveLayers collection.

This is a collection of PDFConductiveLayer Objects.

See also ConductiveLayers Property (PDFOutputSheet Object).

Table 38. PDFConductiveLayers Collection Methods and Properties

Method or Property	Description
Add Method (PDFConductiveLayers Collection)	Adds a PDFConductiveLayer to the PDFConductiveLayers collection.
Remove Method (PDFConductiveLayers Collection)	Removes a PDFConductiveLayer object from the PDFConductiveLayers collection.
Count Property (PDFConductiveLayers Collection)	Returns the number of PDFConductiveLayer objects contained in the PDFConductiveLayers collection.
Item Property (PDFConductiveLayers Collection)	Returns a PDFConductiveLayer object contained in the PDFConductiveLayers collection.

Add Method (PDFConductiveLayers Collection)

Collection: PDFConductiveLayers Collection

Prerequisites: None

Adds a PDFConductiveLayer to the PDFConductiveLayers collection.

Usage

PDFConductiveLayers.Add (ByVal nLayer As Long) As IPDFConductiveLayer

Arguments

- nLayer

  A long that specifies the PDFConductiveLayer Object to add to the collection.

Examples

```
LayersColl.Add 1
```

Remove Method (PDFConductiveLayers Collection)

Collection: PDFConductiveLayers Collection

Prerequisites: None

Removes a PDFConductiveLayer object from the PDFConductiveLayers collection.

Usage

PDFConductiveLayers.Remove (ByVal vIndex As Variant)

Arguments

- vIndex

  The index of the (PDFConductiveLayer Object) to remove from the collection.

Examples

```
LayersColl.Remove(i)
```

Count Property (PDFConductiveLayers Collection)

Collection: PDFConductiveLayers Collection

Access: Read-Only

Prerequisites: None

Returns the number of PDFConductiveLayer objects contained in the PDFConductiveLayers collection.

Usage

PDFConductiveLayers.Count

Arguments

None

Return Values

Long. The number of (PDFConductiveLayer Objects) the collection contains.

Examples

```
ConductiveLayerCnt = LayersColl.Count
```

Item Property (PDFConductiveLayers Collection)

Collection: PDFConductiveLayers Collection

Access: Read-Only

Prerequisites: None

Returns a PDFConductiveLayer object contained in the PDFConductiveLayers collection.

Usage

PDFConductiveLayers.Item (ByVal vIndex As Variant)

Arguments

- vIndex

  Index of the object to retrieve from the collection.

Return Values

PDFConductiveLayer. The object (PDFConductiveLayer Object) located at position vIndex.

Examples

```
Set layerObj = LayersColl.Item(i)
```

PDFOutputSheets Collection

Defines the methods and properties of a PDFOutputSheets collection.

This is a collection of PDFOutputSheet Objects.

See also Sheets Property (PDFOutput Object)

Table 39. PDFOutputSheetsCollection Methods and Properties

Method or Property	Description
Add Method (PDFOutputSheets Collection)	Adds a PDFOutputSheet object to the PDFOutputSheets collection.
Remove Method (PDFOutputSheets Collection)	Removes a PDFOutputSheet object from the PDFOutputSheets collection.
Count Property (PDFOutputSheets Collection)	Returns the number of PDFOutputSheet objects contained in the PDFOutputSheets collection.
Item Property (PDFOutputSheets Collection)	Returns a PDFOutputSheet object contained in the PDFOutputSheets collection.

Add Method (PDFOutputSheets Collection)

Collection: PDFOutputSheets Collection

Prerequisites: None

Adds a PDFOutputSheet object to the PDFOutputSheets collection.

Usage

PDFOutputSheets.Add (ByVal sSheetName As String) As PDFOutputSheet

Arguments

- sSheetName

  A string that contains the name of the PDFOutputSheet Object to add to the collection.

Examples

```
oPDFOutputSheets.Add "TopSilkScreen"
oPDFOutputSheets.Add "BottomSilkScreen"
oPDFOutputSheets.Add "TopSolderMask"
```

Remove Method (PDFOutputSheets Collection)

Collection: PDFOutputSheets Collection

Prerequisites: None

Removes a PDFOutputSheet object from the PDFOutputSheets collection.

Usage

PDFOutputSheets.Remove (ByVal vIndex As Variant)

Arguments

- vIndex

  The index of the PDFOutputSheet Object to remove from the collection.

Examples

```
sheetsColl.Remove(i)
```

Count Property (PDFOutputSheets Collection)

Collection: PDFOutputSheets Collection

Access: Read-Only

Prerequisites: None

Returns the number of PDFOutputSheet objects contained in the PDFOutputSheets collection.

Usage

PDFOutputSheets.Count

Arguments

None

Return Values

Long. The number of PDFOutputSheet Objects the collection contains.

Examples

```
totals = oPDFOutputSheets.Count
```

Item Property (PDFOutputSheets Collection)

Collection: PDFOutputSheets Collection

Access: Read-Only

Prerequisites: None

Returns a PDFOutputSheet object contained in the PDFOutputSheets collection.

Usage

PDFOutputSheets.Item (ByVal vIndex As Variant)

Arguments

- vIndex

  Index of the object to retrieve from the collection.

Return Values

PDFOutputSheet. The PDFOutputSheet Object located at position vIndex.

Examples

```
Set sheetObj = SheelsColl.Item(i)
```

UserLayers Collection

Defines the methods and properties of a UserLayers collection. This is a collection of layers.

See also:

- UserLayers Property (GerberOutputFile Object)
- UserLayers Property (PDFOutputSheet Object)
- BottomSideUserLayers Property (SilkscreenGen Object)
- TopSideUserLayers Property (SilkscreenGen Object)

Table 40. UserLayers Collection Methods

Method	Description
Add Method (UserLayers Collection)	Adds a user layer to the UserLayers collection.

Add Method (UserLayers Collection)

Collection: UserLayers Collection

Prerequisites: None

Adds a user layer to the UserLayers collection.

Usage

UserLayers.Add (ByVal strUserLayer As String)

Arguments

- strUserLayer

  A string that contains the user layer to add to the list of layers.

Examples

```
UserLayers.Add “User_64”
```

Chapter 4
Batch Engines Enumerated Types

Automation Batch Engines support a wide range of enumerated types.

Batch Engines Enumerated Type Descriptions

Batch Engines Enumerated Type Descriptions

This section lists and describes the available Batch Engines enumerated types.

Refer to a specific enumerated type for information related to that type.

Table 41. MGCPCB Engines Enumerated Types

Enumerated Type	Description
EEnginesAngleUnit Enum	Use this enumerator to specify the angle unit, degrees or radians.
EEnginesArcStyle Enum	Use this enumerator to specify the arc style, using the ArcStyle property.
EEnginesBoardItem Enum	Use this enumerator to specify the board and panel element type, using the Add method of the BoardItems Collection.
EEnginesCellItem Enum	Use this enumerator to specify the cell item type, using the Add method of the CellItems collection.
EEnginesCellOutlineType Enum	Use this enumerator to specify the cell outline type, using the CellOutline property of the GeneralInterfaceEngine object.
EEnginesCellSide Enum	Use this enumerator to specify the side of the board on which to generate silkscreen information, using the SilkScreenProcessSide property of the SilkscreenGen object.
EEnginesCellType Enum	Use this enumerator to specify the cell type, using the Add method of the CellTypes collection.
EEnginesCharacterSet Enum	Use this enumerator to specify the data character set, using the CharacterSet property of the GerberMachineFormat or NCDrillMachineFormat objects.
EEnginesColumnAlignment Enum	Use this enumerator to specify the NC drill column alignment: left, right, or center.
EEnginesConductiveItem Enum	Use this enumerator to specify the layer element type, using the Add method of the ConductiveItems collection.
EEnginesDataDelimiter Enum	Use this enumerator to specify the Gerber data delimiter, using the Delimiter property of the GerberMachineFormat object.
EEnginesDataType Enum	Use this enumerator to specify the Gerber data type, using the DataType property of the GerberMachineFormat object.
EEnginesDrawingTask Enum	Use this enumerator to specify the Drawing File Wizard operation, using the DrawingTask property of the DrawingFileWizard object.
EEnginesDRCElementType Enum	Use this enumerator to specify the DRC element type. See DRC Object for more information.
EEnginesDRCFeaturesLayer Enum	Use this enumerator to specify the DRC features layer, using the DRCFeaturesLayer property of the ODBPPOutputEngine object.
EEnginesDRCMissingConductivePad Enum	Use this enumerator to specify the DRC rules for missing conductive pads, using the MissingConductivePads property of the DRC object.

Table 41. MGCPCB Engines Enumerated Types (continued)

Enumerated Type	Description
EEnginesDRCMissingCoverLayerPad Enum	Use this enumerator to specify the DRC rules for missing cover layer pads, using the MissingCoverLayerPads property of the DRC object.
EEnginesDRCMissingSolderPad Enum	Use this enumerator to specify the DRC rules for missing pads on soldermask and solderpaste, using the MissingSolderPads property of the DRC object.
EEnginesDRCNetRule Enum	Use this enumerator to specify the net rule options for DRC Connectivity and Special Rules, using the NetRules property of the DRC object.
EEnginesDRCPadRule Enum	Use this enumerator to specify the holes/pads/vias options for DRC Connectivity and Special Rules, using the PadRules property of the DRC object.
EEnginesDRCPadToPadCheck Enum	Use this enumerator to specify the pad to pad check options, using the PadToPad property of the DRC object.
EEnginesDRCPlaneRule Enum	Use this enumerator to specify the plane options for DRC Connectivity and Special Rules, using the PlaneRules property of the DRC object.
EEnginesDRCRules Enum	Use this enumerator to specify the DRC rules to check, using the Rules property of the DRC object.
EEnginesDRCTraceRule Enum	Use this enumerator to specify the trace options for DRC Connectivity and Special Rules, using the TraceRules property of the DRC object.
EEnginesDRCViasUnder Enum	Use this enumerator to specify the vias under options for DRC Connectivity and Special Rules, using the ViasUnder property of the DRC object.
EEnginesDrillSymbol Enum	Use this enumerator to specify the drill symbol, using the DrillSymbol property of the DrillHole object.
EEnginesDrillSymbolType Enum	Drill symbol type constants.
EEnginesElementType Enum	Use this enumerator to specify the DXF Export element type, using the PutElementExport or PutWBElementExport methods of the DXFExport object.
EEnginesErrCode Enum	Error Code constants.
EEnginesErrorSeverity Enum	Use this enumerator to identify the severity of an error, using the Severity property of the Error object.
EEnginesExtPrintFileFormat Enum	Use this enumerator to specify the Extended Print output type, using the FileFormat property of the ExtendedPrint object.
EEnginesGeneralInterfaceIdentifyPartsType Enum	Use this enumerator to specify the identify parts option, using the IdentifyPartBy property of the GeneralInterfaceEngine object.
EEnginesGeneralInterfaceOutputType Enum	Use this enumerator to specify the output type, using the OutputType property of the GeneralInterfaceEngine object.
EEnginesGeneralInterfaceUnitsType Enum	Use this enumerator to specify the units type, using the UnitsType property of the GeneralInterfaceEngine object.
EEnginesIDFExportComponentSideType Enum	Use this enumerator to specify the mountside of the components to export, using the ComponentSide property of the IDFExport object (for panel only).
EEnginesIDFExportPackageOutlineType Enum	Use this enumerator to specify the package outline layer, using the PackageOutlineLayer property of the IDFExport object.

Table 41. MGCPCB Engines Enumerated Types (continued)

Enumerated Type	Description
EEnginesIDFExportSectionsType Enum	Use this enumerator to specify the section type of the items to export, using the ExportSection property of the IDFExport object.
EEnginesIDFVersionType Enum	Use this enumerator to specify the IDF version, using the Version property of the IDFExport object.
EEnginesMappedUserLayer Enum	Use this enumerator to specify user layer mapping. Use 1-1000 for Layer1 - Layer 1000.
EEnginesNCDrillChartUnits Enum	Use this enumerator to specify the drill chart units, using the UnitEx property of the NCDrillChart object.
EEnginesNCDrillColumn Enum	Use this enumerator to identify the drill chart column type, using the ColumnType property of the NCDrillChartColumn object.
EEnginesNCDrillDataType Enum	Use this enumerator to specify the NC drill data type, using the DataType property of the NCDrillMachineFormat object.
EEnginesNCDrillSweepAxis Enum	Use this enumerator to specify the drill sweep axis, using the SweepAxis property of the NCDrillParameters object.
EEnginesNCDrillUnits Enum	Use this enumerator to specify the NC drill chart units, using the UnitEx property of the NCDrillChart object.
EEnginesNonFuncPad Enum	Use this enumerator to specify the options for non-functional pad removal for pins and vias.
EEnginesODBPPExportMode Enum	Use this enumerator to specify the type of data to export, using the ODBPPExportMode property of the ODBPPOutputEngine object.
EEnginesOutputUnits Enum	Use this enumerator to specify the output units for ODB++, using the Units property of the ODBPPOutputEngine object.
EEnginesPackageOutlineLayer Enum	Use this enumerator to specify the package outline layer, using the PackageOutlineLayer property of the ODBPPOutputEngine object.
EEnginesPDFSheetPageSize Enum	This enumerator specifies the page size for the PDF output sheet.
EEnginesPlotColor Enum	This enumerator specifies PDF plot color settings.
EEnginesPolygonFill Enum	Use this enumerator to specify the Gerber polygon shape fill, using the PolygonFillMethod property of the GerberMachineFormat object.
EEnginesSide Enum	Use this enumerator to specify the placement side, using the PutWBElementExport method of the DXFExport object.
EEnginesSilkScreenBreakType Enum	Use this enumerator to specify the silkscreen break type, using the BreakSilkscreen property of the SilkscreenGen object.
EEnginesSilkscreenDesignLayer Enum	Use this enumerator to specify the silkscreen layers for selected package groups.
EEnginesStepMode Enum	Use this enumerator to specify Gerber and NC Drill step mode options.
EEnginesUnit Enum	MGCPCBEngines Unit constants.
EEnginesZeroTruncation Enum	Use this enumerator to specify Gerber and NC Drill zero truncation options.

EEnginesAngleUnit Enum

Prerequisites: None

Use this enumerator to specify the angle unit, degrees or radians.

Usage

EEnginesAngleUnit.*Constant*

Arguments

- **eengAngleUnitDegrees**

  The numerical value for this constant is 0.
- **eengAngleUnitRadians**

  The numerical value for this constant is 1.

Examples

```
‘ Enum EEnginesAngleUnit

eengAngleUnitDegrees = 0
eengAngleUnitRadians = 1
```

EEnginesArcStyle Enum

Prerequisites: None

Use this enumerator to specify the arc style, using the ArcStyle property.

Usage

EEnginesArcStyle.*Constant*

Arguments

- **eengArcStyleFull**

  Creates full circle hardware arcs (360 degree sweep axis).

  The numerical value for this constant is 1.

- **eengArcStyleNone**

  Disallows the use of hardware arcs. Arcs and circles are approximated with line strings. Used in NC Drill format file.

  The numerical value for this constant is 2.

- **eengArcStyleQuadrant**

  Converts all circles into four arcs (90 degree sweep axis also used for NC Drill machine Format file.

  The numerical value for this constant is 0.

- **eengArcStyleRadius**

  Used for NC Drill machine format file. Not used in Gerber Machine Format

  The numerical value for this constant is 4.

- **eengArcStyleStroke**

  The numerical value for this constant is 3.

Examples

```
‘ Enum EEnginesArcStyle

eengArcStyleFull = 1
eengArcStyleNone = 2
eengArcStyleQuadrant = 0
eengArcStyleRadius = 4
eengArcStyleStroke = 3
```

EEnginesBoardItem Enum

Prerequisites: None

Use this enumerator to specify the board and panel element type, using the Add method of the BoardItems Collection.

Usage

EEnginesBoardItem.*Constant*

Arguments

- **eengBoardItemAll**

  All available options for the type of output

  The numerical value for this constant is 0.
- **eengBoardItemBendAreas**

  The numerical value for this constant is 34 (&H22)
- **eengBoardItemBoardCavities**

  The numerical value for this constant is 32 (&H20).
- **eengBoardItemFlexLayers**

  The numerical value for this constant is 35 (&H23)
- eengBoardItemFlexPadLayers

  The numerical value for this constant is 36 (&H24)
- **eengBoardItemBoardObstruct**

  Used in PDF Output for panel design I only.

  The numerical value for this constant is 1.
- **eengBoardItemBoardOrigin**

  The numerical value for this constant is 2.
- **eengBoardItemBoardOutline**

  The graphical representation of the board shape. In Gerber it is processed as a drawn element.

  The numerical value for this constant is 3.
- **eengBoardItemContours**

  In Gerber all mounting holes, both placed on the board and within cells are processed and drawn using the width of the contour.

  The numerical value for this constant is 4.
- **eengBoardItemDetailViews**

  Used in PDFOutput only

  The numerical value for this constant is 5.

- **eengBoardItemDocumentation**

  Only available with a panel design.

  The numerical value for this constant is 6.

- **eengBoardItemDrcWindow**

  Used in PDFOutput only

  The numerical value for this constant is 7.

- **eengBoardItemDrillDrawingThru**

  In Gerber all drill symbol and drawing information is processed and all drill symbols are drawn not flashed

  The numerical value for this constant is 8.

- **eengBoardItemGeneratedSilkscreenBottom**

  In Gerber, all silkscreen bottom elements which were created by the Silkscreen Generator are processed

  The numerical value for this constant is 9.

- **eengBoardItemGeneratedSilkscreenTop**

  In Gerber, all silkscreen top elements which were created by the Silkscreen Generator are processed

  The numerical value for this constant is 10 (&HA).

- **eengBoardItemManufacturingOutline**

  Used in PDFOutput only

  The numerical value for this constant is 11 (&HB).

- **eengBoardItemMountingHoles**

  The numerical value for this constant is 12 (&HC).

- **eengBoardItemNCDrillOrigin**

  Used in PDFOutput only

  The numerical value for this constant is 13 (&HD).

- **eengBoardItemPanelBorder**

  Used in PDF Output and only for Panel design

  The numerical value for this constant is 14 (&HE).

- **eengBoardItemPanelHole**

  Used in PDF Output and only for Panel design

  The numerical value for this constant is 15 (&HF).

- **eengBoardItemPanelOrigin**

  Used in PDF Output and only for Panel design

  The numerical value for this constant is 16 (&H10).

- **eengBoardItemPanelOutline**

  Only available with a Panel design. In Gerber it is processed as a drawn element.

The numerical value for this constant is 17 (&H11).

- **eengBoardItemPartHoles**

  In Gerber all holes which are not mounting holes or vias are processed

  The numerical value for this constant is 18 (&H12).

- **eengBoardItemPlacementObstructBottom**

  Used in PDFOutput only

  The numerical value for this constant is 19 (&H13).

- **eengBoardItemPlacementObstructTop**

  Used in PDFOutput only

  The numerical value for this constant is 20 (&H14).

- **eengBoardItemRoomBottom**

  Used in PDFOutput only

  The numerical value for this constant is 21 (&H15).

- **eengBoardItemRoomTop**

  Used in PDFOutput only

  The numerical value for this constant is 22 (&H16).

- **eengBoardItemRouteBorder**

  Used in PDFOutput only

  The numerical value for this constant is 23 (&H17).

- **eengBoardItemScoringLine**

  Only available with a panel design.

  The numerical value for this constant is 24 (&H18).

- **eengBoardItemShearingHoles**

  Only available with a panel design.

  The numerical value for this constant is 25 (&H19).

- **eengBoardItemShearingLine**

  Only available with a panel design.

  The numerical value for this constant is 26 (&H1A).

- **eengBoardItemSoldermaskBottom**

  In Gerber all soldermask bottom pads and shapes built into cells including soldermask shapes placed on the board on the soldermask bottom layer are processed.

  The numerical value for this constant is 27 (&H1B).

- **eengBoardItemSoldermaskTop**

  In Gerber all soldermask top pads and shapes built into cells including soldermask shapes placed on the board on the soldermask top layer are processed.

  The numerical value for this constant is 28 (&H1C).

- **eengBoardItemSolderpasteBottom**

  In Gerber all solderpaste bottom pads built into cells are processed.

  The numerical value for this constant is 29(&H1D).
- **eengBoardItemSolderpasteTop**

  In Gerber all solderpaste top pads built into cells are processed.

  The numerical value for this constant is 30 (&H1E).
- **eengBoardItemTestFixtureOutline**

  The numerical value for this constant is 33 (&H21)
- **eengBoardItemToolingHoles**

  Only available with a panel design.

  The numerical value for this constant is 31 (&H1F).

Examples

```
‘ Enum EEnginesBoardItem

eengBoardItemAll = 0
eengBoardItemBoardObstruct = 1
eengBoardItemBoardOrigin = 2
eengBoardItemBoardOutline = 3
eengBoardItemContours = 4
eengBoardItemDetailViews = 5
eengBoardItemDocumentation = 6
eengBoardItemDrcWindow = 7
eengBoardItemDrillDrawingThru = 8
eengBoardItemGeneratedSilkscreenBottom = 9
eengBoardItemGeneratedSilkscreenTop = 10
eengBoardItemManufacturingOutline = 11
eengBoardItemMountingHoles = 12
eengBoardItemNCDrillOrigin = 13
eengBoardItemPanelBorder = 14
eengBoardItemPanelHole = 15
eengBoardItemPanelOrigin = 16
eengBoardItemPanelOutline = 17
eengBoardItemPartHoles = 18
eengBoardItemPlacementObstructBottom = 19
eengBoardItemPlacementObstructTop = 20
eengBoardItemRoomBottom = 21
eengBoardItemRoomTop = 22
eengBoardItemRouteBorder = 23
eengBoardItemScoringLine = 24
eengBoardItemShearingHoles = 25
eengBoardItemShearingLine = 26
eengBoardItemSoldermaskBottom = 27
eengBoardItemSoldermaskTop = 28
```

```
eengBoardItemSolderpasteBottom = 29
eengBoardItemSolderpasteTop = 30
eengBoardItemToolingHoles = 31
```

EEnginesCellItem Enum

Prerequisites: None

Use this enumerator to specify the cell item type, using the Add method of the CellItems collection.

Usage

EEnginesCellItem.*Constant*

Arguments

- **eengCellItemAll**

  Select all cell items applicable for the current engine

  The numerical value for this constant is -1 (&HFFFFFFFF).
- **eengCellItemAssemblyOutline**

  Assembly outline processed for the selected cell type.

  The numerical value for this constant is 1.
- **eengCellItemAssemblyPartNumber**

  Assembly part number processed for the selected cell type.

  The numerical value for this constant is 2.
- **eengCellItemAssemblyReferenceDesignator**

  Assembly reference designator processed for the selected cell type.

  The numerical value for this constant is 4.
- **eengCellItemBondWires**

  The numerical value for this constant is 524288 (&H80000).
- **eengCellItemCapacitorDielectric**

  Only available if embedded passive elements are resident in the design.

  The numerical value for this constant is 2048 (&H800).
- **eengCellItemCapacitorPlate1**

  The numerical value for this constant is 4096 (&H1000).
- **eengCellItemCapacitorPlate2**

  The numerical value for this constant is 8192 (&H2000).
- **eengCellItemCellOrigin**

  Used in PDF Output only.

  The numerical value for this constant is 8.
- **eengCellItemCapacitorPlate2**

  The numerical value for this constant is 8192 (&H2000).
- **eengCellItemDiePins**

The numerical value for this constant is 1048576 (&H100000).

- **eengCellItemInsertionOutline**

  The numerical value for this constant is 32 (&H20).

- **eengCellItemMezzanineCapacitorPlate1**

  Only available if embedded passive elements are resident in the design.

  The numerical value for this constant is 16384 (&H4000).

- **eengCellItemMezzanineCapacitorPlate2**

  Only available if embedded passive elements are resident in the design.

  The numerical value for this constant is 32768 (&H8000).

- **eengCellItemMezzanineViaPad**

  Only available if embedded passive elements are resident in the design.

  The numerical value for this constant is 65536 (&H10000).

- **eengCellItemPlacementOutline**

  The numerical value for this constant is 64 (&H40).

- **eengCellItemResistorMask**

  Only available if embedded passive elements are resident in the design.

  The numerical value for this constant is 131072 (&H20000).

- **eengCellItemResistorOverglaze**

  Only available if embedded passive elements are resident in the design.

  The numerical value for this constant is 262144 (&H40000).

- **eengCellItemSilkScreenOutline**

  The numerical value for this constant is 128 (&H80).

- **eengCellItemSilkScreenPartNumber**

  The numerical value for this constant is 256 (&H100).

- **eengCellItemSilkScreenReferenceDesignator**

  The numerical value for this constant is 512 (&H200).

- **eengCellItemTestPointObstruct**

  Used in PDF Output only.

  The numerical value for this constant is 1024 (&H400).

- **eengCellItemTPAssemblyReferenceDesignator**

  The numerical value for this constant is 33554432 (&H2000000).

- **eengCellItemTPSilkscreenReferenceDesignator**

  The numerical value for this constant is 67108864 (&H4000000).

Examples

```
‘ Enum EEnginesCellItem

eengCellItemAll = -1
eengCellItemAssemblyOutline = 1
eengCellItemAssemblyPartNumber = 2
eengCellItemAssemblyReferenceDesignator = 4
eengCellItemCapacitorDielectric = 2048
eengCellItemCapacitorPlate1 = 4096
eengCellItemCapacitorPlate2 = 8192
eengCellItemCellOrigin = 8
eengCellItemGlueSpots = 16
eengCellItemInsertionOutline = 32
eengCellItemMezzanineCapacitorPlate1 = 16384
eengCellItemMezzanineCapacitorPlate2 = 32768
eengCellItemMezzanineViaPad = 65536
eengCellItemPlacementOutline = 64
eengCellItemResistorMask = 131072
eengCellItemResistorOverglaze = 262144
eengCellItemSilkScreenOutline = 128
eengCellItemSilkScreenPartNumber = 256
eengCellItemSilkScreenReferenceDesignator = 512
eengCellItemTestPointObstruct = 1024
eengCellItemTPAssemblyReferenceDesignator = 33554432
eengCellItemTPSilkscreenReferenceDesignator = 67108864
```

EEnginesCellOutlineType Enum

Prerequisites: None

Use this enumerator to specify the cell outline type, using the CellOutline property of the GeneralInterfaceEngine object.

Usage

EEnginesCellOutlineType.*Constant*

Arguments

- **eengCellAssemblyOutline**

  The numerical value for this constant is 1.
- **eengCellPlacementOutline**

  The numerical value for this constant is 2.
- **eengCellSilkscreenOutline**

  The numerical value for this constant is 3.

Examples

```
‘ Enum EEnginesCellOutlineType

eengCellAssemblyOutline = 1
eengCellPlacementOutline = 2
eengCellSilkscreenOutline = 3
```

EEnginesCellSide Enum

Prerequisites: None

Use this enumerator to specify the side of the board on which to generate silkscreen information, using the SilkScreenProcessSide property of the SilkscreenGen object.

Usage

EEnginesCellSide. *Constant*

Arguments

- **eengCellSideBoth**

  The numerical value for this constant is 3.
- **eengCellSideBottom**

  The numerical value for this constant is 2.
- **eengCellSideNone**

  The numerical value for this constant is 0.
- **eengCellSideTop**

  The numerical value for this constant is 1.

Examples

```
‘Enum EEnginesCellSide

eengCellSideBoth = 3
eengCellSideBottom = 2
eengCellSideNone = 0
eengCellSideTop = 1
```

EEnginesCellType Enum

Prerequisites: None

Use this enumerator to specify the cell type, using the Add method of the CellTypes collection.

Usage

EEnginesCellType.*Constant*

Arguments

- **eengCellTypeAll**

  The numerical value for this constant is 0.
- **eengCellTypeBadBoardIdentifier**

  Available for a panel design only.

  The numerical value for this constant is 1.
- **eengCellTypeBreakAwayTab**

  Available for a panel design only.

  The numerical value for this constant is 2.
- **eengCellTypeBuried**

  The numerical value for this constant is 3.
- **eengCellTypeConnector**

  The numerical value for this constant is 4.
- **eengCellTypeDiscreteAxial**

  The numerical value for this constant is 5.
- **eengCellTypeDiscreteChip**

  The numerical value for this constant is 6.
- **eengCellTypeDiscreteOther**

  The numerical value for this constant is 7.
- **eengCellTypeDiscreteRadial**

  The numerical value for this constant is 8.
- **eengCellTypeEdgeConnector**

  The numerical value for this constant is 9.
- **eengCellTypeElectricalTestIdentifier**

  The numerical value for this constant is 10 (&HA).
- **eengCellTypeEmbeddedPassive**

  Only available if embedded passive elements are resident in the design.

  The numerical value for this constant is 34 (&H22).

- **eengCellTypeGeneral**

  The numerical value for this constant is 11 (&HB).
- **eengCellTypeGeneralPackageCell**

  Applicable for Gerber output for design panel only.

  The numerical value for this constant is 35 (&H23).
- **eengCellTypeGeneralPanelCell**

  Applicable for Gerber output for design panel only.

  The numerical value for this constant is 12 (&HC).
- **eengCellTypeGraphic**

  The numerical value for this constant is 13 (&HD).
- **eengCellTypeICBareDie**

  The numerical value for this constant is 14 (&HE).
- **eengCellTypeICBGA**

  The numerical value for this constant is 15 (&HF).
- **eengCellTypeICDIP**

  The numerical value for this constant is 16 (&H10).
- **eengCellTypeICFlipChip**

  The numerical value for this constant is 17 (&H11).
- **eengCellTypeICLCC**

  The numerical value for this constant is 18 (&H12).
- **eengCellTypeICOther**

  The numerical value for this constant is 19 (&H13).
- **eengCellTypeICPGA**

  The numerical value for this constant is 20 (&H14).
- **eengCellTypeICPLCC**

  The numerical value for this constant is 21 (&H15).
- **eengCellTypeICSIP**

  The numerical value for this constant is 22 (&H16).
- **eengCellTypeICSOIC**

  The numerical value for this constant is 23 (&H17).
- **eengCellTypeJumper**

  The numerical value for this constant is 24 (&H18).
- **eengCellTypeMechanical**

  The numerical value for this constant is 25 (&H19).
- **eengCellTypePanelIdentifier**

  Available for a panel design only

The numerical value for this constant is 26 (&H1A).

- **eengCellTypePanelStiffener**

  Only used for PDF Output and a panel design only.

  The numerical value for this constant is 27 (&H1B).

- **eengCellTypeRegistrationPin**

  Available for a panel design only

  The numerical value for this constant is 28 (&H1C).

- **eengCellTypeRegistrationPinGrid**

  Only used for Gerber in panel design output.

  The numerical value for this constant is 29 (&H1D).

- **eengCellTypeSkyhook**

  Available for a panel design only

  The numerical value for this constant is 30 (&H1E).

- **eengCellTypeSolderPalette**

  Only used for PDF Output and a panel design only.

  The numerical value for this constant is 31 (&H1F).

- **eengCellTypeTestCoupon**

  Available for a panel design only

  The numerical value for this constant is 32 (&H20).

- **eengCellTypeTestPoint**

  The numerical value for this constant is 33 (&H21).

Examples

```
‘ Enum EEnginesCellType

eengCellTypeAll = 0
eengCellTypeBadBoardIdentifier = 1
eengCellTypeBreakAwayTab = 2
eengCellTypeBuried = 3
eengCellTypeConnector = 4
eengCellTypeDiscreteAxial = 5
eengCellTypeDiscreteChip = 6
eengCellTypeDiscreteOther = 7
eengCellTypeDiscreteRadial = 8
eengCellTypeEdgeConnector = 9
eengCellTypeElectricalTestIdentifier = 10
eengCellTypeEmbeddedPassive = 34
eengCellTypeGeneral = 11
eengCellTypeGeneralPackageCell = 35
eengCellTypeGeneralPanelCell = 12
eengCellTypeGraphic = 13
eengCellTypeICBareDie = 14
```

```
eengCellTypeICBGA = 15
eengCellTypeICDIP = 16
eengCellTypeICFlipChip = 17
eengCellTypeICLCC = 18
eengCellTypeICOther = 19
eengCellTypeICPGA = 20
eengCellTypeICPLCC = 21
eengCellTypeICSIP = 22
eengCellTypeICSOIC = 23
eengCellTypeJumper = 24
eengCellTypeMechanical = 25
eengCellTypePanelIdentifier = 26
eengCellTypePanelStiffener = 27
eengCellTypeRegistrationPin = 28
eengCellTypeRegistrationPinGrid = 29
eengCellTypeSkyhook = 30
eengCellTypeSolderPalette = 31
eengCellTypeTestCoupon = 32
eengCellTypeTestPoint = 33
```

EEnginesCharacterSet Enum

Prerequisites: None

Use this enumerator to specify the data character set, using the CharacterSet property of the GerberMachineFormat or NCDrillMachineFormat objects.

Usage

EEnginesCharacterSet.*Constant*

Arguments

- **eengCharacterSetASCII**

  The numerical value for this constant is 1.

- **eengCharacterSetEBCDIC**

  The numerical value for this constant is 0.

Examples

```
‘ Enum EEnginesCharacterSet

eengCharacterSetASCII = 1
eengCharacterSetEBCDIC = 0
```

EEnginesColumnAlignment Enum

Prerequisites: None

Use this enumerator to specify the NC drill column alignment: left, right, or center.

Usage

EEnginesColumnAlignment.*Constant*

Arguments

- **eengColumnAlignCenter**

  The numerical value for this constant is 2.
- **eengColumnAlignLeft**

  The numerical value for this constant is 1.
- **eengColumnAlignRight**

  The numerical value for this constant is 3.

EEnginesConductiveItem Enum

Prerequisites: None

Use this enumerator to specify the layer element type, using the Add method of the ConductiveItems collection.

Usage

EEnginesConductiveItem.*Constant*

Arguments

- **eengCondItemActualPlaneShapes**

  The numerical value for this constant is 1.
- **eengCondItemAll**

  The numerical value for this constant is -1 (&HFFFFFFFF).
- **eengCondItemBondPads**

  The numerical value for this constant is 1048576 (&H100000).
- **eengCondItemCopperBalancing**

  The numerical value for this constant is 2.
- **eengCondItemEmbeddedCapacitorPads**

  Only available if embedded passive elements are resident in the design.

  The numerical value for this constant is 262144 (&H40000).
- **eengCondItemEmbeddedResistorPads**

  Only available if embedded passive elements are resident in the design.

  The numerical value for this constant is 524288 (&H80000).
- **eengCondItemEtchedText**

  The numerical value for this constant is 4.
- **eengCondItemFiducials**

  The numerical value for this constant is 8.
- **eengCondItemMountingHoles**

  The numerical value for this constant is 2097152 (&H200000).
- **eengCondItemPartHoles**

  The numerical value for this constant is 4194304 (&H400000).
- **eengCondItemPartPadsSMD**

  The numerical value for this constant is 16 (&H10).
- **eengCondItemPartPadsTest**

  The numerical value for this constant is 32 (&H20).
- **eengCondItemPartPadsThru**

The numerical value for this constant is 64 (&H40).

- **eengCondItemPinNumbers**

  The numerical value for this constant is 128 (&H80).

- **eengCondItemPlaneData**

  The numerical value for this constant is 256 (&H100).

- **eengCondItemPlaneNoConnect**

  Use in PDF Output only

  The numerical value for this constant is 512 (&H200).

- **eengCondItemPlaneObstruct**

  Use in PDF Output only

  The numerical value for this constant is 1024 (&H400).

- **eengCondItemPlaneShape**

  Use in PDF Output only

  The numerical value for this constant is 2048 (&H800).

- **eengCondItemResistorAreas**

  The numerical value for this constant is 4096 (&H1000).

- **eengCondItemRouteObstruct**

  Use in PDF Output only

  The numerical value for this constant is 8192 (&H2000).

- **eengCondItemRuleArea**

  Use in PDF Output only

  The numerical value for this constant is 16384 (&H4000).

- **eengCondItemTraces**

  The numerical value for this constant is 32768 (&H8000).

- **eengCondItemViaHoles**

  The numerical value for this constant is 65536 (&H10000).

- **eengCondItemViaPads**

  The numerical value for this constant is 131072 (&H20000).

Examples

```
‘ Enum EEnginesConductiveItem

eengCondItemActualPlaneShapes = 1
eengCondItemAll = -1
eengCondItemBondPads = 1048576
eengCondItemCopperBalancing = 2
eengCondItemEmbeddedCapacitorPads = 262144
eengCondItemEmbeddedResistorPads = 524288
eengCondItemEtchedText = 4
```

```
eengCondItemFiducials = 8
eengCondItemMountingHoles = 2097152
eengCondItemPartHoles = 4194304
eengCondItemPartPadsSMD = 16
eengCondItemPartPadsTest = 32
eengCondItemPartPadsThru = 64
eengCondItemPinNumbers = 128
eengCondItemPlaneData = 256
eengCondItemPlaneNoConnect = 512
eengCondItemPlaneObstruct = 1024
eengCondItemPlaneShape = 2048
eengCondItemResistorAreas = 4096
eengCondItemRouteObstruct = 8192
eengCondItemRuleArea = 16384
eengCondItemTraces = 32768
eengCondItemViaHoles = 65536
eengCondItemViaPads = 131072
```

EEnginesDataDelimiter Enum

Prerequisites: None

Use this enumerator to specify the Gerber data delimiter, using the Delimiter property of the GerberMachineFormat object.

Usage

EEnginesDataDelimiter.*Constant*

Arguments

- **eengDataDelimiterAt**

  The numerical value for this constant is 2.
- **eengDataDelimiterHash**

  The numerical value for this constant is 1.
- **eengDataDelimiterStar**

  The numerical value for this constant is 0.

Examples

```
‘ Enum EEnginesDataDelimiter

eengDataDelimiterAt = 2
eengDataDelimiterHash = 1
eengDataDelimiterStar = 0
```

EEnginesDataType Enum

Prerequisites: None

Use this enumerator to specify the Gerber data type, using the DataType property of the GerberMachineFormat object.

Usage

EEnginesDataType.*Constant*

Arguments

- **eengGerber274D**

  The numerical value for this constant is 0.

- **eengGerber274X**

  The numerical value for this constant is 1.

Examples

```
‘ Enum EEnginesDataType

eengGerber274D = 0
eengGerber274X = 1
```

EEnginesDrawingTask Enum

Prerequisites: None

Use this enumerator to specify the Drawing File Wizard operation, using the DrawingTask property of the DrawingFileWizard object.

Usage

EEnginesDrawingTask.*Constant*

Arguments

- **eengDrawingTaskCleanup**

  The numerical value for this constant is 4.
- **eengDrawingTaskCopy**

  The numerical value for this constant is 1.
- **eengDrawingTaskCreate**

  The numerical value for this constant is 0.
- **eengDrawingTaskDelete**

  The numerical value for this constant is 3.
- **eengDrawingTaskMove**

  The numerical value for this constant is 2.

Examples

```
‘ Enum EEnginesDrawingTask

eengDrawingTaskCleanup = 4
eengDrawingTaskCopy = 1
eengDrawingTaskCreate = 0
eengDrawingTaskDelete = 3
eengDrawingTaskMove = 2
```

EEnginesDRCElementType Enum

Prerequisites: None

Use this enumerator to specify the DRC element type. See DRC Object for more information.

Usage

EEnginesDRCElementType.*Constant*

Arguments

- **eengDRCElementAssemblyOutlines**

  The numerical value for this constant is 0.
- **eengDRCElementAssemblyPartNumbers**

  The numerical value for this constant is 1.
- **eengDRCElementAssemblyRefDes**

  The numerical value for this constant is 2.
- **eengDRCElementBadBoardIdentifier**

  The numerical value for this constant is 3.
- **eengDRCElementBendAreas**

  The numerical value for this constant is 58 (&H3A).
- **eengDRCElementBoardOutline**

  The numerical value for this constant is 4.
- **eengDRCElementBondPads**

  The numerical value for this constant is 57 (&H39).
- **eengDRCElementBreakAwayTab**

  Available for a panel design only.

  The numerical value for this constant is 5.
- **eengDRCElementCavities**

  The numerical value for this constant is 56 (&H38).
- **eengDRCElementConductiveText**

  The numerical value for this constant is 6.
- **eengDRCElementContours**

  The numerical value for this constant is 7.
- **eengDRCElementCopperBalancing**

  The numerical value for this constant is 8.
- **eengDRCElementEdgeConnectorPads**

  The numerical value for this constant is 9.

- **eengDRCElementElectricalTestIdentifier**

  The numerical value for this constant is 10 (&HA).
- **eengDRCElementEPMasks**

  The numerical value for this constant is 51 (&H33).
- **eengDRCElementFiducialPads**

  The numerical value for this constant is 11 (&HB).
- **eengDRCElementGeneral**

  The numerical value for this constant is 12 (&HC).
- **eengDRCElementGeneratedSilkscreen**

  The numerical value for this constant is 13 (&HD).
- **eengDRCElementInsertionOutlines**

  The numerical value for this constant is 14 (&HE).
- **eengDRCElementJumperWires**

  The numerical value for this constant is 15 (&HF).
- **eengDRCElementManufOutline**

  The numerical value for this constant is 16 (&H10).
- **eengDRCElementMountingHoles**

  The numerical value for this constant is 17 (&H11).
- **eengDRCElementNegativePlane**

  The numerical value for this constant is 18 (&H12).
- **eengDRCElementPadObstructs**

  The numerical value for this constant is 52 (&H34).
- **eengDRCElementPanelBorder**

  Available for a panel design only.

  The numerical value for this constant is 19 (&H13).
- **eengDRCElementPanelIdentifier**

  Available for a panel design only.

  The numerical value for this constant is 20 (&H14).
- **eengDRCElementPanelObstruct**

  Available for a panel design only.

  The numerical value for this constant is 21 (&H15).
- **eengDRCElementPanelOutline**

  Available for a panel design only.

  The numerical value for this constant is 22 (&H16).
- **eengDRCElementPanelStiffener**

  Available for a panel design only.

The numerical value for this constant is 23 (&H17).

- **eengDRCElementPartHoles**

  The numerical value for this constant is 24 (&H18).

- **eengDRCElementPartPads**

  The numerical value for this constant is 25 (&H19).

- **eengDRCElementPartPadsSMD**

  The numerical value for this constant is 11 (&H35).

- **eengDRCElementPartPadsThru**

  The numerical value for this constant is 54 (&H36).

- **eengDRCElementPlacementObstructs**

  The numerical value for this constant is 26 (&H1A).

- **eengDRCElementPlacementOutlines**

  The numerical value for this constant is 27 (&H1B).

- **eengDRCElementPositivePlane**

  The numerical value for this constant is 28 (&H1C).

- **eengDRCElementRegistrationPin**

  Available for a panel design only

  The numerical value for this constant is 29 (&H1D).

- **eengDRCElementResistorShapes**

  The numerical value for this constant is 30 (&H1E).

- **eengDRCElementRFShapes**

  The numerical value for this constant is 55 (&H37).

- **eengDRCElementRouteArea**

  The numerical value for this constant is 31 (&H1F).

- **eengDRCElementRouteBorder**

  The numerical value for this constant is 32 (&H20).

- **eengDRCElementShearingHole**

  The numerical value for this constant is 33 (&H21).

- **eengDRCElementSilkscreenOutlines**

  The numerical value for this constant is 34 (&H22).

- **eengDRCElementSilkscreenPartNumbers**

  The numerical value for this constant is 35 (&H23).

- **eengDRCElementTestPointCenters**

  Available for a board design only

  The numerical value for this constant is 43 (&H2B).

- **eengDRCElementTestPointObstructs**

Available for a board design only

The numerical value for this constant is 44 (&H2C).

- **eengDRCElementToolingHoles**

  Available for a panel design only

  The numerical value for this constant is 45 (&H2D).

- **eengDRCElementTraceObstructs**

  Available for a board design only

  The numerical value for this constant is 46 (&H2E).

- **eengDRCElementTraces**

  Available for a board design only

  The numerical value for this constant is 47 (&H2F).

- **eengDRCElementViaHoles**

  Available for a board design only

  The numerical value for this constant is 48 (&H30).

- **eengDRCElementViaObstructs**

  Available for a board design only

  The numerical value for this constant is 49 (&H31).

- **eengDRCElementViaPads**

  Available for a board design only

  The numerical value for this constant is 50 (&H32).

Examples

```
‘ Enum EEnginesDRCElementType

eengDRCElementAssemblyOutlines = 0
eengDRCElementAssemblyPartNumbers = 1
eengDRCElementAssemblyRefDes = 2
eengDRCElementBadBoardIdentifier = 3
eengDRCElementBoardOutline = 4
eengDRCElementBreakAwayTab = 5
eengDRCElementConductiveText = 6
eengDRCElementContours = 7
eengDRCElementCopperBalancing = 8
eengDRCElementEdgeConnectorPads = 9
eengDRCElementElectricalTestIdentifier = 10
eengDRCElementFiducialPads = 11
eengDRCElementGeneral = 12
eengDRCElementGeneratedSilkscreen = 13
eengDRCElementInsertionOutlines = 14
eengDRCElementJumperWires = 15
eengDRCElementManufOutline = 16
eengDRCElementMountingHoles = 17
```

```
eengDRCElementNegativePlane = 18
eengDRCElementPanelBorder = 19
eengDRCElementPanelIdentifier = 20
eengDRCElementPanelObstruct = 21
eengDRCElementPanelOutline = 22
eengDRCElementPanelStiffener = 23
eengDRCElementPartHoles = 24
eengDRCElementPartPads = 25
eengDRCElementPlacementObstructs = 26
eengDRCElementPlacementOutlines = 27
eengDRCElementPositivePlane = 28
eengDRCElementRegistrationPin = 29
eengDRCElementResistorShapes = 30
eengDRCElementRouteArea = 31
eengDRCElementRouteBorder = 32
eengDRCElementShearingHole = 33
eengDRCElementSilkscreenOutlines = 34
eengDRCElementSilkscreenPartNumbers = 35
eengDRCElementSilkscreenRefDes = 36
eengDRCElementSkyHook = 37
eengDRCElementSoldermask = 38
eengDRCElementSolderPalette = 39
eengDRCElementSolderpaste = 40
eengDRCElementTestCoupon = 41
eengDRCElementTestPads = 42
eengDRCElementTestPointCenters = 43
eengDRCElementTestPointObstructs = 44
eengDRCElementToolingHole = 45
eengDRCElementTraceObstructs = 46
eengDRCElementTraces = 47
eengDRCElementViaHoles = 48
eengDRCElementViaObstructs = 49
eengDRCElementViaPads = 50
```

EEnginesDRCFeaturesLayer Enum

Prerequisites: None

Use this enumerator to specify the DRC features layer, using the DRCFeaturesLayer property of the ODBPPOutputEngine object.

Note:
To include all DRC feature layers, use the constant 0.

Usage

EEnginesDRCFeaturesLayer.*Constant*

Arguments

- **eengDRCFeaturesPlacementObstruct**

  The numerical value for this constant is 1.
- **eengDRCFeaturesPlaneObstruct**

  The numerical value for this constant is 2.
- **eengDRCFeaturesRouteBorder**

  The numerical value for this constant is 3.
- **eengDRCFeaturesRouteObstructTrace**

  The numerical value for this constant is 5.
- **eengDRCFeaturesRouteObstructVia**

  The numerical value for this constant is 6.
- **eengDRCFeaturesRuleArea**

  The numerical value for this constant is 7.
- **eengDRCFeaturesTestpointObstruct**

  The numerical value for this constant is 4.

EEnginesDRCMissingConductivePad Enum

Prerequisites: None

Use this enumerator to specify the DRC rules for missing conductive pads, using the MissingConductivePads property of the DRC object.

Usage

EEnginesDRCMissingConductivePad.*Constant*

Arguments

- **eengDRCMissingCondPadAllLayers**

  The numerical value for this constant is 0.
- **eengDRCMissingCondPadCoverAndInternal**

  The numerical value for this constant is 1.
- **eengDRCMissingCondPadSelectedLayers**

  The numerical value for this constant is 2.

Examples

```
‘ Enum EEnginesDRCMissingConductivePad

eengDRCMissingCondPadAllLayers = 0
eengDRCMissingCondPadCoverAndInternal = 1
eengDRCMissingCondPadSelectedLayers = 2
```

EEnginesDRCMissingCoverLayerPad Enum

Prerequisites: None

Use this enumerator to specify the DRC rules for missing cover layer pads, using the MissingCoverLayerPads property of the DRC object.

Usage

EEnginesDRCMissingCoverLayerPad.*Constant*

Arguments

- **eengDRCMissingCoverLayerPadAll**

  The numerical value for this constant is -1.
- **eengDRCMissingCoverLayerPadPart**

  The numerical value for this constant is 1.
- **eengDRCMissingCoverLayerPadVia**

  The numerical value for this constant is 2.

Examples

```
‘ Enum EEnginesDRCMissingCoverLayerPad

eengDRCMissingCoverLayerPadAll = -1
eengDRCMissingCoverLayerPadPart = 1
eengDRCMissingCoverLayerPadVia = 2
```

EEnginesDRCMissingSolderPad Enum

Prerequisites: None

Use this enumerator to specify the DRC rules for missing pads on soldermask and solderpaste, using the MissingSolderPads property of the DRC object.

Usage

EEnginesDRCMissingSolderPad.*Constant*

Arguments

- **eengDRCMissingSolderPadAll**

  The numerical value for this constant is -1 (&HFFFFFFFF).
- **eengDRCMissingSolderPadPartSoldermask**

  The numerical value for this constant is 1.
- **eengDRCMissingSolderPadSolderpaste**

  The numerical value for this constant is 4.
- **eengDRCMissingSolderPadViaSoldermask**

  The numerical value for this constant is 2.

Examples

```
‘ Enum EEnginesDRCMissingSolderPad

eengDRCMissingSolderPadAll = -1
eengDRCMissingSolderPadPartSoldermask = 1
eengDRCMissingSolderPadSolderpaste = 4
eengDRCMissingSolderPadViaSoldermask = 2
```

EEnginesDRCNetRule Enum

Prerequisites: None

Use this enumerator to specify the net rule options for DRC Connectivity and Special Rules, using the NetRules property of the DRC object.

Usage

EEnginesDRCNetRule.*Constant*

Arguments

- **eengDRCNetRuleAll**

  The numerical value for this constant is -1 (&HFFFFFFFF).

- **eengDRCNetRuleSinglePoint**

  The numerical value for this constant is 1.

- **eengDRCNetRuleUnroutedPartial**

  The numerical value for this constant is 2.

Examples

```
‘Enum EEnginesDRCNetRule

eengDRCNetRuleAll = -1
eengDRCNetRuleSinglePoint = 1
eengDRCNetRuleUnroutedPartial = 2
```

EEnginesDRCPadRule Enum

Prerequisites: None

Use this enumerator to specify the holes/pads/vias options for DRC Connectivity and Special Rules, using the PadRules property of the DRC object.

Usage

EEnginesDRCPadRule.*Constant*

Arguments

- **eengDRCPadRuleAll**

  The numerical value for this constant is -1 (&HFFFFFFFF).
- **eengDRCPadRuleDanglingViasJumpers**

  The numerical value for this constant is 1.
- **eengDRCPadRuleRoutedUnplatedPins**

  The numerical value for this constant is 2.
- **eengDRCPadRuleUnroutedPins**

  The numerical value for this constant is 4.
- **eengDRCPadRuleIncompleteTraceConnection**

  The numerical value for this constant is 8.

Examples

```
‘ Enum EEnginesDRCPadRule

eengDRCPadRuleAll = -1
eengDRCPadRuleDanglingViasJumpers = 1
eengDRCPadRuleRoutedUnplatedPins = 2
eengDRCPadRuleUnroutedPins = 4
eengDRCPadRuleIncompleteTraceConnection = 8
```

EEnginesDRCPadToPadCheck Enum

Prerequisites: None

Use this enumerator to specify the pad to pad check options, using the PadToPad property of the DRC object.

Usage

EEnginesDRCPadToPadCheck.*Constant*

Arguments

- **eengDRCDisableCheckPadsInSameCell**

  The numerical value for this constant is 1.
- **eengDRCEnableCheckPadsOfSameNet**

  The numerical value for this constant is 2.
- **eengDRCPadToPadAll**

  The numerical value for this constant is -1 (&HFFFFFFFF).

Examples

```
‘ Enum EEnginesDRCPadToPadCheck

eengDRCDisableCheckPadsInSameCell = 1
eengDRCEnableCheckPadsOfSameNet = 2
eengDRCPadToPadAll = -1
```

EEnginesDRCPlaneRule Enum

Prerequisites: None

Use this enumerator to specify the plane options for DRC Connectivity and Special Rules, using the PlaneRules property of the DRC object.

Usage

EEnginesDRCPlaneRule.*Constant*

Arguments

- **eengDRCPlaneRuleAll**

  The numerical value for this constant is -1 (&HFFFFFFFF).

- **eengDRCPlaneRuleIslands**

  The numerical value for this constant is 1.

- **eengDRCPlaneRuleUnroutedNets**

  The numerical value for this constant is 2.

Examples

```
‘ Enum EEnginesDRCPlaneRule

eengDRCPlaneRuleAll = -1
eengDRCPlaneRuleIslands = 1
eengDRCPlaneRuleUnroutedNets = 2
```

EEnginesDRCRules Enum

Prerequisites: None

Use this enumerator to specify the DRC rules to check, using the Rules property of the DRC object.

Usage

EEnginesDRCRules.*Constant*

Arguments

- **eengDRCConnectAndSpecialRules**

  This setting says enable the usage of Connectivity and Special Rules.

  The numerical value for this constant is 1.
- **eengDRCGeneralAndEleToEleRules**

  This setting says enable the checking of clearances between non-net class objects

  The numerical value for this constant is 8.
- **eengDRCNetClassClearancesAndRules**

  The numerical value for this constant is 2.
- **eengDRCPlaneClearancesAndRules**

  The numerical value for this constant is 4.
- **eengDRCRulesAll**

  The numerical value for this constant is -1 (&HFFFFFFFF).

Examples

```
‘ Enum EEnginesDRCRules

eengDRCConnectAndSpecialRules = 1
eengDRCGeneralAndEleToEleRules = 8
eengDRCNetClassClearancesAndRules = 2
eengDRCPlaneClearancesAndRules = 4
eengDRCRulesAll = -1
```

EEnginesDRCTraceRule Enum

Prerequisites: None

Use this enumerator to specify the trace options for DRC Connectivity and Special Rules, using the TraceRules property of the DRC object.

Usage

EEnginesDRCTraceRule.*Constant*

Arguments

- **eengDRCTraceRuleAll**

  The numerical value for this constant is -1 (&HFFFFFFFF).
- **eengDRCTraceRuleHangers**

  The numerical value for this constant is 1.
- **eengDRCTraceRuleLoops**

  The numerical value for this constant is 2.
- **eengDRCTraceRuleWidths**

  The numerical value for this constant is 4.

Examples

```
‘ Enum EEnginesDRCTraceRule

eengDRCTraceRuleAll = -1
eengDRCTraceRuleHangers = 1
eengDRCTraceRuleLoops = 2
eengDRCTraceRuleWidths = 4
```

EEnginesDRCViasUnder Enum

Prerequisites: None

Use this enumerator to specify the vias under options for DRC Connectivity and Special Rules, using the ViasUnder property of the DRC object.

Usage

EEnginesDRCViasUnder.*Constant*

Arguments

- **eengDRCViasUnderAll**

  The numerical value for this constant is -1 (&HFFFFFFFF).
- **eengDRCViasUnderBotPlaceOutlines**

  The numerical value for this constant is 4.
- **eengDRCViasUnderSMDPads**

  The numerical value for this constant is 1.
- **eengDRCViasUnderTopPlaceOutlines**

  The numerical value for this constant is 2.

Examples

```
‘ Enum EEnginesDRCViasUnder

eengDRCViasUnderAll = -1
eengDRCViasUnderBotPlaceOutlines = 4
eengDRCViasUnderSMDPads = 1
eengDRCViasUnderTopPlaceOutlines = 2
```

EEnginesDrillSymbol Enum

Prerequisites: None

Use this enumerator to specify the drill symbol, using the DrillSymbol property of the DrillHole object.

Usage

EEnginesDrillSymbol.*Constant*

Arguments

- **eengDrillSymbol2QuarterCircles**

  The numerical value for this constant is 18 (&H12).
- **eengDrillSymbol2Triangles**

  The numerical value for this constant is 22 (&H16).
- **eengDrillSymbol4TrianglesDiamond**

  The numerical value for this constant is 24 (&H18).
- **eengDrillSymbolCircle**

  The numerical value for this constant is 8.
- **eengDrillSymbolCircleCircle**

  The numerical value for this constant is 14 (&HE).
- **eengDrillSymbolCircleCirclePlus**

  The numerical value for this constant is 23 (&H17).
- **eengDrillSymbolCirclePlus**

  The numerical value for this constant is 2.
- **eengDrillSymbolCircleSlash**

  The numerical value for this constant is 5.
- **eengDrillSymbolCircleSquare**

  The numerical value for this constant is 19 (&H13).
- **eengDrillSymbolCircleStar**

  The numerical value for this constant is 20 (&H14).
- **eengDrillSymbolCircleTriangle**

  The numerical value for this constant is 16 (&H10).
- **eengDrillSymbolCustom**

  The numerical value for this constant is 26 (&H1A).
- **eengDrillSymbolDiamondPlus**

  The numerical value for this constant is 1.
- **eengDrillSymbolHexagon**

The numerical value for this constant is 10 (&HA).

- **eengDrillSymbolHexagonPlus**

  The numerical value for this constant is 21 (&H15).

- **eengDrillSymbolNone**

  The numerical value for this constant is 0.

- **eengDrillSymbolPlus**

  The numerical value for this constant is 25 (&H19).

- **eengDrillSymbolSquare**

  The numerical value for this constant is 9.

- **eengDrillSymbolSquareCircle**

  The numerical value for this constant is 4.

- **eengDrillSymbolSquareDiamond**

  The numerical value for this constant is 3.

- **eengDrillSymbolSquarePlus**

  The numerical value for this constant is 13 (&HD).

- **eengDrillSymbolSquareSlash**

  The numerical value for this constant is 15 (&HF).

- **eengDrillSymbolSquareStar**

  The numerical value for this constant is 17 (&H11).

- **eengDrillSymbolSquareTriangle**

  The numerical value for this constant is 11 (&HB).

- **eengDrillSymbolStarCircle**

  The numerical value for this constant is 12 (&HC).

- **eengDrillSymbolTriangle**

  The numerical value for this constant is 7.

- **eengDrillSymbolX**

  The numerical value for this constant is 5.

EEnginesDrillSymbolType Enum

Prerequisites: None

Drill symbol type constants.

Usage

EEnginesDrillSymbolType.*Constant*

Arguments

- **eengDrillSymbolTypeAutomatic**

  The numerical value for this constant is 8.
- **eengDrillSymbolTypeCharacter**

  The numerical value for this constant is 4.
- **eengDrillSymbolTypeChartSymbol**

  The numerical value for this constant is 1.
- **eengDrillSymbolTypeCustomSymbol**

  The numerical value for this constant is 2.
- **eengDrillSymbolTypeNone**

  The numerical value for this constant is 0.

EEnginesElementType Enum

Prerequisites: None

Use this enumerator to specify the DXF Export element type, using the PutElementExport or PutWBElementExport methods of the DXFExport object.

Usage

EEnginesElementType.*Constant*

Arguments

Constant	Value
eengElementTypeAdhesive	111 (&H6F)
eengElementTypeAll	0
eengElementTypeAssemblyOutlineBottom	1
eengElementTypeAssemblyOutlineTop	2
eengElementTypeAssemblyPartNumberBottom	3
eengElementTypeAssemblyPartNumberTop	4
eengElementTypeAssemblyRefDesBottom	5
eengElementTypeAssemblyRefDesTop	6
eengElementTypeBadBoardIdentifierBottom	7
eengElementTypeBadBoardIdentifierTop	8
eengElementTypeBendAreas	110 (&H6E)
eengElementTypeBoardCavity	9
eengElementTypeBoardOrigin	10 (&HA)
eengElementTypeBoardOutline	11 (&HB)
eengElementTypeBondPads	12 (&HC)
eengElementTypeBondWires	13 (&HD)
eengElementTypeBreakAwayTabBottom	14 (&HE)
eengElementTypeBreakawayTabTop	15 (&HF)
eengElementTypeConductiveShape	16 (&H10)

Constant	Value
eengElementTypeContour	17 (&H11)
eengElementTypeCopperBalancing	18 (&H12)
eengElementTypeCoverlay	112 (&H70)
eengElementTypeCoverlayPads	113 (&H71)
eengElementTypeDesignRuleArea	19(&H13)
eengElementTypeDiePins	20(&H14)
eengElementTypeDrillDrawing	21(&H15)
eengElementTypeElectricalTestIdentifierBottom	22(&H16)
eengElementTypeElectricalTestIdentifierTop	23(&H17)
eengElementTypeEtchedText	24(&H18)
eengElementTypeFiducialPadBottom	25(&H19)
eengElementTypeFiducialPadTeardropBottom	27(&H1B)
eengElementTypeFiducialPadTeardropTop	28(&H1C)
eengElementTypeFiducialPadTop	26(&H1A)
eengElementTypeFloorPlanRooms	29(&H1D)
eengElementTypeGeneralPanelCellBottom	30(&H1E)
eengElementTypeGeneralPanelCellTop	31(&H1F)
eengElementTypeGeneratedSilkscreenBottom	32(&H20)
eengElementTypeGeneratedSilkscreenTop	33(&H21)
eengElementTypeGlueSpotBottom	34(&H22)
eengElementTypeGlueSpotTop	35(&H23)
eengElementTypeInsertionOutlineBottom	36(&H24)
eengElementTypeInsertionOutlineTop	37(&H25)
eengElementTypeJumperWireBottom	38(&H26)
eengElementTypeJumperWireTop	39(&H27)
eengElementTypeManufacturingOutline	40(&H28)
eengElementTypeMetalization	41(&H29)

Constant	Value
eengElementTypeMountingHole	42(&H2A)
eengElementTypeMountingHolePad	43(&H2B)
eengElementTypeMountingHoleTearDrop	44(&H2C)
eengElementTypeNamedSoldermask	114 (&H72)
eengElementTypeNamedSoldermaskPads	115 (&H73)
eengElementTypeNCDrillOrigin	45(&H2D)
eengElementTypePanelBorder	46(&H2E)
eengElementTypePanelIdentifierBottom	47(&H2F)
eengElementTypePanelIdentifierTop	48(&H30)
eengElementTypePanelObstruct	49(&H31)
eengElementTypePanelOrigin	50(&H32)
eengElementTypePanelOutline	51(&H33)
eengElementTypePartPads	52(&H34)
eengElementTypePartPadSMDBottom	54(&H36)
eengElementTypePartPadSMDTeardropsBottom	56(&H38)
eengElementTypePartPadSMDTeardropsTop	57(&H39)
eengElementTypePartPadSMDTOP	55(&H37)
eengElementTypePartPadsTeardrops	53(&H35)
eengElementTypePartPinHoles	58(&H3A)
eengElementTypePinNumbersBottom	59(&H3B)
eengElementTypePinNumbersTop	60(&H3C)
eengElementTypePlacementKeepouts	61(&H3D)
eengElementTypePlacementOutlinesBottom	62(&H3E)
eengElementTypePlacementOutlinesTop	63(&H3F)
eengElementTypePlaneData	64(&H40)
eengElementTypePlaneDataTeardrops	65(&H41)
eengElementTypePlaneNoConnects	66(&H42)

Constant	Value
eengElementTypePlaneObstructs	67(&H43)
eengElementTypePlaneShapes	68(&H44)
eengElementTypeRegistrationPinBottom	69(&H45)
eengElementTypeRegistrationPinTop	70(&H46)
eengElementTypeReservedAreas	71(&H47)
eengElementTypeResistorAreas	72(&H48)
eengElementTypeRouteBorder	73(&H49)
eengElementTypeRouteObstructs	74(&H4A)
eengElementTypeShearingHoles	75(&H4B)
eengElementTypeSilkscreenOutlineBottom	76(&H4C)
eengElementTypeSilkscreenOutlineTop	77(&H4D)
eengElementTypeSilkscreenPartNumberBottom	78(&H4E)
eengElementTypeSilkscreenPartNumberTop	79(&H4F)
eengElementTypeSilkscreenRefDesBottom	80(&H50)
eengElementTypeSilkscreenRefDesTop	81(&H51)
eengElementTypeSoldermaskBottom	82(&H52)
eengElementTypeSoldermaskPadsBottom	84(&H54)
eengElementTypeSoldermaskPadsTeardropBottom	86(&H56)
eengElementTypeSoldermaskPadsTeardropTop	87(&H57)
eengElementTypeSoldermaskPadsTop	85(&H55)
eengElementTypeSoldermaskTop	83(&H53)
eengElementTypeSolderpasteBottom	88(&H58)
eengElementTypeSolderpastePadsBottom	90(&H5A)
eengElementTypeSolderpastePadsTeardropBottom	92(&H5C)
eengElementTypeSolderpastePadsTeardropTop	93(&H5D)
eengElementTypeSolderpastePadsTop	91(&H5B)
eengElementTypeSolderpasteTop	89(&H59)

Constant	Value
eengElementTypeStiffener	116 (&H74)
eengElementTypeStiffenerPads	117 (&H75)
eengElementTypeTestCouponBottom	94(&H5E)
eengElementTypeTestCouponTop	95(&H5F)
eengElementTypeTestpointAssemblyRefDesBottom	96(&H60)
eengElementTypeTestpointAssemblyRefDesTop	97(&H61)
eengElementTypeTestpointObstructs	98(&H62)
eengElementTypeTestpointPadBottom	99(&H63)
eengElementTypeTestpointPadTeardropBottom	101(&H65)
eengElementTypeTestpointPadTeardropTop	102(&H66)
eengElementTypeTestpointPadTop	100(&H64)
eengElementTypeTestpointSilkscreenRefDesBottom	103(&H67)
eengElementTypeTestpointSilkscreenRefDesTop	104(&H68)
eengElementTypeToolingHoles	105(&H69)
eengElementTypeTraces	106(&H6A)
eengElementTypeUserElements	150(&H96)
eengElementTypeViaHoles	107(&H6B)
eengElementTypeViaPads	108(&H6C)
eengElementTypeViaPadsTeardrops	109(&H6D)

EEnginesErrCode Enum

Prerequisites: None

Error Code constants.

Usage

EEnginesErrCode.*Constant*

Arguments

- **eengErrCodeAutomationDisabled**

  The numerical value for this constant is -2147220930 (&H8004023E).
- **eengErrCodeCannotOpenPanel**

  The numerical value for this constant is -2147220951 (&H80040229).
- **eengErrCodeCantChangeDrillSymbolForHole**

  The numerical value for this constant is -2147220945 (&H8004022F).
- **eengErrCodeCantCreateEngineProcess**

  The engine could not be executed from the Go Method.

  The numerical value for this constant is -2147220983 (&H80040209).
- **eengErrCodeCantFindEngineExecutable**

  The engine executable could not be found.

  The numerical value for this constant is -2147220984 (&H80040208).
- **eengErrCodeCantGenerateOutput**

  The numerical value for this constant is -2147220982 (&H8004020A).
- **eengErrCodeCantInitJobPrefsDb**

  Cannot initialize the JobPrefs Db. Check the path on the design

  The numerical value for this constant is -2147220989 (&H80040203).
- **eengErrCodeCantLoadJobPrefsDb**

  Cannot initialize the JobPrefs Db. Check the path on the design

  The numerical value for this constant is -2147220988 (&H80040204).
- **eengErrCodeCantLoadLayoutDB**

  The numerical value for this constant is -2147220947 (&H8004022D).
- **eengErrCodeCantOpenFile**

  Cannot initialize the JobPrefs Db. Check the path on the design

  The numerical value for this constant is -2147220968 (&H80040218).
- **eengErrCodeDatabaseUpgradeRequired**

  The numerical value for this constant is -2147220933 (&H8004023B).

- **eengErrCodeDesignNotProvided**

  Executing a command before the design filename is specified.

  The numerical value for this constant is -2147220980 (&H8004020C).
- **eengErrCodeDRCExceedHazards**

  The numerical value for this constant is -2147220972 (&H80040214).
- **eengErrCodeDRCFailed**

  The numerical value for this constant is -2147220971 (&H80040215).
- **eengErrCodeDRCHazardsFound**

  The numerical value for this constant is -2147220973 (&H80040213).
- **eengErrCodeExtendedPrintFailed**

  The numerical value for this constant is -2147220952 (&H80040228).
- **eengErrCodeFablinkXEVariantDataOutOfDate**

  The numerical value for this constant is -2147220934 (&H8004023A).
- **eengErrCodeFunctionNotSupported**

  The numerical value for this constant is -2147220969 (&H80040217).
- **eengErrCodeHoleInvalidDrillSymbolSize**

  The numerical value for this constant is -2147220946 (&H8004022E).
- **eengErrCodeHoleInvalidOptionCombChrSz**

  The numerical value for this constant is -2147220926 (&H80040242).
- **eengErrCodeIDFExportFailed**

  The numerical value for this constant is -2147220956 (&H80040224).
- **eengErrCodeInvalidBoardItemParameter**

  The numerical value for this constant is --2147220954 (&H80040226).
- **eengErrCodeInvalidDRCElement**

  The numerical value for this constant is -2147220967 (&H80040219).
- **eengErrCodeInvalidColumnType**

  The numerical value for this constant is -2147220943 (&H80040231).
- **eengErrCodeInvalidDrillChartDigits**

  The numerical value for this constant is -2147220975 (&H80040211).
- **eengErrCodeInvalidDrillChartPrecision**

  The numerical value for this constant is -2147220974 (&H80040212).
- **eengErrCodeInvalidDrillHoleSize**

  The numerical value for this constant is -2147220977 (&H8004020F).
- **eengErrCodeInvalidDrillMachineDataFormat**

  The numerical value for this constant is 2147220932 (&H8004023C).
- **eengErrCodeInvalidDrillSymbol**

The numerical value for this constant is -2147220944 (&H80040230).

- **eengErrCodeInvalidElementTypeForFlexDesign**

  The numerical value for this constant is -2147220929 (&H8004023F)

- **eengErrCodeInvalidElementTypeForNonFlexDesign**

  The numerical value for this constant is -2147220928 (&H80040240)

- **eengErrCodeInvalidElementTypeForPcbDesign**

  The numerical value for this constant is -2147220942 (&H80040232).

- **eengErrCodeInvalidElementTypeNoWireBondsFound**

  The numerical value for this constant is -2147220939 (&H80040235).

- **eengErrCodeInvalidFileExtension**

  The numerical value for this constant is -2147220976 (&H80040210).

- **eengErrCodeInvalidLayer**

  The numerical value for this constant is -2147220941 (&H80040233).

- **eengErrCodeInvalidLayerSpan**

  The numerical value for this constant is -2147220953 (&H80040227).

- **eengErrCodeInvalidMachineDataFormat**

  The numerical value for this constant is -2147220979 (&H8004020D).

- **eengErrCodeInvalidMachineUnit**

  The numerical value for this constant is -2147220978 (&H8004020E).

- **eengErrCodeInvalidParameter**

  The numerical value for this constant is -2147220991 (&H80040201).

- **eengErrCodeInvalidParameterGerberBoard**

  The parameter is not valid for a board design(.pcb).

  The numerical value for this constant is -2147220966 (&H8004021A).

- **eengErrCodeInvalidParameterGerberPanel**

  The parameter is not valid for a panel design(.pnl).

  The numerical value for this constant is -2147220965 (&H8004021B).

- **eengErrCodeInvalidParameterPDFBoard**

  The numerical value for this constant is -2147220964 (&H8004021C).

- **eengErrCodeInvalidParameterPDFPanel**

  The parameter is only available for a PDF board design(.pcb)

  The numerical value for this constant is -2147220963 (&H8004021D).

- **eengErrCodeInvalidParameterSilkscreenGenBoard**

  The numerical value for this constant is -2147220960 (&H80040220).

- **eengErrCodeInvalidParameterSilkscreenGenPanel**

  The numerical value for this constant is -2147220959 (&H80040221).

- **eengErrCodeLicenseStillRequired**

  The numerical value for this constant is -2147220955 (&H80040225).
- **eengErrCodeMethodNotValidForPanelDesign**

  The numerical value for this constant is -2147220949 (&H8004022B).
- **eengErrCodeMissingPlaneData**

  The numerical value for this constant is -2147220985 (&H80040207).
- **eengErrCodeMOVFailed**

  The numerical value for this constant is -2147220970 (&H80040216).
- **eengErrCodeNoDCodeMappingFile**

  The numerical value for this constant is -2147220987 (&H80040205).
- **eengErrCodeNoDefaultDCodeMappingFile**

  The numerical value for this constant is -2147220981 (&H8004020B)
- **eengErrCodeNoJobPrefsDbFile**

  Cannot find the JobPrefs DB file. Check design filename.

  The numerical value for this constant is -2147220990 (&H80040202)
- **eengErrCodeNoLicenseFound**

  Could not acquire an Automation Pro license(xeautoextended).

  The numerical value for this constant is -2147220961 (&H8004021F)
- **eengErrCodeNoProductLicenseFound**

  The numerical value for this constant is -2147220931 (&H8004023D)
- **eengErrCodeNoUserInfo**

  The numerical value for this constant is -2147220962 (&H8004021E)
- **eengErrCodeNoUserLayerSpecified**

  The numerical value for this constant is -2147220958 (&H80040222)
- **eengErrCodeOutputFileNotSet**

  The numerical value for this constant is -2147220957 (&H80040223)
- **eengErrCodePDFMethodNotValidIfExtendedPrint**

  The numerical value for this constant is -2147220948 (&H8004022C)
- **eengErrCodePlaneDataCheck**

  The numerical value for this constant is -2147220986 (&H80040206)
- **eengErrCodeVariantNotFoundInDesign**

  The numerical value for this constant is -2147220950 (&H8004022A)

Examples

```
' Enum EEnginesErrCode
```

```
eengErrCodeCantCreateEngineProcess = -2147220983
eengErrCodeCantFindEngineExecutable = -2147220984
eengErrCodeCantGenerateOutput = -2147220982
eengErrCodeCantInitJobPrefsDb = -2147220989
eengErrCodeCantLoadJobPrefsDb = -2147220988
eengErrCodeCantOpenFile = -2147220968
eengErrCodeDesignNotProvided = -2147220980
eengErrCodeDRCExceedHazards = -2147220972
eengErrCodeDRCFailed = -2147220971
eengErrCodeDRCHazardsFound = -2147220973
eengErrCodeFunctionNotSupported = -2147220969
eengErrCodeIDFExportFailed = -2147220956
eengErrCodeInvalidDRCElement = -2147220967
eengErrCodeInvalidDrillChartDigits = -2147220975
eengErrCodeInvalidDrillChartPrecision = -2147220974
eengErrCodeInvalidDrillHoleSize = -2147220977
eengErrCodeInvalidFileExtension = -2147220976
eengErrCodeInvalidMachineDataFormat = -2147220979
eengErrCodeInvalidMachineUnit = -2147220978
eengErrCodeInvalidParameter = -2147220991
eengErrCodeInvalidParameterGerberBoard = -2147220966
eengErrCodeInvalidParameterGerberPanel = -2147220965
eengErrCodeInvalidParameterPDFBoard = -2147220964
eengErrCodeInvalidParameterPDFPanel = -2147220963
eengErrCodeInvalidParameterSilkscreenGenBoard = -2147220960
eengErrCodeInvalidParameterSilkscreenGenPanel = -2147220959
...
```

EEnginesErrorSeverity Enum

Prerequisites: None

Use this enumerator to identify the severity of an error, using the Severity property of the Error object.

Usage

EEnginesErrorSeverity.*Constant*

Arguments

- **eengError**

  The numerical value for this constant is 1.
- **eengFatalError**

  The numerical value for this constant is 2.
- **eengWarning**

  The numerical value for this constant is 0.

Examples

```
‘ Enum EEnginesErrorSeverity

eengError = 1
eengFatalError = 2
eengWarning = 0
```

EEnginesExtPrintFileFormat Enum

Prerequisites: None

Use this enumerator to specify the Extended Print output type, using the FileFormat property of the ExtendedPrint object.

Usage

EEnginesExtPrintFileFormat.*Constant*

Arguments

- **eengExtPrintFormatHPGL2**

  The numerical value for this constant is 4.
- **eengExtPrintFormatPDF**

  The numerical value for this constant is 1.
- **eengExtPrintFormatPS2**

  The numerical value for this constant is 2.
- **eengExtPrintFormatPS3**

  The numerical value for this constant is 3.

EEnginesGeneralInterfaceIdentifyPartsType Enum

Prerequisites: None

Use this enumerator to specify the identify parts option, using the IdentifyPartBy property of the GeneralInterfaceEngine object.

Usage

EEnginesGeneralInterfaceIdentifyPartsType.*Constant*

Arguments

- **eengGenIntPartLabel**

  The numerical value for this constant is 2.
- **eengGenIntPartName**

  Fabmaster and C-Link only.

  The numerical value for this constant is 3.
- **eengGenIntPartNumber**

  The numerical value for this constant is 1.

Examples

```
‘ Enum EEnginesGeneralInterfaceIdentifyPartsType

eengGenIntPartLabel = 2
eengGenIntPartName = 3
eengGenIntPartNumber = 1
```

EEnginesGeneralInterfaceOutputType Enum

Prerequisites: None

Use this enumerator to specify the output type, using the OutputType property of the GeneralInterfaceEngine object.

Usage

EEnginesGeneralInterfaceOutputType.*Constant*

Arguments

- **eengGenIntOutputBetasoft**

  The numerical value for this constant is 1.
- **eengGenIntOutputCLink**

  The numerical value for this constant is 7.
- **eengGenIntOutputConnList**

  The numerical value for this constant is 5.
- **eengGenIntOutputFabMaster**

  The numerical value for this constant is 6.
- **eengGenIntOutputGenCad**

  The numerical value for this constant is 4.
- **eengGenIntOutputGenericAIS**

  The numerical value for this constant is 2.
- **eengGenIntOutputGenericATE**

  The numerical value for this constant is 3.
- **eengGenIntOutputIPCD356**

  The numerical value for this constant is 8.

Examples

```
‘ Enum EEnginesGeneralInterfaceOutputType

eengGenIntOutputBetasoft = 1
eengGenIntOutputCLink = 7
eengGenIntOutputConnList = 5
eengGenIntOutputFabMaster = 6
eengGenIntOutputGenCad = 4
eengGenIntOutputGenericAIS = 2
eengGenIntOutputGenericATE = 3
eengGenIntOutputIPCD356 = 8
```

EEnginesGeneralInterfaceUnitsType Enum

Prerequisites: None

Use this enumerator to specify the units type, using the UnitsType property of the GeneralInterfaceEngine object.

Usage

EEnginesGeneralInterfaceUnitsType.*Constant*

Arguments

- **eengGeneralInterfaceUnitOfOne**

  The numerical value for this constant is 1.
- **eengGeneralInterfaceUnitOfOneOneHundredth**

  The numerical value for this constant is 3.
- **eengGeneralInterfaceUnitOfOneTenth**

  The numerical value for this constant is 2.

Examples

```
‘ Enum EEnginesGeneralInterfaceUnitsType

eengGeneralInterfaceUnitOfOne = 1
eengGeneralInterfaceUnitOfOneOneHundredth = 3
eengGeneralInterfaceUnitOfOneTenth = 2
```

EEnginesIDFExportComponentSideType Enum

Prerequisites: None

Use this enumerator to specify the mountside of the components to export, using the ComponentSide property of the IDFExport object (for panel only).

Usage

EEnginesIDFExportComponentSideType.*Constant*

Arguments

- **eengComponentBottomOnly**

  The numerical value for this constant is 1.
- **eengComponentTopAndBottom**

  The numerical value for this constant is 2.
- **eengComponentTopOnly**

  The numerical value for this constant is 0.

Examples

```
‘ Enum EEnginesIDFExportComponentSideType

eengComponentBottomOnly = 1
eengComponentTopAndBottom = 2
eengComponentTopOnly = 0
```

EEnginesIDFExportPackageOutlineType Enum

Prerequisites: None

Use this enumerator to specify the package outline layer, using the PackageOutlineLayer property of the IDFExport object.

Usage

EEnginesIDFExportPackageOutlineType.*Constant*

Arguments

- **eengIDFPackageOutlineAssemblyOutline**

  The numerical value for this constant is 1.
- **eengIDFPackageOutlineInsertionOutline**

  The numerical value for this constant is 2.
- **eengIDFPackageOutlinePlacementOutline**

  The numerical value for this constant is 0.

Examples

```
‘ Enum EEnginesIDFExportPackageOutlineType

eengIDFPackageOutlineAssemblyOutline = 1
eengIDFPackageOutlineInsertionOutline = 2
eengIDFPackageOutlinePlacementOutline = 0
```

EEnginesIDFExportSectionsType Enum

Prerequisites: None

Use this enumerator to specify the section type of the items to export, using the ExportSection property of the IDFExport object.

Usage

EEnginesIDFExportSectionsType.*Constant*

Arguments

- **eengBoardCavities**

  The numerical value for this constant is 7.
- **eengComponents**

  The numerical value for this constant is 5.
- **eengDrillHoles**

  The numerical value for this constant is 4.
- **eengOtherOutline**

  The numerical value for this constant is 6.
- **eengPanelBorder**

  The numerical value for this constant is 1.
- **eengPanelObstruct**

  The numerical value for this constant is 2.
- **eengPanelOutline**

  The numerical value for this constant is 0.
- **eengRouteObstruct**

  The numerical value for this constant is 3.
- **eengSlotHoleMntHoleAsContour**

  The numerical value for this constant is 8.

Examples

```
‘ Enum EEnginesIDFExportSectionsType

eengComponents = 5
eengDrillHoles = 4
eengOtherOutline = 6
eengPanelBorder = 1
eengPanelObstruct = 2
eengPanelOutline = 0
eengRouteObstruct = 3
```

EEnginesIDFVersionType Enum

Prerequisites: None

Use this enumerator to specify the IDF version, using the Version property of the IDFExport object.

Usage

EEnginesIDFVersionType.*Constant*

Arguments

- **eengVersion2**

  The numerical value for this constant is 1.
- **eengVersion3**

  The numerical value for this constant is 2.

Examples

```
‘ Enum EEnginesIDFVersionType

eengVersion2 = 1
eengVersion3 = 2
```

EEnginesMappedUserLayer Enum

Prerequisites: None

Use this enumerator to specify user layer mapping. Use 1- 1000 for Layer1 - Layer 1000.

Note:
The maximum number of supported user-defined layers is 1000.

Usage

EEnginesMappedUserLayer.*Constant*

Arguments

- **eengMappedBottomAssembly**

  The numerical value for this constant is 408 (H198).
- **eengMappedBottomSilkscreen**

  The numerical value for this constant is 404 (H194).
- **eengMappedBottomSolderMask**

  The numerical value for this constant is 406 (H196).
- **eengMappedBottomSolderPaste**

  The numerical value for this constant is 401 (H191).
- **eengMappedDocument**

  The numerical value for this constant is 402 (H192).
- **eengMappedTopAssembly**

  The numerical value for this constant is 407 (H197).
- **eengMappedTopSilkscreen**

  The numerical value for this constant is 403 (H193).
- **eengMappedTopSolderMask**

  The numerical value for this constant is 405 (H195).
- **eengMappedTopSolderPaste**

  The numerical value for this constant is 400 (H190).

Examples

```
 ‘ Enum EEnginesMappedUserLayer

eengMappedBottomAssembly = 408
eengMappedBottomSilkscreen = 404
eengMappedBottomSolderMask = 406
eengMappedBottomSolderPaste = 401
```

```
eengMappedDocument = 402
eengMappedTopAssembly = 407
eengMappedTopSilkscreen = 403
eengMappedTopSolderMask = 405
eengMappedTopSolderPaste = 400
```

EEnginesNCDrillChartUnits Enum

Prerequisites: None

Use this enumerator to specify the drill chart units, using the UnitEx property of the NCDrillChart object.

Usage

EEnginesNCDrillChartUnits.*Constant*

Arguments

- **eengChartUnitsInch**

  The numerical value for this constant is 0.
- **eengChartUnitsMils**

  The numerical value for this constant is 1.
- **eengChartUnitsMM**

  The numerical value for this constant is 2.
- **eengChartUnitsUM**

  The numerical value for this constant is 3.

Examples

‘ Enum NcDrillChartUnits

```
eengChartUnitsInch = 0eengChartUnitsMils = 1eengChartUnitsMM =
 2eengChartUnitsUM = 3
```

EEnginesNCDrillColumn Enum

Prerequisites: None

Use this enumerator to identify the drill chart column type, using the ColumnType property of the NCDrillChartColumn object.

Usage

EEnginesNCDrillColumn.*Constant*

Arguments

- **eengColumnDiameter**

  The numerical value for this constant is 1.
- **eengColumnDiameterRange**

  The numerical value for this constant is 7.
- **eengColumnFCF**

  The numerical value for this constant is 8.
- **eengColumnHolename**

  The numerical value for this constant is 5.
- **eengColumnPlated**

  The numerical value for this constant is 3.
- **eengColumnPunched**

  The numerical value for this constant is 4.
- **eengColumnQuantity**

  The numerical value for this constant is 6.
- **eengColumnSpan**

  The numerical value for this constant is 10 (HA).
- **eengColumnSymbol**

  The numerical value for this constant is 0.
- **eengColumnTolerance**

  The numerical value for this constant is 2.
- **eengColumnType**

  The numerical value for this constant is 11 (HB).
- **eengColumnUser**

  The numerical value for this constant is 9.

Examples

```
‘ Enum EEnginesNCDrillColumn

eengColumnDiameter = 1
eengColumnHolename = 5
eengColumnPlated = 3
eengColumnPunched = 4
eengColumnQuantity = 6
eengColumnSymbol = 0
eengColumnTolerance = 2
```

EEnginesNCDrillDataType Enum

Prerequisites: None

Use this enumerator to specify the NC drill data type, using the DataType property of the NCDrillMachineFormat object.

Usage

EEnginesNCDrillDataType.*Constant*

Arguments

- **eengExcellon**

  The numerical value for this constant is 0.

- **eengTruDrill**

  The numerical value for this constant is 1.

Examples

```
‘ Enum EEnginesNCDrillDataType

eengExcellon = 0
eengTruDrill = 1
```

EEnginesNCDrillSweepAxis Enum

Prerequisites: None

Use this enumerator to specify the drill sweep axis, using the SweepAxis property of the NCDrillParameters object.

Usage

EEnginesNCDrillSweepAxis.*Constant*

Arguments

- **eengSweepaxisHorizontal**

  The numerical value for this constant is 0.

- **eengSweepaxisVertical**

  The numerical value for this constant is 1.

Examples

```
‘ Enum EEnginesNCDrillSweepAxis

eengSweepaxisHorizontal = 0
eengSweepaxisVertical = 1
```

EEnginesNCDrillUnits Enum

Prerequisites: None

Use this enumerator to specify the NC drill chart units, using the UnitEx property of the NCDrillChart object.

Usage

EEnginesNCDrillUnits.*Constant*

Arguments

- **eengUnitsEnglish**

  The numerical value for this constant is 0.

- **eengUnitsMetric**

  The numerical value for this constant is 1.

Examples

```
‘ Enum EEnginesNCDrillUnits

eengUnitsEnglish = 0
eengUnitsMetric = 1
```

EEnginesNonFuncPad Enum

Prerequisites: None

Use this enumerator to specify the options for non-functional pad removal for pins and vias.

Usage

EEnginesNonFuncPad.*Constant*

Arguments

- **eengNonFuncRemoveNone**

  The numerical value for this constant is 0.
- **eengNonFuncRemoveNoNet**

  Valid only for pins.

  The numerical value for this constant is 2.
- **eengNonFuncRemoveUnused**

  The numerical value for this constant is 1.

Examples

```
‘ Enum EEnginesNonFuncPad

eengNonFuncRemoveNone = 0
eengNonFuncRemoveNoNet = 2
eengNonFuncRemoveUnused = 1
```

EEnginesODBPPExportMode Enum

Prerequisites: None

Use this enumerator to specify the type of data to export, using the ODBPPExportMode property of the ODBPPOutputEngine object.

Usage

EEnginesODBPPExportMode.*Constant*

Arguments

- **eengAdvancedPackagingExport**

  The numerical value for this constant is 3.
- **eengAssemblyExport**

  The numerical value for this constant is 2.
- **eengEmbeddedPassivesExport**

  The numerical value for this constant is 4.
- **eengFabExport**

  The numerical value for this constant is 1.
- **eengFullExport**

  The numerical value for this constant is 0.

Examples

```
‘ Enum EEnginesODBPPExportMode

eengAssemblyExport = 2
eengFabExport = 1
eengFullExport = 0
```

EEnginesOutputUnits Enum

Prerequisites: None

Use this enumerator to specify the output units for ODB++, using the Units property of the ODBPPOutputEngine object.

Usage

EEnginesOutputUnits.*Constant*

Arguments

- **eengOutputUnitsEnglish**

  The numerical value for this constant is 0.
- **eengOutputUnitsMetric**

  The numerical value for this constant is 1.

EEnginesPackageOutlineLayer Enum

Prerequisites: None

Use this enumerator to specify the package outline layer, using the PackageOutlineLayer property of the ODBPPOutputEngine object.

Usage

EEnginesPackageOutlineLayer.*Constant*

Arguments

- **eengPackageOutlineAssemblyOutline**

  The numerical value for this constant is 1.
- **eengPackageOutlineDefaultUserLayer**

  The numerical value for this constant is 4.
- **eengPackageOutlineInsertionOutline**

  The numerical value for this constant is 2.
- **eengPackageOutlinePanelOutline**

  Valid only for panels.

  The numerical value for this constant is 5.
- **eengPackageOutlinePanelText**

  Valid only for panels.

  The numerical value for this constant is 6.
- **eengPackageOutlinePlacementOutline**

  The numerical value for this constant is 0.
- **eengPackageOutlineSilkscreen**

  The numerical value for this constant is 3.

Examples

```
‘ Enum EEnginesPackageOutlineLayer

eengPackageOutlineAssemblyOutline = 1
eengPackageOutlineDefaultUserLayer = 4
eengPackageOutlineInsertionOutline = 2
eengPackageOutlinePanelOutline = 5
eengPackageOutlinePanelText = 6
eengPackageOutlinePlacementOutline = 0
eengPackageOutlineSilkscreen = 3
```

EEnginesPDFSheetPageSize Enum

Prerequisites: None

This enumerator specifies the page size for the PDF output sheet.

Usage

EEnginesPDFSheetPageSize.*Constant*

Arguments

- **ePDFA0**

  The numerical value for this constant is 3.
- **ePDFA1**

  The numerical value for this constant is 4.
- **ePDFA2**

  The numerical value for this constant is 5.
- **ePDFA3**

  The numerical value for this constant is 6.
- **ePDFA4**

  The numerical value for this constant is 7.
- **ePDFB5**

  The numerical value for this constant is 8.
- **ePDFC5**

  The numerical value for this constant is 9.
- **ePDFD**

  The numerical value for this constant is 10 (&HA).
- **ePDFE**

  The numerical value for this constant is 11 (&HB).
- **ePDFLegal**

  The numerical value for this constant is 2.
- **ePDFLetter**

  The numerical value for this constant is 1.
- **ePDFNoneSize**

  The numerical value for this constant is 0.

Examples

```
‘ Enum EEnginesPDFSheetPageSize

ePDFA0 = 3
ePDFA1 = 4
ePDFA2 = 5
ePDFA3 = 6
ePDFA4 = 7
ePDFB5 = 8
ePDFC5 = 9
ePDFD = 10
ePDFE = 11
ePDFLegal = 2
ePDFLetter = 1
ePDFNoneSize = 0
```

EEnginesPlotColor Enum

Prerequisites: None

This enumerator specifies PDF plot color settings.

Usage

EEnginesPlotColor.*Constant*

Arguments

- **eengPlotColorBlackAndWhite**

  The numerical value for this constant is 1.
- **eengPlotColorColor**

  The numerical value for this constant is 2.
- **eengPlotColorGreyScale**

  The numerical value for this constant is 3.

EEnginesPolygonFill Enum

Prerequisites: None

Use this enumerator to specify the Gerber polygon shape fill, using the PolygonFillMethod property of the GerberMachineFormat object.

Usage

EEnginesPolygonFill.*Constant*

Arguments

- **eengPolygonFillDraw**

  The numerical value for this constant is 1.

- **eengPolygonFillRaster**

  The numerical value for this constant is 0.

Examples

```
‘ Enum EEnginesPolygonFill

eengPolygonFillDraw = 1
eengPolygonFillRaster = 0
```

EEnginesSide Enum

Prerequisites: None

Use this enumerator to specify the placement side, using the PutWBElementExport method of the DXFExport object.

Usage

EEnginesSide.*Constant*

Arguments

- **eengSideBoth**

  The numerical value for this constant is 2.
- **eengSideBottom**

  The numerical value for this constant is 1.
- **eengSideTop**

  The numerical value for this constant is 0.

EEnginesSilkScreenBreakType Enum

Prerequisites: None

Use this enumerator to specify the silkscreen break type, using the BreakSilkscreen property of the SilkscreenGen object.

Usage

EEnginesSilkScreenBreakType.*Constant*

Arguments

- **eengConductivePads**

  The numerical value for this constant is 0.

- **eengSolderMaskPads**

  The numerical value for this constant is 1.

EEnginesSilkscreenDesignLayer Enum

Prerequisites: None

Use this enumerator to specify the silkscreen layers for selected package groups.

Usage

EEnginesSilkscreenDesignLayer.*Constant*

Arguments

- **eengSilkscreenBoardOutline**

  The numerical value for this constant is 3.
- **eengSilkscreenOutline**

  The numerical value for this constant is 2.
- **eengSilkscreenPanelOutline**

  The numerical value for this constant is 4.
- **eengSilkscreenPartNumbers**

  The numerical value for this constant is 1.
- **eengSilkscreenReferenceDesignators**

  The numerical value for this constant is 0.

Examples

```
‘ Enum EEnginesSilkscreenDesignLayer

eengSilkscreenBoardOutline = 3
eengSilkscreenOutline = 2
eengSilkscreenPanelOutline = 4
eengSilkscreenPartNumbers = 1
eengSilkscreenReferenceDesignators = 0
```

EEnginesStepMode Enum

Prerequisites: None

Use this enumerator to specify Gerber and NC Drill step mode options.

Usage

EEnginesStepMode.*Constant*

Arguments

- **eengStepAbsolute**

  The numerical value for this constant is 0.
- **eengStepIncremental**

  The numerical value for this constant is 1.

Examples

```
‘ Enum EEnginesStepMode

eengStepAbsolute = 0
eengStepIncremental = 1
```

EEnginesUnit Enum

Prerequisites: None

MGCPCBEngines Unit constants.

Usage

EEnginesUnit.*Constant*

Arguments

- **eengUnitCurrent**

  The numerical value for this constant is 0.
- **eengUnitDatabase**

  The numerical value for this constant is 1.
- **eengUnitInch**

  The numerical value for this constant is 3.
- **eengUnitMils**

  The numerical value for this constant is 2.
- **eengUnitMM**

  The numerical value for this constant is 4.
- **eengUnitUM**

  The numerical value for this constant is 5.

Examples

```
‘ Enum EEnginesUnit

eengUnitCurrent = 0
eengUnitDatabase = 1
eengUnitInch = 3
eengUnitMils = 2
eengUnitMM = 4
eengUnitUM = 5
```

EEnginesZeroTruncation Enum

Prerequisites: None

Use this enumerator to specify Gerber and NC Drill zero truncation options.

Usage

EEnginesZeroTruncation.*Constant*

Arguments

- **eengZeroTruncationLeading**

  The numerical value for this constant is 1.
- **eengZeroTruncationNone**

  The numerical value for this constant is 2.
- **eengZeroTruncationTrailing**

  The numerical value for this constant is 0.

Examples

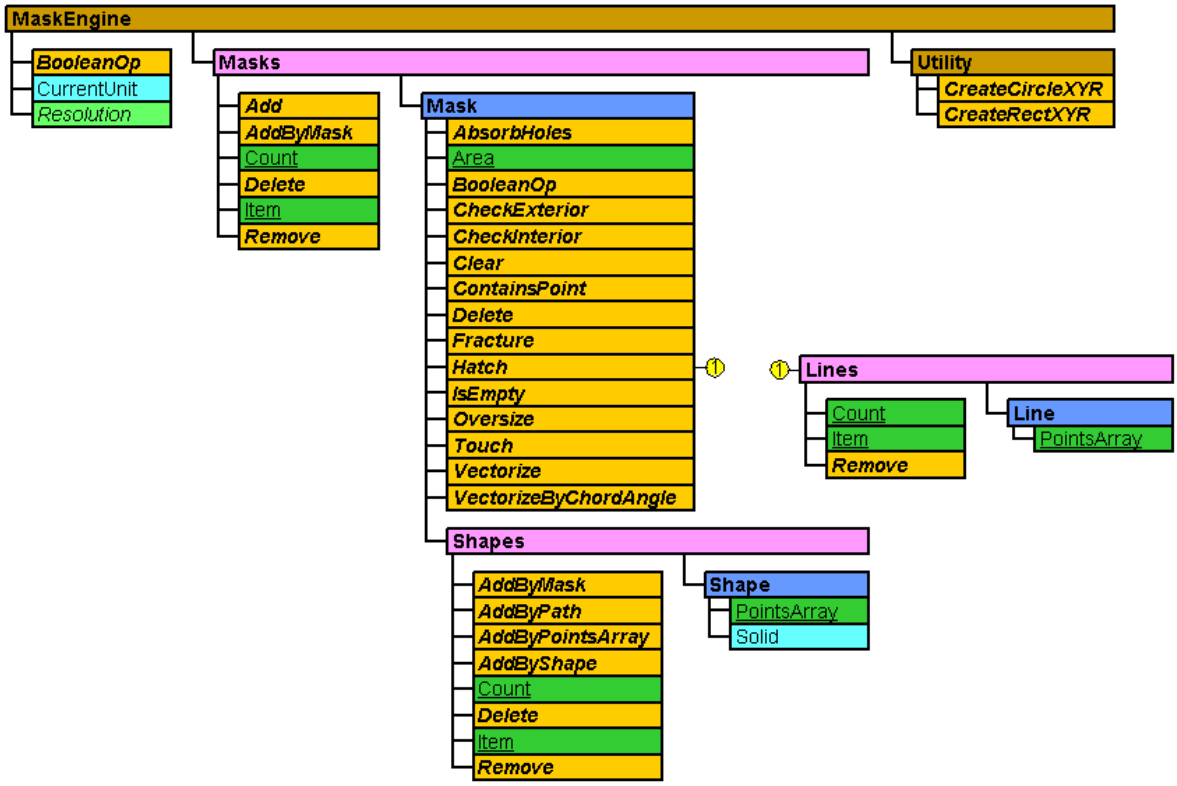
```
‘ Enum EEnginesZeroTruncation

eengZeroTruncationLeading = 1
eengZeroTruncationNone = 2
eengZeroTruncationTrailing = 0
```

Chapter 5
Mask Engine

The Mask Engine Automation Controller (MEAC) provides a simple interface to allow operations to be performed on geometric shapes – operations such as OR, AND, Oversize and area calculations.

Figure 16. Mask Engine Data Model



Mask Engine Overview
Mask Engine Objects
Mask Engine Collections
Mask Engine Enumerated Types

Mask Engine Overview

Before using the Mask Engine Automation Controller, you should become familiar with some of the MEAC concepts.

Point Arrays
How to Connect to the MEAC
How to Compute Placement Density and Board Area
Mask Engine Scripting Samples

Point Arrays

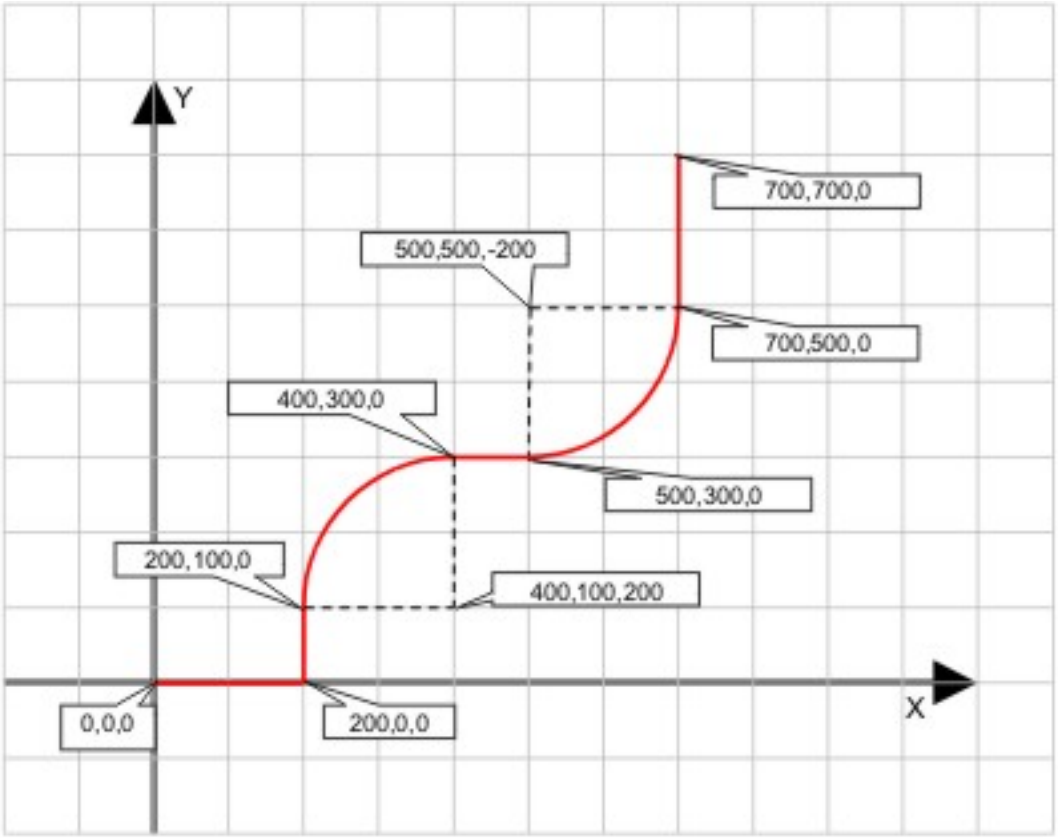
The geometric shapes that the Mask Engine evaluates are defined by point arrays.

Each entry in the point array has three values that define the X, Y and R values of a point. X and Y are the coordinates of the point and R is the radius if an arc is defined.

Figure 17: Points Array shows the conventions for defining point arrays:

- **Non-arcs** — points are defined with an X, Y coordinate, the R value is always 0.
- **Arcs** — defined with 3 points:
  - The first point is the start point of the arc (R value is 0).
  - The second point is the center point of the arc. The R value for this point is equal to the radius of the arc. If the R value is negative, a counter clockwise arc is defined. If the R value is positive, a clockwise arc is defined.
  - The third point is the end point of the arc (R value is 0).
- **Circles** — defined with three points, a point on the circumference (R=0), the circle center (R=radius) and finally back to the starting point (R=0) on the circumference. The Utility.CreateCircleXYR on page 616 method constructs the points array for a circle.
- **Rectangles** — defined using the Utility.CreateRectXYR on page 617 method. It constructs the points array for a rectangle so that it is recognized as a rectangle and not as a regular polygon.

Figure 17. Points Array



How to Connect to the MEAC

The following example shows how to connect to the Mask Engine Automation Controller (MEAC).

```
' get masking engine

Set mskeng = CreateObject("MGCPCBEngines.MaskEngine")
' and add its typelib to give us access to the public constants
Scripting.AddTypeLibrary ("MGCPCBEngines.MaskEngine")
Scripting.AddTypeLibrary ("MGCPCB.Application")

' set the units for the mask engine so that we know the size of the
' incoming data
mskeng.currentUnit = emeUnitMils
```

How to Compute Placement Density and Board Area

You can compute the placement density and area of a board by using the Mask Engine.

The following sample creates a mask and shape to contain the geometries (PointsArrays) of the placement outlines. The area of the placement outline mask is then computed. Similarly, a mask and shape is created to contain the geometry (PointsArray) of the board outline. The area of the board outline mask is computed. Finally, the placement density and board areas are output using the placement area and the board outline area values.

```
Public Sub GetArea(lay)


 ' for this layer of the board
  Dim lyr 'As LayerObject
  Dim layCnt
  layCnt = doc.LayerCount
  'lyr = 1
  Dim i
  If lay = "Top" Then lay = 1
  If lay = "Bottom" Then lay = layCnt


 ' create a mask to hold all placement data
 Set maskCu = mskeng.Masks.Add
 ' get collection of shapes on that mask, this will initially be empty
 Set shapesCu = maskCu.shapes

  If lay = "All" Then
   Set clgs = doc.PlacementOutlines(epcbSelectSelected)
  Else
   Set clgs = doc.PlacementOutlines(epcbSelectSelected, lay)
  End If

  For Each clg In clgs
   'Debug.Print clg.Name
   ‘ Get the geomerty/pointsArray of the placement outline
   pnts = clg.Geometry.PointsArray
   ‘ Add them to the mask
   Call shapesCu.AddByPointsArray(1 + UBound(pnts, 2), pnts)
  Next
  ‘ Calculate the area of the Placment outlines mask
  areaCu = maskCu.Area(emeUnitInch)
  sel = areaCu
End Sub
```

The next sample uses the result of getting the board area above to calculate the placement.

```
Sub boa()
 ' get the area of placement on this board layer in square inches
 ' get board outline and find its area
 Set bo = doc.BoardOutline
 ‘ get the geometry/PointsArray of the board outline
 pnts = bo.Geometry.PointsArray
 Set maskBO = mskeng.Masks.Add
 ‘ Add the BO PointsArray to the mask
```

```
 Call maskBO.shapes.AddByPointsArray(1 + UBound(pnts, 2), pnts)
 ‘ Calculate the area of the BO mask
 areaBO = maskBO.Area(emeUnitInch)
 ' compute the copper density on this layer as a percentage
 MsgBox "Placement Density for layer " & i & " = " & _
  FormatNumber(areaCu * 100 / areaBO, 1) & "%"
 MsgBox ("Board Placement Area " & areaBO)
 'Next
 sel = areaBO
End Sub
```

Mask Engine Scripting Samples

Several mask engine scripting samples are available in *standard\examples\pcb\Automation \AutoProEngines*. The samples use MGCPCBEngines Automation, which requires an Automation Pro license.

The *AutoProEngines* directory contains the following samples, which use VBScript:

ComputeArea.vbs	The examples included in Mask Engine Overview are located in this file.
CopperDensity.vbs	Compute the copper density for layer one of the PCB using the MaskEnginesAutomation controller.
HatchMask.vbs	Masks a negative trace mask for layer one and then hatches it and places the data on a user layer using the MaskEnginesAutomation controller.
mergePlaneShapes.vbs	Merge overlapping plane shapes and update them in Layout.
traceMask.vbs	Make a negative mask of all traces on layer one of the board.

Mask Engine Objects

The Mask Engine Automation Controller supports several categories of Mask Engine objects.

Line Object
Mask Object
MaskEngine Object
Shape Object
Utility Object

Line Object

The Line object provides access to Line properties. The following section describes the properties available in the Line object.

Table 42. Line Object Properties

Property	Description
PointsArray Property (Line Object)	Returns data points for this line.

PointsArray Property (Line Object)

Object: Line Object

Access: Read-Only

Prerequisites: None

Returns data points for this line.

Usage

```
Line.PointsArray ([ByVal eUnit As EMaskEngineUnit = emeUnitCurrent])
```

Arguments

- eUnit (optional)

  The units that will be used for the points array. (EMaskEngineUnit on page 645) Default is current units. Returns the data points four values, x1,y1 to x2,y2.

Return Values

PointsArray. Returns the data points for this line. This is returned in an array of four values, x1,y1 to x2,y2.

Mask Object

The Mask object provides access to Mask methods and properties. The following section describes the methods and properties available in the Mask object.

Table 43. Mask Object Methods and Properties

Method or Property	Description
AbsorbHoles Method (Mask Object)	Absorbs the holes into the shapes. Any holes in the shape are absorbed into the shape's outline so as make the shape defined by a single contour.
Area Property (Mask Object)	Returns the area of the mask.
BooleanOp Method (Mask Object)	Performs the specified boolean operation on this mask and input mask, placing the result in this mask. The result of this Boolean operation may be an empty mask. The difference between this method and the BooleanOp method in the MaskEngine interface is that this modifies the contents of the current mask.
CheckExterior Method (Mask Object)	Checks whether the distance between two masks is less than the specified distance.
CheckInterior Method (Mask Object)	Checks whether the distance between opposite edges of the mask is less than the specified distance.
Clear Method (Mask Object)	Clears the contents of this mask.
ContainsPoint Method (Mask Object)	Checks whether the mask contains the specified point.
Delete Method (Mask Object)	Deletes this mask.
Fracture Method (Mask Object)	Fractures this mask into a number of pieces, each with less than the specified number of points. The act of fracturing makes an implicit call to AbsorbHoles().
Hatch Method (Mask Object)	Hatch this mask with lines. The hatch lines are returned in an output mask
IsEmpty Method (Mask Object)	Checks whether the mask contains any shapes.
Oversize Method (Mask Object)	Oversizes this mask by the specified amount. If the dAmount value is positive, the mask will be made larger and the holes will be made smaller. If the dAmount value is negative, the converse will occur. When oversizing, the corners may be rounded or squared off. The default oversize type is rounded. Square corners are more efficient.
Shapes Property (Mask Object)	Gets the collection of shapes in the mask.
Touch Method (Mask Object)	Checks whether this mask and a another specified mask touch or intersect.
Vectorize Method (Mask Object)	Converts the curved segments into multiple points, specifying the maximum permissible deviation from the true arc. Converts the curved segments of the mask into multiple line segments. Sufficient segments are generated such that the deviation between the line segment and the true-arced segment is less that the specified maximum deviation.
VectorizeByChordAngle Method (Mask Object)	Converts the curved segments into multiple points by specifying the chord angle (in degrees) to use between each point.

AbsorbHoles Method (Mask Object)

Object: Mask Object

Prerequisites: None

Absorbs the holes into the shapes. Any holes in the shape are absorbed into the shape's outline so as make the shape defined by a single contour.

Usage

```
Mask.AbsorbHoles()
```

Arguments

None

Area Property (Mask Object)

Object: Mask Object

Access: Read-Only

Prerequisites: None

Returns the area of the mask.

Usage

```
Mask.Area = Double
```

Arguments

None

Return Values

Double. A double that returns the area of Mask in the requested units. If the mask is empty, the returned area will be zero.

BooleanOp Method (Mask Object)

Object: Mask Object

Prerequisites: None

Performs the specified boolean operation on this mask and input mask, placing the result in this mask. The result of this Boolean operation may be an empty mask. The difference between this method and the BooleanOp method in the MaskEngine interface is that this modifies the contents of the current mask.

Usage

```
Mask.BooleanOp(ByVal eBoolOp As EMaskEngineBooleanOp,
        ByVal pMask As IMaskEngineMask)
```

Arguments

- eBoolOp

  Boolean operation to be performed on the selected mask and input mask. (EMaskEngineBooleanOp on page 640)

- pMask

  Input Mask Object to perform the Boolean operation on.

CheckExterior Method (Mask Object)

Object: Mask Object

Prerequisites: None

Checks whether the distance between two masks is less than the specified distance.

Usage

```
Mask.CheckExterior(ByVal pMask As IMaskEngineMask, ByVal dSpacing As Double, [ByVal eUnit As
    EMaskEngineUnit = emeUnitCurrent]) As IMaskEngineMask
```

Arguments

- pMask

  Input mask to check against the current mask

- dSpacing

  A double that represents the spacing distance to use to check the spacing between the two masks.

- eUnit (Optional)

  The units that will be used for the spacing. (EMaskEngineUnit on page 645) Default is current units.

Return Values

IMaskEngineMask. If a violation exists, the resulting Mask Object will hold the overlapping region. If no violation exists, the resulting mask will be empty. Conceptually, the two masks are oversized by one half of the spacing distance and then ANDed together, the result is placed in the return mask. If no violations exist, the return mask will be empty.

CheckInterior Method (Mask Object)

Object: Mask Object

Prerequisites: None

Checks whether the distance between opposite edges of the mask is less than the specified distance.

Usage

```
Mask.CheckInterior(ByVal dSpacing As Double, [ByVal eUnit As EMaskEngineUnit = emeUnitCurrent])
        As IMaskEngineMask
```

Arguments

- dSpacing

  A double that represents the spacing distance to use to check that the masks are closer than this distance.

- eUnit (Optional)

  The units that will be used for the spacing. (EMaskEngineUnit on page 645) Default is current units.

Return Values

IMaskEngineMask. If a violation exists, the resulting Mask Object will hold the overlapping region. If no violation exists, the resulting mask will be empty. Conceptually, the mask is shrunk by one half of the spacing and this is then oversized by one half of the spacing and subtracted from the original mask. The result is placed in the return mask. If no violations exist, the return mask will be empty.

Clear Method (Mask Object)

Object: Mask Object

Prerequisites: None

Clears the contents of this mask.

Usage

```
Mask.Clear()
```

Arguments

None

ContainsPoint Method (Mask Object)

Object: Mask Object

Prerequisites: None

Checks whether the mask contains the specified point.

Usage

```
Mask.ContainsPoint(ByVal dX As Double, ByVal dY As Double, [ByVal eUnit As EMaskEngineUnit =
    emeUnitCurrent]) As Boolean
```

Arguments

- dX

  The X-Coordinate for the point to check mask against
- dY

  The Y-Coordinate for the point to check mask against
- eUnit (Optional)

  The units that will be used for the input points. (EMaskEngineUnit on page 645) Default is current units.

Return Values

Boolean. If True the mask contains this point. If False, the mask does not contain this point.

Delete Method (Mask Object)

Object: Mask Object

Prerequisites: None

Deletes this mask.

Usage

```
Mask.Delete()
```

Arguments

None

Fracture Method (Mask Object)

Object: Mask Object

Prerequisites: None

Fractures this mask into a number of pieces, each with less than the specified number of points. The act of fracturing makes an implicit call to AbsorbHoles().

Usage

```
Mask.Fracture(ByVal nNumPointsMax As Long)
```

Arguments

- nNumPointsMax

  Use no more than this number of points in the resulting Fracture.

Hatch Method (Mask Object)

Object: Mask Object

Prerequisites: None

Hatch this mask with lines. The hatch lines are returned in an output mask

Usage

```
Mask.Hatch(ByVal eAngle As EMaskEngineHatchAngle, ByVal dSpacing As Double, ByVal dLineWidth
        As Double, [ByVal eUnit As EMaskEngineUnit = emeUnitCurrent]) As IMaskEngineLines
```

Arguments

- eAngle

  The angle of the hatch lines. (EMaskEngineHatchAngle on page 642)
- dSpacing

  The spacing between the hatch lines.
- dLineWidth

  Specifies the line width of the hatch lines.
- eUnit (Optional)

  The units that will be used for the spacing and line width. (EMaskEngineUnit on page 645)
  Default is current units.

Return Values

IMaskEngineLines. Returns the Lines Collection - if the mask is empty, an empty collection of lines will be returned.

IsEmpty Method (Mask Object)

Object: Mask Object

Prerequisites: None

Checks whether the mask contains any shapes.

Usage

```
Mask.IsEmpty() As Boolean
```

Arguments

None

Return Values

Boolean. If True the mask is empty. If False the mask in not empty.

Oversize Method (Mask Object)

Object: Mask Object

Prerequisites: None

Oversizes this mask by the specified amount. If the dAmount value is positive, the mask will be made larger and the holes will be made smaller. If the dAmount value is negative, the converse will occur. When oversizing, the corners may be rounded or squared off. The default oversize type is rounded. Square corners are more efficient.

Usage

```
Mask.Oversize(ByVal dAmount As Double,
        [ByVal eType As EMaskEngineOversizeType = emeOversizeTypeRound],
        [ByVal eUnit As EMaskEngineUnit = emeUnitCurrent])
```

Arguments

- dAmount

  A double which specifies the amount to oversize the mask by.

- eType (Optional)

  Specifies the type of corners to use when over-sizing. (EMaskEngineOversizeType on page 643) Default oversize type is rounded.

- eUnit (Optional)

  The units that will be used for the for the dAmount parameter. (EMaskEngineUnit on page 645) Default is current units.

Shapes Property (Mask Object)

Object: Mask Object

Access: Read-Only

Prerequisites: None

Gets the collection of shapes in the mask.

Usage

```
Mask.Shapes = IMaskEngineShapes
```

Arguments

None

Return Values

IMaskEngineShapes. Returns the collection of shapes associated with this mask. (Shapes Collection) If there are no shapes in this mask (empty mask), returns an empty collection.

Touch Method (Mask Object)

Object: Mask Object

Prerequisites: None

Checks whether this mask and a another specified mask touch or intersect.

Usage

```
Mask.Touch(ByVal pMask As IMaskEngineMask) As Boolean
```

Arguments

- pMask

  Input mask to check against this mask.

Return Values

True | False. If True the masks touch or intersect. If False, the masks do not touch or intersect.

Vectorize Method (Mask Object)

Object: Mask Object

Prerequisites: None

Converts the curved segments into multiple points, specifying the maximum permissible deviation from the true arc. Converts the curved segments of the mask into multiple line segments. Sufficient segments are generated such that the deviation between the line segment and the true-arced segment is less that the specified maximum deviation.

Usage

```
Mask.Vectorize(ByVal dMaxDeviation As Double, [ByVal eUnit As EMaskEngineUnit = emeUnitCurrent])
```

Arguments

- dMaxDeviation

  A double that specifies the maximum permissible deviation from the true arc.

- eUnit (Optional)

  The units that will be used for the dMaxDeviation parameter. (EMaskEngineUnit on page 645) Default is current units.

VectorizeByChordAngle Method (Mask Object)

Object: Mask Object

Prerequisites: None

Converts the curved segments into multiple points by specifying the chord angle (in degrees) to use between each point.

Usage

```
Mask.VectorizeByChordAngle(ByVal dChordAngle As Double)
```

Arguments

- dChordAngle

  A double that chord angle (in degrees) to use.

MaskEngine Object

The MaskEngine object provides access to MaskEngine methods and properties. The following section describes the methods and properties available in the MaskEngine object.

Table 44. MaskEngine Object Methods and Properties

Method or Property	Description
BooleanOp Method (MaskEngine Object)	Performs the specified boolean operation on the two input masks and places the result in the output mask.
CurrentUnit Property (MaskEngine Object)	Sets or gets the current coordinate units.
Masks Property (MaskEngine Object)	Gets the collection of masks owned by the engine.
Resolution Property (MaskEngine Object)	Defines the resolution (base units) of the coordinate data.
Utility Property (MaskEngine Object)	Gets the Utility object.

BooleanOp Method (MaskEngine Object)

Object: MaskEngine Object

Prerequisites: None

Performs the specified boolean operation on the two input masks and places the result in the output mask.

Usage

```
MaskEngine.BooleanOp(ByVal eBoolOp As EMaskEngineBooleanOp,  ByVal pMask1 As
        IMaskEngineMask, ByVal pMask2 As IMaskEngineMask) As IMaskEngineMask
```

Arguments

- eBoolOp

  The boolean operation to perform. (EMaskEngineBooleanOp on page 640)
- pMask1

  First input Mask Object.
- pMask2

  Second input Mask Object.

Return Values

IMaskEngineMask. Returns a Mask Object.

CurrentUnit Property (MaskEngine Object)

Object: MaskEngine Object

Access: Read/Write

Prerequisites: None

Sets or gets the current coordinate units.

Usage

```
MaskEngine.CurrentUnit = EMaskEngineUnit
```

Arguments

None

Return Values

EMaskEngineUnit. on page 645 The unit to use for the coordinates associated with the masks. This allows the engine to scale the incoming data.

Masks Property (MaskEngine Object)

Object: MaskEngine Object

Access: Read-Only

Prerequisites: None

Gets the collection of masks owned by the engine.

Usage

```
MaskEngine.Masks = IMaskEngineMasks
```

Arguments

None

Return Values

IMaskEngineMasks. Returns the Masks Collection currently owned by the engine. Returns an empty collection if no masks are currently defined.

Resolution Property (MaskEngine Object)

Object: MaskEngine Object

Access: Write-Only

Prerequisites: None

Defines the resolution (base units) of the coordinate data.

Usage

```
MaskEngine.Resolution = EMaskEngineResolutionUnit
```

Arguments

None

Return Values

EMaskEngineResolutionUnit. on page 644 Set the minimum resolution of the incoming coordinate data (base units). By default the resolution is 1nm. This is used to define the minimum physical distance between two points – used when rounding data.

Utility Property (MaskEngine Object)

Object: MaskEngine Object

Access: Read-Only

Prerequisites: None

Gets the Utility object.

Usage

```
MaskEngine.Utility = IMaskEngineUtility
```

Arguments

None

Return Values

IMaskEngineUtility. Returns the Utility Object.

Shape Object

The Shape object provides access to Shape properties. The following section describes the properties available in the Shape object.

Table 45. Shape Object Properties

Property	Description
PointsArray Property (Shape Object)	Returns the PointsArray of this shape.
Solid Property (Shape Object)	Sets or returns whether a shape is a solid or a hole.

PointsArray Property (Shape Object)

Object: Shape Object

Access: Read-Only

Prerequisites: None

Returns the PointsArray of this shape.

Usage

```
Shape.PointsArray ([ByVal eUnit As EMaskEngineUnit = emeUnitCurrent])
```

Arguments

- EUnit

  (Optional) The units that will be used for the points array. (EmaskEngineUnit on page 645)
  Default is current units.

Return Values

PointsArray. The PointsArray for this shape. It should be noted that the returned points array is an XYR array in the same format as is passed into and returned from Layout. Any holes intersecting the edge of the shape will be merged with the shape's outline.

Solid Property (Shape Object)

Object: Shape Object

Access: Read/Write

Prerequisites: None

Sets or returns whether a shape is a solid or a hole.

Usage

```
Shape.Solid = True | False
```

Arguments

None

Return Values

True | False. If True, this shape is solid. If False, this shape is a hole.

Utility Object

The Utility object provides access to Utility methods. The following section describes the methods available in the Utility object.

Table 46. Utility Object Methods

Method	Description
CreateCircleXYR Method (Utility Object)	Creates an XYR array for a circle.
CreateRectXYR Method (Utility Object)	Creates an XYR array for a rectangle.

CreateCircleXYR Method (Utility Object)

Object: Utility Object

Prerequisites: None

Creates an XYR array for a circle.

Usage

```
Utility.CreateCircleXYR(ByVal dCenterX As Double, ByVal dCenterY As Double,  ByVal dRadius As
        Double)
```

Arguments

- dCenterX

  A double containing the X-coordinate of the circle.
- dCenterY

  A double containing the Y-coordinate of the circle.
- dRadius

  A double containing the radius of the circle.

Return Values

Array. A Variant that contains an array. This array is structured as a points array that can be used as input when constructing a shape to add to a mask.

CreateRectXYR Method (Utility Object)

Object: Utility Object

Prerequisites: None

Creates an XYR array for a rectangle.

Usage

```
Utility.CreateRectXYR(ByVal dMinX As Double, ByVal dMinY As Double,
        ByVal dMaxX As Double, ByVal dMaxY As Double)
```

Arguments

- dMinX

  A double containing the X-coordinate of the lower left corner.
- dMinY

  A double containing the Y-coordinate of the lower left corner.
- dMaxX

  A double containing the X-coordinate of the upper right corner.
- dMaxY

  A double containing the Y-coordinate of the upper right corner.

Return Values

Array. A Variant that contains an array. This array is structured as a points array that can be used as input when constructing a shape to add to a mask.

Mask Engine Collections

The Mask Engine Automation Controller supports several categories of Mask Engine collections.

Lines Collection
Masks Collection
Shapes Collection

Lines Collection

The Lines collection is a collection of Line objects. The following section defines the methods and properties of the Lines collection.

Table 47. Lines Collection Methods and Properties

Method or Property	Description
Count Property (Lines Collection)	Returns the number of Line objects contained in the Lines collection.
Item Property (Lines Collection)	Returns a Line object contained in the collection.
Remove Method (Lines Collection)	Removes a Line object from the Lines collection.

Count Property (Lines Collection)

Object: Lines Collection

Access: Read-Only

Prerequisites: None

Returns the number of Line objects contained in the Lines collection.

Usage

```
Lines.Count = Long
```

Arguments

None

Return Values

Long. The number of Line Objects the collection contains.

Item Property (Lines Collection)

Object: Lines Collection

Access: Read-Only

Prerequisites: None

Returns a Line object contained in the collection.

Usage

```
Lines.Item (ByVal vIndex As Variant) = Line
```

Arguments

- vIndex

  Numerical index of the Line object to retrieve from the collection.

Return Values

Line. Returns a Line Object.

Remove Method (Lines Collection)

Object: Lines Collection

Prerequisites: None

Removes a Line object from the Lines collection.

Usage

```
Lines.Remove(ByVal vIndex As Variant)
```

Arguments

- vIndex

  vIndex. The index of the Line Object to remove from the collection.

Masks Collection

The Masks collection is a collection of Mask objects. The following section defines the methods and properties of the Masks collection.

Table 48. Masks Collection Methods and Properties

Method or Property	Description
Add Method (Masks Collection)	Adds a Mask object to the Masks collection.
AddByMask Method (Masks Collection)	Adds a mask to this collection by copying an existing one.
Count Property (Masks Collection)	Returns the number of Mask objects contained in the Masks collection.
Delete Method (Masks Collection)	Deletes all masks from the collection.
Item Property (Masks Collection)	Returns a Mask object contained in the collection.
Remove Method (Masks Collection)	Removes a Mask object from the Masks collection.

Add Method (Masks Collection)

Object: Masks Collection

Prerequisites: None

Adds a Mask object to the Masks collection.

Usage

```
Masks.Add() As IMaskEngineMask
```

Arguments

- IMaskEngineMask

  The Mask Object to add to the collection.

AddByMask Method (Masks Collection)

Object: Masks Collection

Prerequisites: None

Adds a mask to this collection by copying an existing one.

Usage

```
Masks.AddByMask(ByVal pMaskIn As IMaskEngineMask) As IMaskEngineMask
```

Arguments

- pMaskIn

  Mask to copy to this collection.

Return Values

IMaskEngineMask. Returns a Mask Object identical in content to the input mask.

Count Property (Masks Collection)

Object: Masks Collection

Access: Read-Only

Prerequisites: None

Returns the number of Mask objects contained in the Masks collection.

Usage

```
Masks.Count = Long
```

Arguments

None

Return Values

Long. The number of Mask Objects the collection contains.

Delete Method (Masks Collection)

Object: Masks Collection

Prerequisites: None

Deletes all masks from the collection.

Usage

Masks.Delete ()

Arguments

None

Item Property (Masks Collection)

Object: Masks Collection

Access: Read-Only

Prerequisites: None

Returns a Mask object contained in the collection.

Usage

```
Masks.Item (ByVal vIndex As Variant) = Mask
```

Arguments

- vIndex

  Numerical index of the Mask Object to retrieve from the collection.

Return Values

Mask. Returns a Mask Object.

Remove Method (Masks Collection)

Object: Masks Collection

Prerequisites: None

Removes a Mask object from the Masks collection.

Usage

```
Masks.Remove(ByVal vIndex As Variant)
```

Arguments

- vIndex

  The index of the Mask Object to remove from the collection.

Shapes Collection

The Shapes collection is a collection of Shape objects. The following section defines the methods and properties of the Shapes collection.

Table 49. Shapes Collection Methods and Properties

Method or Property	Description
AddByMask Method (Shapes Collection)	Copies (or moves) all the shapes from an existing mask into the shapes collection for this mask. Note that no shapes collection is returned.
AddByPath Method (Shapes Collection)	Adds a shape defined by a path with width to the collection of shapes for this mask.
AddByPointsArray Method (Shapes Collection)	Adds a shape defined by an XYR array to the shapes collection.
AddByShape Method (Shapes Collection)	Copies a shape defined by an existing shape to the shapes collection.
Count Property (Shapes Collection)	Returns the number of Shape objects contained in the Shapes collection.
Delete Method (Shapes Collection)	Deletes all shapes from the mask.
Item Property (Shapes Collection)	Returns a Shape object contained in the collection.
Remove Method (Shapes Collection)	Removes a Shape object from the Shapes collection.

AddByMask Method (Shapes Collection)

Object: Shapes Collection

Prerequisites: None

Copies (or moves) all the shapes from an existing mask into the shapes collection for this mask. Note that no shapes collection is returned.

Usage

```
Shapes.AddByMask(ByVal pMaskIn As IMaskEngineMask,
        [ByVal bCopy As Boolean = True])
```

Arguments

- pMaskIn

  Input Mask Object to use for the copy or move. (IMaskEngineMask)

- bCopy

  (Optional) If True, copy all the shapes from the input mask to the Shapes Collection for this mask. If False, move all the shapes from the input mask to the Shapes Collection for this mask.

AddByPath Method (Shapes Collection)

Object: Shapes Collection

Prerequisites: None

Adds a shape defined by a path with width to the collection of shapes for this mask.

Usage

```
Shapes.AddByPath(ByVal nPnts As Long, pSafeArrayOfPoints As Variant,
        ByVal dWidth As Double, [ByVal bIsSolid As Boolean = True],
        [ByVal eType As EMaskEngineOversizeType = emeOversizeTypeRound],
        [ByVal eUnit As EMaskEngineUnit = emeUnitCurrent])
```

Arguments

- nPnts

  Number of points in the array.

- pSafeArrayOfPoints

  XYR array of points of points that defines the path.

- dWidth

  Width of the defined path.

- bIsSolid

  (Optional) True if this path is a solid shape. False if this shape is a hole. Default is true.

- eType

  (Optional) Specifies the type of corners to use when over-sizing. (EMaskEngineOversizeType on page 643) Default oversize type is rounded.

- eUnit

  (Optional) The units that will be used for the points array. (EMaskEngineUnit on page 645) Default is current units.

AddByPointsArray Method (Shapes Collection)

Object: Shapes Collection

Prerequisites: None

Adds a shape defined by an XYR array to the shapes collection.

Usage

```
Shapes.AddByPointsArray(ByVal nPnts As Long, pSafeArrayOfPoints As Variant,
        [ByVal bIsSolid As Boolean = True],
        [ByVal eUnit As EMaskEngineUnit = emeUnitCurrent]) As IMaskEngineShape
```

Arguments

- nPnts

  Number of points in the array.
- pSafeArrayOfPoints

  XYR array of points of points that defines the shape to be added.
- bIsSolid

  (Optional) True if this is a solid shape. False if this shape is a hole. Default is true.
- eUnit

  (Optional) The units that will be used for the points array. (EMaskEngineUnit on page 645) Default is current units.

Return Values

IMaskEngineShape. Returns a Shape Object.

AddByShape Method (Shapes Collection)

Object: Shapes Collection

Prerequisites: None

Copies a shape defined by an existing shape to the shapes collection.

Usage

```
Shapes.AddByShape(ByVal
        pShape As IMaskEngineShape,
        [ByVal bIsSolid As Boolean = True]) As IMaskEngineShape
```

Arguments

- pShape

  Copy this input Shape Object to the Shapes collection.
- bIsSolid

  (Optional) True if this is a solid shape. False if this shape is a hole. Default is true.

Return Values

IMaskEngineShape. Returns a Shape Object.

Count Property (Shapes Collection)

Object: Shapes Collection

Access: Read-Only

Prerequisites: None

Returns the number of Shape objects contained in the Shapes collection.

Usage

```
Shapes.Count = Long
```

Arguments

None

Return Values

Long. The number of Shape Objects the collection contains.

Delete Method (Shapes Collection)

Object: Shapes Collection

Prerequisites: None

Deletes all shapes from the mask.

Usage

```
Shapes.Delete ()
```

Arguments

None

Item Property (Shapes Collection)

Object: Shapes Collection

Access: Read-Only

Prerequisites: None

Returns a Shape object contained in the collection.

Usage

```
Shapes.Item (ByVal vIndex As Variant) = Shape
```

Arguments

- vIndex

  Numerical index of the Shape Object to retrieve from the collection.

Return Values

Shape. Returns a Shape Object.

Remove Method (Shapes Collection)

Object: Shapes Collection

Prerequisites: None

Removes a Shape object from the Shapes collection.

Usage

```
Shapes.Remove(ByVal vIndex As Variant)
```

Arguments

- vIndex

  The index of the Shape Object to remove from the collection.

Mask Engine Enumerated Types

The Mask Engine Automation Controller supports a variety of Mask Engine enumerated types.

Table 50. Mask Engine Enumerated Types

Enumerated Type	Description
EMaskEngineBooleanOp Enum	Use this enumerator to specify boolean operations in Mask or MaskEngine objects, using the BooleanOp method.
EMaskEngineErrCode Enum	Mask Engine Error Codes.
EMaskEngineHatchAngle Enum	Use this enumerator to specify the hatch angle, using the Hatch method of the Mask object.
EMaskEngineOversizeType Enum	Use this enumerator to specify the type of corners to use when a mask or shape is oversized.
EMaskEngineResolutionUnit Enum	Mask Engine Resolution Constants; defines the resolution (base units) of the system providing the input data.
EMaskEngineUnit Enum	Mask Engine Unit Constants.

EMaskEngineBooleanOp Enum

Prerequisites: None

Use this enumerator to specify boolean operations in Mask or MaskEngine objects, using the BooleanOp method.

Usage

EMaskEngineBooleanOp.*Constant*

Arguments

- **emeBooleanOpAND**

  The numerical value for this constant is 1.

  AND operation used with MaskEngine.BooleanOp. on page 607
- **emeBooleanOpOR**

  The numerical value for this constant is 2.

  OR operation used with MaskEngine.BooleanOp. on page 607
- **emeBooleanOpSubtract**

  The numerical value for this constant is 0.

  SUBTRACT operation used with MaskEngine.BooleanOp. on page 607
- **emeBooleanOpXOR**

  The numerical value for this constant is 3.

  XOR operation used with MaskEngine.BooleanOp. on page 607

EMaskEngineErrCode Enum

Prerequisites: None

Mask Engine Error Codes.

Usage

EMaskEngineErrCode.*Constant*

Arguments

- **emeErrCodeAutomationDisabled**

  The numerical value for this constant is -2147220984 (&H80040208).
- **emeErrCodeBadObject**

  The numerical value for this constant is -2147220991 (&H80040201).

  Object does not exist or is not available.
- **emeErrCodeInvalidEnumValue**

  The numerical value for this constant is -2147220989 (&H80040203).

  Enumerated value does not match a known value.
- **emeErrCodeInvalidParameter**

  The numerical value for this constant is -2147220992 (&H80040200).

  Parameter is not the correct type or doesn't exist.
- **emeErrCodeInvalidPointsArray**

  The numerical value for this constant is -2147220990 (&H80040202).

  The PointsArray was invalid.
- **emeErrCodeNoLicenseAvailable**

  The numerical value for this constant is -2147220987 (&H80040205).

  An Automation Pro license(xeautoextended feature) and/or an Xpedition FabLink 200 license could not be acquired.
- **emeErrCodeNoProductLicenseAvailable**

  The numerical value for this constant is -2147220985 (&H80040207).
- **emeErrCodeUnitsNotSet**

  The numerical value for this constant is -2147220988 (&H80040204).

  The default units are not set.

EMaskEngineHatchAngle Enum

Prerequisites: None

Use this enumerator to specify the hatch angle, using the Hatch method of the Mask object.

Usage

EMaskEngineHatchAngle.*Constant*

Arguments

- **emeHatchAngle_0**

  The numerical value for this constant is 0.
- **emeHatchAngle_135**

  The numerical value for this constant is 3.
- **emeHatchAngle_45**

  The numerical value for this constant is 1.
- **emeHatchAngle_90**

  The numerical value for this constant is 2.

EMaskEngineOversizeType Enum

Prerequisites: None

Use this enumerator to specify the type of corners to use when a mask or shape is oversized.

Usage

EMaskEngineOversizeType.*Constant*

Arguments

- **emeOversizeTypeRound**

  The numerical value for this constant is 0.
- **emeOversizeTypeSquare**

  The numerical value for this constant is 1.

EMaskEngineResolutionUnit Enum

Prerequisites: None

Mask Engine Resolution Constants; defines the resolution (base units) of the system providing the input data.

Usage

EMaskEngineResolutionUnit.*Constant*

Arguments

- **emeResolutionUnit10nm**

  The numerical value for this constant is 1.
- **emeResolutionUnit1nm**

  The numerical value for this constant is 0.

EMaskEngineUnit Enum

Prerequisites: None

Mask Engine Unit Constants.

Usage

EMaskEngineUnit.*Constant*

Arguments

- **emeUnitCurrent**

  The numerical value for this constant is 0.
- **emeUnitDatabase**

  The numerical value for this constant is 1.
- **emeUnitInch**

  The numerical value for this constant is 3.
- **emeUnitMils**

  The numerical value for this constant is 2.
- **emeUnitMM**

  The numerical value for this constant is 4.
- **emeUnitUM**

  The numerical value for this constant is 5.

Third-Party Information

Details on open source and third-party software that may be included with this product are available in the *<your_software_installation_location>/legal* directory.