A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

AddArrowPoint
AddBezierPoint
AddFreehandPoint
AddNode
AddPageFrame
AddTabStop
AfterObject
AlignObjects
AlignTextToBaseline
AlignToCenterOfPage
AlignToGrid
AppendCurveLine
AppendObjectToSelection
ApplyBitmapEffect
ApplyBlend
ApplyContour
ApplyDistortion
ApplyDropShadow
ApplyEnvelopeFrom
ApplyExtrude

ApplyFountainFill
ApplyFullColorFill
ApplyLensEffect
ApplyNoFill
ApplyOutline
ApplyPerspectiveEffect
ApplyPostscriptFill
ApplyPreset
ApplyPresetEnvelope
ApplyRotatedExtrude
ApplyStyle
ApplyTextureFill
ApplyTwoColorFill
ApplyUniformFillColor

B

BeforeObject
BeginCommandGroup
BeginDrawArrow
BeginDrawBezier
BeginDrawCurve
BeginDrawFreehand
BeginEditObject
BreakApart

C

ChangeLayerName
ChangeLayerColor
ClearEffect
ClearNodeSelection
ClickedDialogButton
CloneObject
CloseCurve
Combine
ConvertBitmapTo
ConvertColor
ConvertOutlineToObject
ConvertToBitmap
ConvertToCurves
CopyEffectFrom
CopyPowerClip
CopyPropertiesFrom
CopyToClipboard
CopyToLayer
CreateAngleDimension
CreateArtisticText
CreateCallout
CreateConnector
CreateDimension
CreateEllipse
CreateGridBoxes

D

E

F

G

N

O

P

Q

R

S

ZoomToHeight
ZoomToPage
ZoomToRectangle
ZoomToSelection
ZoomToWidth

File commands

FileClose (DRAW)

ReturnValue = .FileClose(.PromptUser = *boolean*)

This command closes the current drawing.

Return Value

Returns one of the following values:

- TRUE (-1)－the file was closed
- FALSE (0)－the file was not closed

Parameter	Description
.PromptUser	Set to TRUE (-1) to prompt the user before closing the file. Set to FALSE (0) to close the file without prompting the user.

Note

- This command must be preceded by the .FileSave command or changes will be lost.

Example

```
.FileClose TRUE
```

The above example prompts the user before closing the active CorelDRAW document.

```
.FileClose
```

The above example closes the active CorelDRAW document without prompting the user.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

FileExit (DRAW)

ReturnValue = .FileExit .PromptUser = *boolean*

This command exits the application.

Return Value

Returns one of the following values:

- TRUE (-1)－the application was closed
- FALSE (0)－the application was not closed.

Parameter	Description
.PromptUser	Set to TRUE (-1) to prompt the user before closing the application. Set to FALSE (0) to exit the application without prompting the user.

Note

- This command must be preceded by the .FileSave command or changes will be lost.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

FileExport (DRAW)

.FileExport .FileName = *string*, .FilterID = *long*, .Width = *long*, .Height = *long*, .XResolution = *long*, .YResolution = *long*, .ImageType = *long;* Antialiasing = *long;* Overwrite = *boolean;* SelectionOnly = *boolean*

This command saves the current drawing in a format that other programs can read.

Parameter	Description
.FileName	Lets you specify the name of the file to export.
.FilterID	Lets you specify the type of file filter. 769 = Windows Bitmap (BMP) 770 = Paintbrush (PCX) 771 = Targa Bitmap (TGA) 772 = TIFF Bitmap (TIF) 773 = CompuServe Bitmap (GIF) 774 = JPEG Bitmap (JPG) 776 = Scitex CT Bitmap (SCT) 787 = GEM Paint File (IMG) 788 = Adobe Photoshop (PSD) 791 = MACPaint Bitmap (MAC) 792 = OS/2 Bitmap (BMP) 800 = CALS Compressed Bitmap (CAL) 802 = Portable Network Graphics (PNG) 806 = Kodak FlashPix Image (FPX) 1280 = Computer Graphics Metafile (CGM) 1281 = HPGL Plotter File (PLT) 1282 = Micrografx 2.x, 3.x (DRW) 1284 = GEM File (GEM) 1285 = IBM PIF (PIF) 1287 = WordPerfect Graphics (WPG) 1289 = Encapsulated PostScript (EPS) 1293 = Macintosh Pict (PCT) 1294 = Windows Metafile (WMF) 1296 = AutoCad (DXF) 1298 = Scodl (SCD) 1300 = Enhanced Windows Metafile (EMF) 1302 = True Type Font (TTF) 1303 = Adobe Type 1 Font (PFB) 1305 = Adobe Illustrator (AI) 1312 = Corel Barista (HTM) 1313 = Corel Image Map (HTM) 1329 = Frame Vector Metafile (FMV) 1333 = Adobe Portable Document File (PDF) 1792 = Corel PHOTO-PAINT Image (CPT) Ver 5.0/6.0 1793 = Corel Presentation Exchange 6/7 (CMX) 1794 = Corel Presentation Exchange 5.0 (CMX) 1799 = Corel PHOTO-PAINT Image (CPT) 2048 = ANSI Text (TXT) 2049 = MS Word for Windows 6/7 (DOC) 2050 = MS Word for Windows 2.x (DOC) 2051 = MS Word 4.0, 5.0, 5.5 (DOC) 2052 = MS Word for Macintosh 4.0, 5.0 (DOC) 2053 = Rich Text Format (RTF) 2055 = Corel WordPerfect 6/7/8 (WPD) 2056 = Corel WordPerfect 5.1 (WP5) 2057 = Corel Word Perfect 5.0 (WP5) 2058 = Corel Word Perfect 4.2 (WP5) 2059 = Word Star for Windows 1.x, 2.0 (WSW) 2060 = Word Star 7.0 (WSD) 2061 = Word Star 2000 (WSD) 2062 = XYWrite for Windows (XY) 2068 = MS Word 97 (DOC)
.Width	Lets you specify the width of the image in pixels.
.Height	Lets you specify the height of the image in pixels.
.XResolution	Lets you specify the horizontal resolution of the image in dots per inch (dpi).
.YResolution	Lets you specify the vertical resolution of the image in dots per inch (dpi).

.ImageType	Lets you specify the image type. 1 = Monochrome bitmap 3 = 8-bit paletted color bitmap 4 = 24-bit RGB color bitmap 6 = 32-bit CMYK bitmap 10 = 4-bit, 16 colors (standard VGA palette)
.Antialiasing	0 = None 1 = Normal 2 = Super-Sampling
.Overwrite	When set to TRUE (-1) overwrites the file if one exists.
.SelectionOnly	When set to TRUE (-1) exports only the current selection.

Example

```
.FileExport "C:\TEMP1.BMP", 769, 320, 400, 72, 72, 4
```

The above example exports a CorelDRAW file to a Windows bitmap named "TEMP1.BMP".

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

FileImport (DRAW)

.FileImport .FileName = *string,* .FilterID = *long,* MaintainLayersAndPages = *boolean*

This command brings graphics into CorelDRAW from other programs.

Parameter	Description
.FileName	Lets you specify the name of the file to import.
.FilterID	Lets you specify the type of file filter: 769 = Windows Bitmap (BMP) 770 = Paintbrush (PCX) 771 = Targa Bitmap (TGA) 772 = TIFF Bitmap (TIF) 773 = CompuServe Bitmap (GIF) 774 = JPEG Bitmap (JPG) 775 = Kodak Photo CD Image (PCD) 776 = Scitex CT Bitmap (SCT) 784 = Windows 3.x/NT Cursor resource (CUR) 785 = Windows 3.x/NT Icon resource (ICO) 786 = Windows 3.x/NT Bitmaps resource (EXE) 787 = GEM Paint File (IMG) 788 = Adobe Photoshop (PSD) 789 = Picture Publisher 4 (PP4) 791 = MACPaint Bitmap (MAC) 792 = OS/2 Bitmap (BMP) 793 = Wavelet Compressed Bitmaps (WVL) 800 = CALS Compressed Bitmap (CAL) 802 = Portable Network Graphics (PNG) 803 = Picture Publisher 5.0, 6 (PP5, PP6) 806 = Kodak FlashPix Image (FPX) 1280 = Computer Graphics Metafile (CGM) 1281 = HPGL Plotter File (PLT) 1282 = Micrografx 2.x, 3.x (DRW) 1284 = GEM File (GEM) 1285 = IBM PIF (PIF) 1287 = WordPerfect Graphics (WPG) 1289 = Encapsulated PostScript (EPS) 1290 = PostScript Interpreted (PS, PRN, ES) 1291 = MET Maetfile (MET) 1292 = NAP Metafile (NAP) 1293 = Macintosh Pict (PCT) 1294 = Windows Metafile (WMF) 1296 = AutoCad (DXF) 1298 = Scodl (SCD) 1300 = Enhanced Windows Metafile (EMF) 1302 = True Type Font (TTF) 1303 = Adobe Type 1 Font (PFB) 1305 = Adobe Illustrator (AI) 1312 = Corel Barista (HTM) 1313 = Corel Image Map (HTM) 1314 = Placeable Enhanced PDF (PDF) 1315 = Visio (VSD) 1329 = Frame Vector Metafile (FMV) 1333 = Adobe Portable Document File (PDF) 1334 = Lotus Pic (PIC) 1339 = Micrografx Designer 6.0 (DSF) 1557 = HyperText Markup Language (HTM) 1792 = Corel PHOTO-PAINT Image (CPT) Ver 5.0/6.0 1793 = Corel Presentation Exchange 6/7 (CMX) 1794 = Corel Presentation Exchange 5.0 (CMX) 1796 = CorelDRAW Compressed (CDX) 1797 = Corel CMX Compressed (CPX) 1799 = Corel PHOTO-PAINT Image (CPT) 2048 = ANSI Text (TXT) 2049 = MS Word for Windows 6/7 (DOC) 2050 = MS Word for Windows 2.x (DOC) 2051 = MS Word 4.0, 5.0, 5.5 (DOC) 2052 = MS Word for Macintosh 4.0, 5.0 (DOC)

	2053 = Rich Text Format (RTF) 2055 = Corel WordPerfect 6/7/8 (WPD) 2056 = Corel WordPerfect 5.1 (WP5) 2057 = Corel Word Perfect 5.0 (WP5) 2058 = Corel Word Perfect 4.2 (WP5) 2059 = Word Star for Windows 1.x, 2.0 (WSW) 2060 = Word Star 7.0 (WSD) 2061 = Word Star 2000 (WSD) 2062 = XYWrite for Windows (XY) 2063 = Ami Professional 2.0, 3.0 (SAM) 2068 = MS Word 97 (DOC)
.MaintainLayersAndPages	When set to TRUE (-1) maintain layers and pages during import.

Example

```
.FileNew
.FileImport "C:\TEST1.BMP", 769, FALSE
```

The above example creates a new document and imports a Windows bitmap file named "TEST1.BMP" into the document.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

FileNew (DRAW)

ReturnValue = .FileNew

This command creates a new drawing.

Return Value

Returns one of the following values:

- TRUE (-1) the file was created
- FALSE (0) the file was not created

Note

- You cannot change the active CorelDRAW document in a script except by using the .FileNew or .FileOpen command. Changing the active CorelDRAW document with keyboard and mouse actions does not affect an executing script.

Example

```
.FileNew
```

The above example creates a new CorelDRAW document.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

FileOpen (DRAW)

.FileOpen .FileName = *string*

This command loads a drawing or Styles template into CorelDRAW.

Parameter	Description
.FileName	Lets you specify the name of the file to open.

Note

- You cannot change the active CorelDRAW document in a script except by using the .FileNew or .FileOpen command. Changing the active CorelDRAW document with keyboard and mouse actions does not affect an executing script.

Example

```
.FileOpen "C:\TEST1.CDR"
```

The above example opens a CorelDRAW file named "TEST1.CDR".

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

FilePrint (DRAW)

.FilePrint

This command prints the active document.

Example

```
.FilePrint
```

The above example sends the active document to the printer.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

FileSave (DRAW)

.FileSave .FileName = *string*, .ThumbNailSize = *long*, .SaveSelectedOnly = *boolean*, .FileVersion = *long*, .IncludeCMXData = *boolean*

This command saves the active document.

Parameter	Description
.FileName	Lets you specify the name of the file to save.
.ThumbNailSize	Lets you specify the size of the thumbnail: 0 = None 1 = 1k (mono) 5 = 5k (color) 10 = 10k (color)
.SaveSelectedOnly	Set to TRUE (-1) to save selected items only. Set to FALSE (0) to save entire document.
.FileVersion	Lets you specify the file version of the document being saved. 0 = Version 9.0 1 = Version 5.0 2 = Version 6.0 3 = Version 7.0 4 = Version 8.0
.IncludeCMXData	Set to TRUE (-1) to include CMX data with the saved file. Set to FALSE (0) to disable this feature.

Example

```
.FileSave "C:\TEST1.CDR", 1, 0, 0, 0
```

The above example saves a version 9 CorelDRAW document named "TEST1.CDR", with a 1k thumbnail. CMX data is not saved.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetCDRFileKeywords (DRAW)

.GetCDRFileKeywords .Filename = *filename*

This command returns the keywords of the current document.

Parameter	Description
.Filename	Returns the keywords for the current document.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetCDRFileCompRatio (DRAW)

.GetCDRFileCompRatio .Filename = *filename*

This command returns the file compression ratio of the current document.

Parameter	Description
.Filename	Returns the file compression ratio for the current document.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetCDRFileLastSavedBy (DRAW)

.GetCDRFileLastSavedBy .Filename = *filename*

This command returns the who saved the file last for the current document.

Parameter	Description
.Filename	Returns the last saved by name for the current document.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetCDRFileNotes (DRAW)

.GetCDRFileNotes .Filename = *filename*

This command returns the notes of the current document.

Parameter	Description
.Filename	Returns the notes for the current document.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetCDRFileThumbnail (DRAW)

.GetCDRFileThumbnail .CDRFilename = *filename,* .BMPFilename = *filename*

This command extracts the thumbnail of the CorelDRAW file and returns a BMP file.

Parameter	Description
.CDRFilename	Lets you specify the CorelDRAW file name from which to extract the thumbnail.
.BMPFilename	Returns the thumbnail image for the specified CorelDRAW file name.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetCDRFileVersion (DRAW)

.GetCDRFileVersion .Filename = *filename*

This command returns the file version of the current document.

Parameter	Description
.Filename	Returns the file name for the current document.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetCurrentWorkspaceDescription (DRAW)

.GetCurrentWorkspaceDescription .Name = *string*

This command returns the description for a specifies workspace.

Parameter	Description
.Name	Lets you specify the workspace name for which you want a description.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetCurrentWorkspaceName (DRAW)

.GetCurrentWorkspaceName

This command returns the current workspace name.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetWorkspaceCount (DRAW)

.GetWorkspaceCount

This command returns the number of workspaces.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetWorkspaceDescription (DRAW)

.GetWorkspaceDescription .Name = *string*

This command returns the workspace description.

Parameter	Description
.Name	Lets you specify the workspace name for which you want the description.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

GetWorkspaceName (DRAW)

.GetWorkspaceName .Index = *long*

This command returns the name of the I-th workspace.

Parameter	Description
.Index	Lets you specify the workspace index number for which you want the name.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

IsDefaultWorkspace (DRAW)

.IsDefaultWorkspace .Name = *string*

This command specifies the default workspace.

Parameter	Description
.Name	Lets you specify the workspace name that you want to make the default workspace.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

IsDocument (DRAW)

This command returns true if a document is open.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

SetCurrentWorkspace (DRAW)

.SetCurrentWorkspace .Name = *string*

This command saves the current CorelDRAW workspace settings.

Parameter	Description
.Name	Lets you specify the workspace name to save for the current settings.

{button ,AL(`OVR1 File commands;',0,"Defaultoverview",)} **Related Topics**

Edit commands

BeginCommandGroup (DRAW)

.BeginCommandGroup .UndoString = *string*

This command starts a group of commands to have a clean undo stack.

Parameter	Description
.UndoString	Lets you specify the string to be used.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

CopyPropertiesFrom (DRAW)

.CopyPropertiesFrom .FromObjectID = *long*, .OutlinePen = *boolean*, .OutlineColor = *boolean*, .Fill = *boolean*, .TextAttibutes = *boolean*

This command copies the properties from the object with the specified object ID to the selected object.

Parameter	Description
.FromObjectID	Lets you specify the object ID of the source object. Use .GetObjectsCDRStaticID to get an object's ID.
.OutlinePen	Set to TRUE (-1) to copy outline pen properties. Set to FALSE (0) to exclude outline pen properties.
.OutlineColor	Set to TRUE (-1) to copy outline color properties. Set to FALSE (0) to exclude outline color properties.
.Fill	Set to TRUE (-1) to copy fill properties. Set to FALSE (0) to exclude fill properties.
.TextAttibutes	Set to TRUE (-1) to copy text properties. Set to FALSE (0) to exclude text properties.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

CopyToClipboard (DRAW)

.CopyToClipboard

This command places a copy of the selected object(s) or text onto the Clipboard.

Example

```
.CreateRectangle 750000, -750000, 0, 0, 0
.CopyToClipboard
.InsertPages 0, 2
.PasteFromClipboard
```

The above example copies a rectangle to the Clipboard, inserts 2 pages, then pastes the contents of the Clipboard to the third page.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

CutToClipboard (DRAW)

.CutToClipboard

This command removes the selected object(s) or text from your document and places a copy onto the Clipboard.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

EndCommandGroup (DRAW)

.EndCommandGroup

This command ends a group of commands.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

InsertOLEObject (DRAW)

.InsertOLEObject .ProgID = *string*

This command inserts an OLE object in a CorelDRAW document.

Parameter	Description
.ProgID	Lets you specify the OLE object's Windows registry name.

Example

```
.InsertOLEObject "CorelPhotoPaint.Image.6"
```

The above example inserts a Corel PHOTO-PAINT image into a CorelDRAW document.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

InsertOLEObjectFromFile (DRAW)

.InsertOLEObjectFromFile .FileName = *string*, .CreateLink = *boolean*

This command inserts an OLE object from a file into a CorelDRAW document.

Parameter	Description
.FileName	The filename.
.CreateLink	Set to TRUE (-1) to create a link. Set to FALSE (0) to disable this option.

Example

```
.InsertOLEObjectFromFile "C:\WINWORD\WORDFILE.DOC", -1
```

The above example inserts a Microsoft Word file in a CorelDRAW document.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

OLEObjectDoVerb (DRAW)

.OLEObjectDoVerb .Verb = *long*

This command performs the specified action on an OLE object.

Parameter	Description
.Verb	Lets you specify the OLE object action to perform. 0 = Primary 1 = Secondary 2 = Tertiary etc.

Note

- Primary and secondary verbs depend on the object type.

Example

```
.InsertOLEObject "CorelPhotoPaint.Image.7"
.OLEObjectDoVerb 0
```

The above example inserts a Corel PHOTO-PAINT OLE object into a DRAW document and invokes in-place editing.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

PasteCustomClipboardFormat (DRAW)

.PasteCustomClipboardFormat .Format = *string*

This command specifies the custom format for pasting from the Clipboard.

Parameter	Description
.Format	Lets you specify the type of format. Options include: "Corel 32-bit Presentation Exchange Data" "Corel Presentation Exchange Data" "Corel Metafile" "Rich Text Format"

Example

```
.PasteCustomClipboardFormat "Rich Text Format"
```

The above example inserts the contents of the Clipboard into a CorelDRAW document as Rich Text.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

PasteFromClipboard (DRAW)

.PasteFromClipboard

This command places a copy of the object(s) on the Clipboard into your drawing.

Example

```
.CreateRectangle 750000, -750000, 0, 0, 0
.CopyToClipboard
.InsertPages 0, 2
.PasteFromClipboard
```

The above example copies a rectangle to the Clipboard, inserts 2 pages, then pastes the contents of the Clipboard in to the last page inserted.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

PasteSystemClipboardFormat (DRAW)

.PasteSystemClipboardFormat .Format = *long*

This command specifies the system format for pasting from the Clipboard.

Parameter	Description
.Format	Lets you specify the type of format. 1 = CF Text 2 = Bitmap 3 = Metafile Pict 8 = DIB 14 = Enhanced Metafile

Example

```
.PasteSystemClipboardFormat 2
```

The above example pastes a bitmap from the Clipboard into the active document.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

Redo (DRAW)

.Redo

This command restores changes reversed by the Undo command. Redo becomes available immediately after you select the Undo command.

Example

```
.Redo
```

The above command reverses the last .Undo command and reinstates the previous deletion or reversal of actions.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

Repeat (DRAW)

.Repeat

This command applies, if possible, the most recent command or action to selected object.

Example

```
.Repeat
```

The above example repeats the last command.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

RepeatLastCommand (DRAW)

.RepeatLastCommand

This command applies, if possible, the last command or action to selected object.

Example

```
.RepeatLastCommand
```

The above example repeats the last command.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

SetErrorHandling (DRAW)

.SetErrorHandling .Msg = *boolean*

This command sets how the errors should be returned.

Parameter	Description
.Msg	Set to TRUE (-1) returns error messages. Set to FALSE (0) returns error code.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

Undo (DRAW)

.Undo

This command reverses actions performed during the current session. Use Undo after you have made a change you do not want to implement. Immediately after you select .Undo, the .Redo command becomes available, allowing you to restore what you just undid. You cannot undo the following operations: any change of view (e.g., Zoom-in or Zoom-out); any file operations (e.g., Open, Save, or Import); any selection operations (e.g., Marquee select or Node select).

Example

```
.Undo
```

The above example undoes the last command.

{button ,AL(`OVR1 Edit commands;',0,"Defaultoverview",)} **Related Topics**

View commands

RedrawAllScreens (DRAW)

.RedrawAllScreens

This command forces CorelDRAW to redraw all open document windows.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

RedrawScreen (DRAW)

.RedrawScreen

This command forces CorelDRAW to redraw the windows of the active document.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

ResumePainting (DRAW)

.ResumePainting

This command instructs CorelDRAW to resume screen updating. To stop screen updating, use the .SupressPainting command.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

SetFullScreenPreview (DRAW)

.SetFullScreenPreview .FullScreen = *boolean*

This command removes everything but your drawing from the screen. You cannot edit your drawing in this mode.

Parameter	Description
.FullScreen	Set to TRUE (-1) to remove everything but your drawing from the screen. Set to FALSE (0) to return to normal mode.

Example

```
.SetFullScreenPreview -1
```

The above example displays a full-screen preview of the active image.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

SetVisible (DRAW)

.SetVisible .Visible = *boolean*

This command makes the CorelDRAW Script Edtior visible.

Parameter	Description
.Visible	Set to TRUE (-1) to show the CorelDRAW Script Editor. Set to False (0) to hide the CorelDRAW Script Editor.

Example

```
.SetVisible -1
```

The above example makes the CorelDRAW Script Editor visible.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

SuppressPainting (DRAW)

.SuppressPainting ShowDialog = *boolean*

This command instructs CorelDRAW to suppress screen updating. To resume screen updating, use the .ResumePainting command.

Parameter	Description
.ShowDialog	Set to TRUE (-1) displays the script running dialog box. Set to FALSE (0) disables the script running message dialog box.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

ZoomIn (DRAW)

.ZoomIn

This command changes the view by zooming in twice the size.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

ZoomOut (DRAW)

.ZoomOut

This command changes the view by zooming out half the size.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

ZoomToAllObjects (DRAW)

.ZoomToAllObjects

This command changes the view by zooming to all the objects on the page.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

ZoomToHeight (DRAW)

.ZoomToHeight

This command changes the view by zooming in to the page height.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

ZoomToPage (DRAW)

.ZoomToPage

This command changes the view by zooming to the page.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

ZoomToRectangle (DRAW)

.ZoomToRectangle .Top = *long*, .Left = *long*, .Bottom = *long*, .Right = *long*

This command changes the view by zooming in on a specified area.

Parameter	Description
.Top	Lets you specify the Y coordinate of the top of the rectangle in tenths of a micron.
.Left	Lets you specify the X coordinate of the left of the rectangle in tenths of a micron.
.Bottom	Lets you specify the Y coordinate of the bottom of the rectangle in tenths of a micron.
.Right	Lets you specify the X coordinate of the right of the rectangle in tenths of a micron.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

ZoomToSelection (DRAW)

.ZoomToSelection

This command changes the view by zooming in on the current selection.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

ZoomToWidth (DRAW)

.ZoomToWidth

This command changes the view by zooming in on the page width.

{button ,AL(`OVR1 View commands;',0,"Defaultoverview",)} **Related Topics**

Layout commands

AddPageFrame (DRAW)

.AddPageFrame

This command puts a printable background frame around the page.

Example

```
.AddPageFrame
```

The above example creates a frame around the new page.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

ChangeLayerColor (DRAW)

.ChangeLayerColor .LayerName = *string,* .PageNum = *long*

This command lets you set the outline color for the override color..

Parameter	Description
.LayerName	Lets you specify the name of the Layer.
.PageNum	Lets you specify the pafe number: 0 = Master

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

ChangeLayerName (DRAW)

.ChangeLayerName .NewLayerName = *string*

This command lets you assign a new name to the active layer.

Parameter	Description
.NewLayerName	Lets you specify the new name of the Layer.

Example

```
.ChangeLayerName "NewName"
```

The above example changes the layer name to "NewName."

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

CopyToLayer (DRAW)

.CopyToLayer .LayerName = *string*

This command places a copy of the selected object on the layer indicated in the LayerName.

Parameter	Description
.LayerName	Lets you specify the name of the destination layer.

Example

```
.CreateRectangle -200000, 200000, -900000, 900000, 0
.CopyToLayer "Layer2"
```

The above example creates a rectangle and copies it to "Layer2."

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

DeleteLayer (DRAW)

.DeleteLayer .LayerName = *string*

This command deletes the active layer and any objects on it.

Parameter	Description
.LayerName	Lets you specify the name of the layer to delete. This parameter is optional. If no layer name is specified the active layer is deleted.

Example

```
.MoveToLayer "NewLayer1"
.DeleteLayer
```

The above example moves to the layer named "NewLayer 1" and deletes it.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

DeletePages (DRAW)

.DeletePages .UnusedParameter, .NumberOfPages = *long,* .StartPage = *long*

This command deletes pages from the current drawing.

Parameter	Description
.UnusedParameter	This parameter is not used
.NumberOfPages	Lets you specify the number of pages to delete. Note: The current page is included in the deletion.
.StartPage	Lets you specify the page number to begin deleting pages to delete.

Example

```
.CreateRectangle 750000, -750000, 0, 0, 0
.CopyToClipboard
.InsertPages 0, 4
.PasteFromClipboard
.DeletePages , 2, 2
```

The above example inserts 4 pages after the current page, pastes the contents of the Clipboard on the fourth page, then deletes two pages starting on the second page.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

DisplayFacingPages (DRAW)

.DisplayFacingPages .FacingPages = *boolean*, .LeftFirst = *boolean*

This command displays two consecutive pages on the screen at the same time.

Parameter	Description
.FacingPages	Set to TRUE (-1) to display two consecutive pages on the screen at the same time. Working in this view allows you to draw objects that lie partially on both pages at the same time. Set to FALSE (0) to disable this option.
.LeftFirst	Set to TRUE (-1) to display odd pages on the left. Set to FALSE (0) to display odd pages on the right.

Example

```
.FileNew
.DisplayFacingPages 0, -1 'Displays one page
```

The above example displays one page.

```
.FileNew
.CreateEllipse -250000, -500000, 250000, 500000, 0, 0, 0
.CreateRectangle 750000, -750000, 0, 0, 0
.CopyToClipboard
.InsertPages 0, 4
.PasteFromClipboard
.DisplayFacingPages -1, -1 'Displays two pages
```

The above example displays facing pages with the current page on the left.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

EditLayer (DRAW)

.EditLayer .LayerName = *string*, .PageNum = *long,* .NewName = *string,* .Visible = *boolean,* .Printable = *boolean,* .Locked = *boolean,* .Master = *boolean,* .WireOverride = *boolean,* .ChangeColor = *boolean*

This command sets all the layer properties.

Parameter	Description
.LayerName	Lets you specify the layer name.
.PageNum	Lets you specify the page number.
,NewName	Lets you specify the new name for the layer.
.Visible	Set to TRUE (-1) makes the layer visible.
.Printable	Set to TRUE (-1) makes the layer printable.
.Locked	Set to TRUE (-1) makes the layer locked.
.Master	Set to TRUE (-1) makes the layer the master layer.
.WireOverride	Set to TRUE (-1) forces the view to wireframe display.
.ChangeColor	Set to TRUE (-1) changes the color of the wireframe view.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

GetCurrentPageName (DRAW)

.GetCurrentPageName

This command returns the name of the current page.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

GetCurrentPageOrientation (DRAW)

.GetCurrentPageOrientation .Orient = *long*

This command returns the orientation of the current document page.

Parameter	Description
.Orient	Lets you specify the page orientation: 0 = Portrait 1 = Landscape

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

GetCurrentPageSize (DRAW)

.GetCurrentPageSize .Width = *long,* .Height = *long*

This command returns the width and height of the current document page.

Parameter	Description
.Width	Returns the width of the current page in tenths of a micron.
.Height	Returns the height of the current page in tenths of a micron.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

GetDocumentCount (DRAW)

.GetDocumentCount

This command returns the number of open documents.

Example

ICount& = .GetDocumentCount()

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

GetDocumentName (DRAW)

.GetDocumentName .Name = *string**

This command returns the name of the current document.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

GetPageCount (DRAW)

.GetPageCount

This command returns the number of pages in the current document.

Parameter	Description
.Name	Returns the name of the current document.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

GetPageSize (DRAW)

.GetPageSize .Width = *long**, .Height = *long**

This command returns the width and height of the document page.

Parameter	Description
.Width	Returns the width of the page in tenths of a micron.
.Height	Returns the height of the page in tenths of a micron.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

InsertPages (DRAW)

.InsertPages .BeforePage = *boolean*, .NumberOfPages = *long*[, .StartPageNumber = *long*]

This command inserts the specified number of pages into the current drawing.

Parameter	Description
.BeforeCurrentPage	Set to TRUE (-1) to position insertion point before the current page. Set to FALSE (0) to position insertion point after the current page.
.NumberOfPages	Lets you specify the number of pages to insert.
.StartPageNumber (optional)	Lets you specify the page number from where to insert pages.

Example

```
.InsertPages 0, 4, 5
```

The above example inserts 4 pages after the current page, starting on the fifth page.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

MoveLayerTo (DRAW)

.MoveLayerTo .DestLayer = *string,* .BeforeDest = *boolean*

This command moves the current layer to another position.

Parameter	Description
.DestLayer	Lets you specify the name of the destination layer.
.BeforeDest	Set to TRUE (-1) places the layer before the destination layer.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

MoveToLayer (DRAW)

.MoveToLayer .LayerName = *string*

This command moves the selected object to the layer selected in the Layers list.

Parameter	Description
.LayerName	Lets you specify the name of the destination layer.

Example

```
.MoveToLayer "NewLayer1"
```

The above example moves the selected object(s) to the layer named "NewLayer1."

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

NewLayer (DRAW)

.NewLayer .LayerName = *string*

This command lets you create a new layer and assign a name.

Parameter	Description
.LayerName	Lets you specify the name of the new layer.

Example

```
.NewLayer "NewLayer1"
```

The above example creates a new layer named "NewLayer1."

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SelectLayer (DRAW)

.SelectLayer .LayerName = *string*

This command lets you select a layer, making it the active layer.

Parameter	Description
.LayerName	Lets you specify the name of the selected layer.

Example

```
.SelectLayer "NewLayer1"
```

The above example selects the layer named "NewLayer1" and makes it the active layer.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetApplyToDuplicate (DRAW)

.SetApplyToDuplicate .ApplyToDuplicate = *boolean*

This command opens and closes a block of object-duplicating commands. An object must be selected to use this command. The duplicated object can be repositioned, resized, skewed, or rotated.

Parameter	Description
.ApplyToDuplicate	Set to TRUE (-1) to open a block of object-duplicating commands. Set to FALSE (0) to close the block.

Note

- The following commands can be used to duplicate objects within the .SetApplyToDuplicate block:

  .SetPosition
  .SkewObject
  .SetSize
  .RotateObject

  The duplicated object is selected.

Example

```
.CreateEllipse -250000, -500000, 250000, 500000, 0, 0, 0
.SetPosition 55555, 900000
.SetApplyToDuplicate TRUE
.SetPosition 0, 0 'Creates another object
.ApplyUniformFillColor 2, 255, 0, 0, 0
.SetPosition 55555, 100000 'Creates another object
.ApplyUniformFillColor 2, 0, 255, 0, 0
.SkewObject -15000000, 2000000, 3 'Creates another object
.SetSize 444444, 555555 'Creates another object
.RotateObject 45000000, 0, 0, 0 'Creates another object
.SetApplyToDuplicate FALSE
.SetPosition 0, 0
```

The above example creates an ellipse then creates 5 more ellipses in the SetApplyToDuplicate block.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetColorOverride (DRAW)

.SetColorOverride .Override = *boolean*

This command outlines objects on a layer in the selected color. Objects on the selected layer will appear with a wireframe outline of the chosen color.

Parameter	Description
.Override	Set to TRUE (-1) to outline objects on a layer in the selected color. Set to FALSE (0) to disable this option.

Example

```
.StoreColor DRAW_COLORMODEL_PANTONE, 3, 255, 0, 0
.SetColorOverride
```

The above example sets the override color to cyan.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetCurrentDocument (DRAW)

.SetCurrentDocument .

This command makes the selected document the current document.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetCurrentPage (DRAW)

.SetCurrentPage .CurrentPage = *long*

This command makes the specified page the current page.

Parameter	Description
.CurrentPage	Lets you specify which page to make the current page.

Example

```
.SetCurrentPage 2
```

The above example sets the second page as the current page.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetCurrentPageName (DRAW)

.SetCurrentPageName .Name = *string*

This command sets the current page name.

Parameter	Description
.Name	Lets you specify the name for the current page.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetCurrentPageOrientation (DRAW)

.SetCurrentPageOrientation .Orient = *long*

This command sets the orientation for the current page.

Parameter	Description
.Orient	Lets you specify the page orientation: 0 = Portrait 1 = Landscape

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetCurrentPageSize (DRAW)

.SetCurrentPageSize .Width = *long,* .Height = *long*

This command sets the size for the current page.

Parameter	Description
.Width	Lets you specify the page width in tenths of a micron.
.Height	Lets you specify the page height in tenths of a micron.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetDocVisible (DRAW)

.SetDocVisible .Show = *boolean*

This command makes the current document visible or hidden.

Parameter	Description
.Show	Set to TRUE (-1) to make a document visible. Set to FALSE (0) to hide a document.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetLayerLocked (DRAW)

.SetLayerLocked .Locked = *boolean*[, .LayerName = *string*][, .PageNum = *long*]

This command enables or disables selection of objects on a layer. Locking a layer prevents objects on it from being accidentally moved or changed in any way. You cannot add new objects to a locked layer.

Parameter	Description
.Locked	Set to TRUE (-1) to lock a layer, preventing objects on it from being accidentally moved or changed in any way. You cannot add new objects to a locked layer. Set to FALSE (0) to unlock a layer.
.LayerName (optional)	Lets you specify the layer name.
.PageNum (optional)	Specifes the page number. 0=Master

Example

```
.SetLayerLocked -1, 3, 2
```

The above example locks third layer on the second page.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetLayerPrintable (DRAW)

.SetLayerPrintable .Printable = *boolean*[, .LayerName = *string*][, .PageNum = *long*]

This command enables or disables printing of objects on the current layer.

Parameter	Description
.Printable	Set to TRUE (-1) to enable printing of the current layer. Set to FALSE (0) to disable printing of the current layer.
.LayerName (optional)	Lets you specify the layer name.
.PageNum (optional)	Specifes the page number. 0=Master

Note

- If .SetOptionsForAllPages is set TRUE (-1), then the .SetLayerPrintable command applies to all pages.

Example

```
.SetLayerPrintable 0, 3, 2
```

The above example disables printing of the third layer on the second page.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetLayerVisible (DRAW)

.SetLayerVisible .Visible = *boolean*[, .LayerName = *string*][, .PageNum = *long*]

This command makes objects on a layer visible or invisible.

Parameter	Description
.Visible	Set to TRUE (-1) to make the current layer visible. Set to FALSE (0) to make the current layer invisible.
.LayerName (optional)	Lets you specify the layer name.
.PageNum (optional)	Specifes the page number. 0=Master

Note

- If .SetOptionsForAllPages is set TRUE (-1), then the .SetLayerVisible command applies to all pages.

Example

```
.SetLayerVisible -1, 3, 2
```

The above example makes the third layer on the second page visible.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetMultiLayer (DRAW)

.SetMultiLayer .MultiLayer = *boolean*

This command lets you select objects on all layers that are not locked or invisible.

Parameter	Description
.MultiLayer	Set to TRUE (-1) to enable selection of objects across all layers except those which are locked or invisible. Set to FALSE (0) to disable selection of objects across all layers only objects on the current layer can be selected.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetOptionsForAllPages (DRAW)

.SetOptionsForAllPages .AllPages = *boolean*

This command enables CorelDRAW options to be set for all pages.

Parameter	Description
.AllPages	Set to TRUE (-1) to enable options to be set for all pages. Set to FALSE (0) to disable this option.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetPageLayout (DRAW)

.SetPageLayout .LayoutType = *long*

This command lets you specify a page layout.

Parameter	Description
.LayoutType	Lets you specify the style of the page layout: 1 = Full Page: Prints one full page per sheet. 2 = Book: Prints two pages per sheet, which you would cut down the middle. 3 = Booklet: Prints two pages per sheet, which you would fold vertically to obtain a side fold. 4 = Tent Card: Prints two pages per sheet, which you would fold horizontally to obtain a top fold. 5 = Side-Fold Card: Prints four pages per sheet, which you would fold first horizontally to create the top fold, then vertically to create the side fold. 6 = Top-Fold Card: Prints four pages per sheet, which you would fold first vertically to create the side fold, then horizontally to create the top fold.

Example

```
.SetPageLayout 3
```

The above example sets the page layout to booklet style.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetPageOrientation (DRAW)

.SetPageOrientation .Orient = *long*

This command changes the orientation of the page.

Parameter	Description
.Orient	DRAW_ORIENT_PORTRAIT = portrait DRAW_ORIENT_LANDSCAPE = landscape

Note

- You can include the SCPCONST.CSI and DRWCONST.SCI files in your script. These files define contants for all referenced parameters used with the .SetPageOrientation command.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetPageSize (DRAW)

.SetPageSize .Width = *long*, .Height = *long*

This command lets you set the page size for the document.

Parameter	Description
.Width	Lets you specify the new page width in tenths of a micron.
.Height	Lets you specify the new page height in tenths of a micron.

Note

- You can use the LENGTHCONVERT function, or one of the FROM... or TO... functions to specify length measurements.

Example

```
.SetPageSize 1000000,1350000
```

The above example sets the page size to 1,000,000 microns wide by 1,350,000 microns high (or 3.94 inches by 5.31 inches).

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetPageSizeFromPrinter (DRAW)

.SetPageSizeFromPrinter

This command sets the page size and orientation of the current document to the current settings of the default printer.

Example

```
.SetPageSizeFromPrinter
```

The above example queries the printer to set the page size.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetPaperColor (DRAW)

.SetPaperColor .

This command lets you color the Preview screen (and the Drawing Window, if you are working in the Editable Preview) to approximate the paper you plan to print it on.

Example

```
.StoreColor DRAW_COLORMODEL_CMYK, 0, 255, 0, 0
.SetPaperColor
```

The above example sets the paper color to magenta.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetPosition (DRAW)

.SetPosition .XPos = *long*, .YPos = *long*

This command sets the position for placement of the selected object

Parameter	Description
.XPos	Lets you specify the X-coordinate of the new position in tenths of a micron.
.YPos	Lets you specify the Y-coordinate of the new position in tenths of a micron.

Example

```
.CreateRectangle 1350000, -1000000, 750000, -500000, 0
.CreateArtisticText "1"
.SetPosition -950000, 1250000
```

The above example creates a rectangle and positions a number '1' in its upper-left corner.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetReferencePoint (DRAW)

.SetReferencePoint .ReferencePoint = *long*

This command sets the specified Reference Point for a selected object. The reference point is used to set the object handle for subsequent commands such as .SetPosition.

Parameter	Description
.ReferencePoint	Lets you specify the reference point to set. 1 = Upper-right 2 = Upper-middle 3 = Upper-left 4 = Middle-left 5 = Lower-left 6 = Lower-middle 7 = Lower-right 8 = Middle-right 9 = Center

Note

- You can include the SCPCONST.CSI and DRWCONST.SCI files in your script. These files define contants for all referenced parameters used with the .SetReferencePoint command.

Example

```
.CreateRectangle 1250000, -1000000, 750000, -500000, 0
.SetReferencePoint 9
.SetPosition 0, 0
```

The above example creates a rectangle, sets its reference point to the center and positions it in the center of the page.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetSize (DRAW)

.SetSize .XSize = *long*, .YSize = *long*

This command lets you scale, mirror, or set the size of the selected object.

Parameter	Description
.XSize	Lets you specify the new horizontal size of the selected object, in tenths of a micron.
.YSize	Lets you specify the new vertical size of the selected object, in tenths of a micron.

Note

- To mirror an object, use negative values for the .XSize and .YSize parameters.

Example

```
.CreateRectangle 1000000, 750000, 500000, 100000, 0
id& = .GetObjectsCDRStaticID()
status& = .GetSize (XSize&, YSize&)
.SelectObjectOfCDRStaticID id&
.SetSize 2*XSize&, 3*YSize&
```

The above example gets the size of the selected rectangle and sets the width to twice the original size, and the height to three times the original size.

```
.CreateRectangle 1000000, 750000, 500000, 100000, 0
id& = .GetObjectsCDRStaticID()
status& = .GetSize (XSize&, YSize&)
.SelectObjectOfCDRStaticID id&
.SetSize -XSize&, YSize&
```

The above example horizontally mirrors the selected object, maintaining its original size.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

SetToMasterLayer (DRAW)

.SetToMasterLayer .Master = *boolean*[, .LayerName = *string*]

This command lets you set the selected object to a master layer. When you want the same element, for example, a company logo, to appear on every page of a document, use this command to set the "master layers" to contain the repeating elements.

Parameter	Description
.Master	Set to TRUE (-1) to enable, applying the Master Layer template to all layers. Set to FALSE (0) to disable this option.
.LayerName (optional)	Lets you specify the layer name.

Example

```
.CreateRectangle 1350000, -1000000, 750000, -500000, 0
.SetToMasterLayer -1
```

The above example sets the rectangle to the master layer.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

ShowPageBorder (DRAW)

.ShowPageBorder .ShowBorder = *boolean*

This command enables and disables the page border.

Parameter	Description
.ShowBorder	Set to TRUE (-1) to show the page border. Set to FALSE (0) to suppress the page border.

Example

```
.ShowPageBorder -1
```

The above example shows the page border.

```
.ShowPageBorder 0
```

The above example hides the page border.

{button ,AL(`OVR1 Layout commands;',0,"Defaultoverview",)} **Related Topics**

Styles commands

ApplyStyle (DRAW)

.ApplyStyle .Style = *string*

This command lets you apply a style to the selected object.

Parameter	Description
.Style	Lets you specify the name of the style.

Example

```
.SelectAllObjects
.ApplyStyle "Default Graphic"
```

The above example applies the 'Default Graphic' style to all selected objects.

{button ,AL(`OVR1 Styles commands;',0,"Defaultoverview",)} **Related Topics**

CreateNewStyle (DRAW)

.CreateNewStyle .StyleType = *long,* .StyleName = *string*

This command lets you create a new style.

Parameter	Description
.StyleType	Lets you specify the type of style: 0 = Artistic text 1 = Paragraph text 2= Graphic
.StyleName	Lets you specify the name of the new style.

{button ,AL(`OVR1 Styles commands;',0,"Defaultoverview",)} **Related Topics**

DeleteStyle (DRAW)

.DeleteStyle .Style = *string*

This command deletes styles. When you delete a style, objects with that style revert to the default style for that object type. The object's appearance does not change when it reverts to the default style.

Parameter	Description
.Style	Lets you specify the name of the style to delete.

Example

```
.DeleteStyle "Style 1"
```

The above example deletes the style named "Style 1."

{button ,AL(`OVR1 Styles commands;',0,"Defaultoverview",)} **Related Topics**

LoadStyles (DRAW)

.LoadStyles .StyleSheet = *string*

This command loads the styles from a template into the active drawing.

Parameter	Description
.StyleSheet	Lets you specify the name of the template to use.

Example

```
.LoadStyles "C:\mine.cdt"
```

The above example loads the styles from the template file "MINE.CDT" into the active document.

{button ,AL(`OVR1 Styles commands;',0,"Defaultoverview",)} **Related Topics**

RenameStyle (DRAW)

.RenameStyle .OldName = *string*, .NewName = *string*

This command renames a style name.

Parameter	Description
.OldName	Lets you specify the old name of the style.
.NewName	Lets you specify the new name of the style.

{button ,AL(`OVR1 Styles commands;',0,"Defaultoverview",)} **Related Topics**

RevertToStyle (DRAW)

.RevertToStyle

This command converts an object to its original style.

{button ,AL(`OVR1 Styles commands;',0,"Defaultoverview",)} **Related Topics**

SaveStyleAs (DRAW)

.SaveStyleAs .Style = *string*, .Fill = *boolean*, .Outline = *boolean*, .Typeface = *boolean*, .TypeStyle = *boolean*, .Size = *boolean*, .Justification = *boolean*, .Tabs = *boolean*, .Hyphenation = *boolean*, .SpaceChar = *boolean*, .SpaceWord = *boolean*, .SpaceLine = *boolean*, .BeforePara = *boolean*, .AfterPara = *boolean*, .Underline = *boolean*, .Overline = *boolean*, .Strikeout = *boolean*, .BulletIndent = *boolean*, .FirstLineIndent = *boolean*, .RestOfLinesIndent = *boolean*, .RightMargin = *boolean*, .SuperOrSubScript = *boolean*, .Capitalize = *boolean*, .Bullet = *boolean*

This command saves the style of the current object as a new style.

Parameter	Description
.Style	Lets you specify the name of the new style.
.Fill	Set to TRUE (-1) to include fill properties. Set to FALSE (0) to exclude these properties.
.Outline	Set to TRUE (-1) to include outline properties. Set to FALSE (0) to exclude these properties.
.Typeface	Set to TRUE (-1) to include typeface properties. Set to FALSE (0) to exclude these properties.
.TypeStyle	Set to TRUE (-1) to include type style properties. Set to FALSE (0) to exclude these properties.
.Size	Set to TRUE (-1) to include size properties. Set to FALSE (0) to exclude these properties.
.Justification	Set to TRUE (-1) to include text justification properties. Set to FALSE (0) to exclude these properties.
.Tabs	Set to TRUE (-1) to include tab stop properties. Set to FALSE (0) to exclude these properties.
.Hyphenation	Set to TRUE (-1) to include hyphenation properties. Set to FALSE (0) to exclude these properties.
.SpaceChar	Set to TRUE (-1) to include character spacing properties. Set to FALSE (0) to exclude these properties.
.SpaceWord	Set to TRUE (-1) to include word spacing properties. Set to FALSE (0) to exclude these properties.
.SpaceLine	Set to TRUE (-1) to include line spacing properties. Set to FALSE (0) to exclude these properties.
.BeforePara	Set to TRUE (-1) to include paragraph spacing properties (before the paragraph). Set to FALSE (0) to exclude these properties.
.AfterPara	Set to TRUE (-1) to include paragraph spacing properties (after the paragraph). Set to FALSE (0) to exclude these properties.
.Underline	Set to TRUE (-1) to include text underline properties. Set to FALSE (0) to exclude these properties.
.Overline	Set to TRUE (-1) to include text overline properties. Set to FALSE (0) to exclude these properties.
.Strikeout	Set to TRUE (-1) to include text strikeout properties. Set to FALSE (0) to exclude these properties.
.BulletIndent	Set to TRUE (-1) to include bullet indentation properties. Set to FALSE (0) to exclude these properties.
.FirstLineIndent	Set to TRUE (-1) to include indentation properties for the first line. Set to FALSE (0) to exclude these properties.
.RestOfLinesIndent	Set to TRUE (-1) to include indentation properties for remaining lines (hanging indent). Set to FALSE (0) to exclude these properties.
.RightMargin	Set to TRUE (-1) to include right margin properties. Set to FALSE (0) to exclude these properties.
.SuperOrSubScript	Set to TRUE (-1) to include superscript or subscript properties. Set to FALSE (0) to exclude these properties.
.Capitalize	Set to TRUE (-1) to include capitalization properties. Set to FALSE (0) to exclude these properties.
.Bullet	Set to TRUE (-1) to include bullet properties. Set to FALSE (0) to exclude these properties.

{button ,AL(`OVR1 Styles commands;',0,"Defaultoverview",)} **Related Topics**

SaveStyleProp (DRAW)

.SaveStyleProp .StyleName = *string,* .UseFill = *boolean,* .UseOutline = *boolean,* .UseFont = *long,* .UseAlignment = *long,* .UseSpacing = *long,* .UseLines = *long,* .UseIndentsAndMargins = *long,* .UseTextEffects = *long*

This command lets you save the style based on the current properties.

Parameter	Description
.StyleName	Lets you specify the name of the new Style.
.UseFill	Set to TRUE (-1) saves the fill as part of the style.
.UseOutline	Set to TRUE (-1) saves the outline as part of the style.
.UseFont	Lets you specify the font: 1 = Typeface 2 = Typestyle 4 = Size
.UseAlignment	Specifices the alignement: 1 = Justification 2 = Tabs 4 = Hyphenation
.UseSpacing	Lets you specify the spacing: 1 = InterChar Spacing 2 = InterWord Spacing 4 = Interline Spacing 8 = Spacing Before Paragraphs
.UseLines	Lets you specify the lines: 1 = Underline 2 = Overline 4 = Strikeout
.UseIndentsAndMargins	Lets you specify the indents and margins: 1 = Bullet Indent 2 = First Line Indent 4 = Rest of Lines Indent 8 = Right Margin
.UseTextEffects	Lets you specify the text effect 1 = Superscript/Subscript 2 = Capitalization Effect 4= Bullet

{button ,AL(`OVR1 Styles commands;',0,"Defaultoverview",)} **Related Topics**

SaveTemplate (DRAW)

.SaveTemplate .StyleSheet = *string*

This command lets you save the styles in the active document as a template.

Parameter	Description
.StyleSheet	Lets you specify the name of the Style Sheet to save.

Example

```
.SaveTemplate "C:\TMPLATE1.CDT"
```

The above example saves a template named "TMPLATE1.CDT".

{button ,AL(`OVR1 Styles commands;',0,"Defaultoverview",)} **Related Topics**

Object selection commands

AfterObject (DRAW)

.AfterObject .ObjectID = *long*

This command selects the object that is after the reference object in the object tree. Use .lObjectID to specify the reference object.

Parameter	Description
.ObjectID	Lets you specify the object ID of the reference object. Use .GetObjectsCDRStaticID to get an object's ID.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

AppendObjectToSelection (DRAW)

.AppendObjectToSelection .ObjectID = *long*

This command adds the object with the specified object ID to the existing selection.

Parameter	Description
.ObjectID	Lets you specify the object ID of the object to append. Use .GetObjectsCDRStaticID to get an object's ID.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

BeforeObject (DRAW)

.BeforeObject .ObjectID = *long*

This command selects the object that is before the reference object in the object tree. Use .lObjectID to specify the reference object.

Parameter	Description
.ObjectID	Lets you specify the object ID of the reference object. Use .GetObjectsCDRStaticID to get an object's ID.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

ClearNodeSelection (DRAW)

.ClearNodeSelection

This command clears the selection of nodes.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

FindNextObjectOfStyle (DRAW)

ReturnValue = .FindNextObjectOfStyle()

This function finds the next object with the current style.

Return Value

Returns one of the following values

- TRUE (-1) an object is found
- FALSE (0) no object is found.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

FindObjectOfStyle (DRAW)

ReturnValue = .FindObjectOfStyle(.StyleName = *string*)

This function finds the next object with the specified style.

Return Value

Returns one of the following values:

- TRUE (-1) an object is found
- FALSE (0) no object is found.

Parameter	Description
.StyleName	Lets you specify the name of the style.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

GetObjectType (DRAW)

ReturnValue& = .GetObjectType()

This function returns a value that indicates the type of selected object. If more than one object is selected, the function returns the type of the last selected object.

Return Value

Returns one of the following values:

- 0 Reserved for future use
- 1 Rectangle
- 2 Ellipse
- 3 Curve
- 4 Text
- 5 Bitmap
- 6 Paragraph Text
- 7 OLE
- 9 Symmetrical Polygon
- 12 Grouped objects

Note

- You can include the SCPCONST.CSI and DRWCONST.SCI files in your script. These files define contants for all referenced parameters used with the .GetObjectType command.

Example

```
objType& = .GetObjectType()
MESSAGE objType&
```

The above example displays a number that corresponds to the type of selected object in a message box.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

GetUserClick (DRAW)

.GetUserClick .XCoord = *long*, .YCoord = *long*,[.Timeout = *long*,][.IgnoreSnap = *boolean*]

This function returns the coordinates of a user click.

Parameter	Description
.XCoord	Returns the X coordinate of the selection click.
.YCoord	Returns the Y coordinate of the selection click.
.Timeout (optional)	Lets you specify the amount of time, in seconds, to wait for the user to click. Defaults to 10.
.IgnoreSnap (optional)	Defaults to TRUE (-1) 0 does not ignore the snap -1 ignores the snap

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

IsSelection (DRAW)

.IsSelection

This command returns true if an object is selected.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectAllObjects (DRAW)

.SelectAllObjects

This command selects every object in your drawing, including any not currently in view.

Example

```
.SelectAllObjects
```

The above example selects all objects in the active document.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectNextNode (DRAW)

.SelectNextNode

This command selects the next node in the current curve.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectNextObject (DRAW)

.SelectNextObject .SelectInsideGroup = *boolean*

This command lets you select the next object in the drawing. Repeat this command until the object you want is selected.

Parameter	Description
.SelectInsideGroup	Set to TRUE (-1) to permit object selection within a group of objects. Set to FALSE (0) to disable this option.

Example

```
.SelectNextObject -1
```

The above example selects the next object in the drawing. If that object is in a group, it can be selected.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectNode (DRAW)

.SelectNode .Index = *long,* .AddToSelection = *boolean*

This command lets you select the I-th node of the current curve.

Parameter	Description
.Index	Lets you specify the I-th node of the current curve.
.AddToSelection	Set to TRUE (-1) adds the specified node to the current selection.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectNodeAt (DRAW)

.SelectNodeAt .X = *long,* .Y = *long,* .AddToSelection = *boolean*

This command lets you select a specific node.

Parameter	Description
.X	Lets you specify the X coordinate of the node in the current curve.
.Y	Lets you specify the Y coordinate of the node in the current curve.
.AddToSelection	Set to TRUE (-1) adds the specified node to the current selection.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectObjectAtPoint (DRAW)

.SelectObjectAtPoint(.XPos = *long*, .YPos = *long*, SelectInsideGroup = *boolean*)

This command toggles the selection of an object at the specified point. Using this command is the same as holding down SHIFT and clicking an object during a DRAW session.

Parameter	Description
.XPos	Lets you specify one of the X-coordinates of the selected object in tenths of a micron, relative to the center of the page.
.YPos	Lets you specify one of the Y-coordinates of the selected object in tenths of a micron, relative to the center of the page.
.SelectInsideGroup	Set to TRUE (-1) to permit object selection within a group of objects. Set to FALSE to disable this option.

Example

```
.CreateRectangle 1350000, -1000000, 1300000, 0, 0
.CreateRectangle 1000000, -750000, 500000, 100000, 0
.CreateRectangle 100000, -500000, -100000, 50000, 0
.CreateRectangle -750000, -500000, -250000, 50000, 0
.UnSelectAll
.SelectObjectAtPoint -750000, 500000, 0
.ApplyUniformFillColor 2, 255, 0, 0, 0
```

The above example creates four rectangles, then selects the second one and fills it with cyan.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectObjectOfCDRStaticID (DRAW)

.SelectObjectOfCDRStaticID .CDRStaticID = *long*

This command selects the object with the specified CDRStaticID.

Parameter	Description
.CDRStaticID	Lets you specify the CDRStaticID number of the object to select.

Example

```
.CreateRectangle 750000, -600000, 250000, -100000, 0
IDRect& = .GetObjectsCDRStaticID()
.SelectObjectOfCDRStaticID IDRect&
```

The above example demonstrates object selection using the object's CDRStaticID.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectObjectOfType (DRAW)

.SelectObjectOfType .Type = *long*

This command selects the object of a given type.

Parameter	Description
.Type	Lets you specify the type of object to select.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectObjectsInRect (DRAW)

.SelectObjectsInRect .Top = *long*, .Left = *long*, .Bottom = *long*, .Right = *long*, .IncludeIntersecting = *boolean*

This command selects all objects found within the defined rectangular area

Parameter	Description
.Top	Lets you specify the Y-coordinate of the upper-left corner of the distribution rectangle in tenths of a micron, relative to the center of the page.
.Left	Lets you specify the X-coordinate of the upper-left corner of the distribution rectangle in tenths of a micron, relative to the center of the page.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the distribution rectangle in tenths of a micron, relative to the center of the page.
.Right	Lets you specify the X-coordinate of the lower-right corner of the distribution rectangle in tenths of a micron, relative to the center of the page.
.IncludeIntersecting	Set to TRUE (-1) to included intersecting objects in the selection. Set to FALSE (0) to disable this option.

Example

```
.SelectObjectsInRect 1350000, -1000000, -1350000, 1000000, 0
```

The above example selects all objects within the specified rectangle.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

SelectPreviousObject (DRAW)

.SelectPreviousObject .SelectInsideGroup = *boolean*

This command lets you select the previously selected object in the drawing. Repeat this command until the object you want is selected. The objects are selected in the order in which they were created.

Parameter	Description
.SelectInsideGroup	Set to TRUE (-1) to permit object selection within a group of objects. Set to FALSE (0) to disable this option.

Example

```
.SelectPreviousObject -1
```

The above example selects the previous object in the group.

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

UnSelectAll (DRAW)

.UnSelectAll

This command deselects all objects.

Example

```
.UnSelectAll
```

The above example deselects all selected object(s).

{button ,AL(`OVR1 Object selection commands;',0,"Defaultoverview",)} **Related Topics**

Object creation commands

AddBezierPoint (DRAW)

.AddBezierPoint .X = *long*, .Y = *long*, .Constrain = *boolean*, .Cusp = *boolean*

This command creates the second point of a bezier segment created by using the .BeginDrawBezier command. The segment itself is not added until the .EndDrawBezier command is called.

Parameter	Description
.X	Lets you specify the X-coordinate of the point in tenths of a micron, relative to the center of the page.
.Y	Lets you specify the Y-coordinate of the point in tenths of a micron, relative to the center of the page.
.Constrain	Set to TRUE (-1) to use the constrain angle when positioning the point.
.Cusp	Set to TRUE (-1) to make the new node cusped. Set to FALSE (0) to make it symmetrical (except for end nodes).

Example

```
.InitBezierTool
.BeginDrawBezier -3085992, 163280, FALSE
.MoveBezierControl -1649128, 1142960, FALSE
.AddBezierPoint 146952, -277576, FALSE, FALSE
.MoveBezierControl -326560, -1453192, FALSE
.EndDrawBezier
```

The above example draws a simple bezier curve.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

AddFreehandPoint (DRAW)

.AddFreehandPoint .ConvertToDPCoords = *boolean*, .X = *long*, .Y = *long*

This command adds a point to a freehand curve created by using the .BeginDrawFreehand command. The segment itself is not added until .EndDrawFreehand command is called.

Parameter	Description
.ConvertToDPCoords	Set to TRUE (-1) to convert the X and Y coordinates to physical coordinates on the screen. This parameter should always be set to true unless you know that the coordinates you are using are already physical coodinates on the screen.
.X	Lets you specify the X-coordinate of the point in tenths of a micron, relative to the center of the page.
.Y	Lets you specify the Y-coordinate of the point in tenths of a micron, relative to the center of the page.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

AppendCurveLine (DRAW)

.AppendCurveLine .X1 = *long*, .Y1 = *long*, .X2 = *long*, .Y2 = *long*

This command adds a line segment to an existing curve. An open curve must be selected.

Parameter	Description
.X1	Lets you specify the X-coordinate of the start point in tenths of a micron, relative to the center of the page.
.Y1	Lets you specify the Y-coordinate of the start point in tenths of a micron, relative to the center of the page.
.X2	Lets you specify the X-coordinate of the end point in tenths of a micron, relative to the center of the page.
.Y2	Lets you specify the Y-coordinate of the end point in tenths of a micron, relative to the center of the page.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

BeginDrawBezier (DRAW)

.BeginDrawBezier .X = *long*, .Y = *long*, .Cusp = *boolean*

This command creates the first point of a bezier segment. The second point is added by the .AddBezierPoint command. The segment itself is not added until the .EndDrawBezier command is called. This command must be preceeded by the .InitBezierTool command.

Parameter	Description
.X	Lets you specify the X-coordinate of the point in tenths of a micron, relative to the center of the page.
.Y	Lets you specify the Y-coordinate of the point in tenths of a micron, relative to the center of the page.
.Cusp	Set to TRUE (-1) to make the new node cusped. Set to FALSE (0) to make it symmetrical (except for end nodes).

Note

- The bezier commands include:

  .AddBezierPoint
  .MoveBezierControl

Example

```
.InitBezierTool
.BeginDrawBezier -3085992, 163280, FALSE
.MoveBezierControl -1649128, 1142960, FALSE
.AddBezierPoint 146952, -277576, FALSE, FALSE
.MoveBezierControl -326560, -1453192, FALSE
.EndDrawBezier
```

The above example draws a simple bezier curve.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

BeginDrawCurve (DRAW)

.BeginDrawCurve .X = *long*, .Y = *long*

This command sets the coordinates of the starting node when drawing curves in Freehand mode.

Parameter	Description
.X	Lets you specify the X-coordinate of the starting node of the curve in tenths of a micron, relative to the center of the page.
.Y	Lets you specify the Y-coordinate of the starting node of the curve in tenths of a micron, relative to the center of the page.

Note

- The .BeginDrawCurve command must be followed by a contiguous block of one or more DrawCurve commands, and one .EndDrawCurve command. The DrawCurve commands include:

  .DrawCurveClosePath
  .DrawCurveCurveTo
  .DrawCurveLineTo
  .DrawCurveMoveTo

Example

```
.BeginDrawCurve -500000, 1000000
.DrawCurveLineTo 500000, -500000
.EndDrawCurve
```

The above example demonstrates the DrawCurve commands.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

BeginDrawFreehand (DRAW)

.BeginDrawFreehand .ConvertToDPCoords = *boolean*, .X = *long*, .Y = *long*

This commandcreates the first point of a freehand segment. Additional points are added to the segment by using the .AddFreehandPoint command. The curve itself is not added until the .EndDrawFreehand command is called.

Parameter	Description
.ConvertToDPCoords	Set to TRUE (-1) to convert the X and Y coordinates to physical coordinates on the screen. This parameter should always be set to true unless you know that the coordinates you are using are already physical coodinates on the screen.
.X	Lets you specify the X-coordinate of the point in tenths of a micron, relative to the center of the page.
.Y	Lets you specify the Y-coordinate of the point in tenths of a micron, relative to the center of the page.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CloneObject (DRAW)

.CloneObject .XOffset = *long*, .YOffset = *long*

This command copies the selected object and offsets the copy from the original. Most changes applied to the original object (called the "master") are automatically applied to the copy (called the "clone"). For example, if you change the master's fill, the clone's fill will change as well. If you change the attributes of the clone, the attribute you change will no longer depend on the master's attributes. For example, after you change a clone's fill, its fill will no longer change when you change the master's fill. Likewise, if you stretch a clone, it will no longer stretch when you stretch its master.

Parameter	Description
.XOffset	Lets you specify the horizontal distance to offset the clone object.
.YOffset	Lets you specify the vertical distance to offset the clone object.

Example

```
.CreateEllipse -250000, -500000, 250000, 500000, 0, 0, 0
.CloneObject
```

The above example creates an ellipse, then makes a clone.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

ConvertToCurves (DRAW)

.ConvertToCurves

This command converts the selected polygon, rectangle, ellipse, or text object to a series of curves you can shape with the Shape tool.

Example

```
.CreateRectangle 500000, -750000, -500000, 750000, 0
.ConvertToCurves
```

The above example converts the selected rectangle to a curve object.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CreateAngleDimension (DRAW)

.CreateAngleDimension .X1 = *long*, Y1= *long*, X2 = *long*, Y2 = *long*, X3 = *long*, Y3 = *long*, X4 = *long*, Y4 = *long*, LargeAngle = *boolean,* WitnessExtension = *long*, WitnessGap = *long*, LabelGap = *long*

This command is used to create an angular dimension.

Parameter	Description
.1X1, 1Y1	Lets you specify the apex coordinates.
.1X2, 1Y2	Lets you specify the baseline coordinates.
.1X3, 1Y3	Lets you specify the endline coordinates.
.1X4, 1Y4	Lets you specify the text point.
.LargeAngle	Set to TRUE the angle measured is <= 180 degrees. Set to FALSE the angle measured is >180 degrees.
.WitnessExtension	Lets you specify the length of the extension line that protrudes beyond the dimension line.
.WitnessGap	Lets you specify the distance between the snapped to object and the extension line.
.LabelGap	Lets you specify the amount of space between the text and the dimension line.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CreateCallout (DRAW)

.CreateCallout .X1 = *long*, Y1 = *long*, X2 = *long*, Y2 = *long*, X3 = *long*, Y3 = *long*, Text = *string*

This command is used to create a callout line that points to and labels an object.

Parameter	Description
.X1, Y1	Lets you specify where the first callout segment starts.
.X2, Y2	Lets you specify where the first callout segment ends.
.X3, Y3	Lets you specify the location of the callout text.
.Text	Lets you specify the callout text.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CreateConnector (DRAW)

.CreateConnector .X1 = *long*, Y1 = *long*, X2 = *long*, Y2 = *long*, Placement = *boolean*

This command connects objects with a line. When you move an object that has a connector line attached the connetor line also moves.

Parameter	Description
.X1, Y1	Lets you specify the first coordinate of the line.
.X2, Y2	Lets you specify the second coordinate of the line.
.Placement	Set to TRUE to keep connector line fixed to the nodes that it was originally attached to. Set to FALSE to draw the shortest line between the two objects it connects.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CreateDimension (DRAW)

.CreateDimension .X1 = *long*, Y1 = *long*, X2 = *long*, Y2 = *long*, X3 = *long*, Y3 = *long*, Style = *long*, WitnessExtention = *long*, WitnessGap = *long*, LabelGap = *long*

This command creates a horizontal, vertical or slanted dimension line.

Parameter	Description
.X1, Y1	Lets you specify the start point coordinate of the dimension line.
.X2, Y2	Lets you specify the end point coordinate of the dimension line.
.X3, Y3	Lets you specify the dimension text location.
.Style	0=vertical dimension
	1=horizontal dimension
	2=slanted dimension
.WitnessExtention	Lets you specify the length of the extension line that protrudes beyond the dimension line.
.WitnessGap	Lets you specify the distance between the snapped to object and the extension line.
.LabelGap	Lets you specify the amount of space between the text and the dimension line.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CreateEllipse (DRAW)

.CreateEllipse .Top = *long*, .Left = *long*, .Bottom = *long*, .Right = *long*, .StartAngle = *long*, .EndAngle = *long*, .Arc = *boolean*

This command draws ellipses and circles.

Parameter	Description
.Top	Lets you specify the Y-coordinate of the upper-left corner of the bounding rectangle of the ellipse in tenths of a micron, relative to the center of the page.
.Left	Lets you specify the X-coordinate of the upper-left corner of the bounding rectangle of the ellipse in tenths of a micron, relative to the center of the page.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the bounding rectangle of the ellipse in tenths of a micron, relative to the center of the page.
.Right	Lets you specify the X-coordinate of the lower-right corner of the bounding rectangle of the ellipse in tenths of a micron, relative to the center of the page.
.StartAngle	If .CreateEllipse is used to create an arc, .StartAngle specifies the starting angle in degrees.
.EndAngle	If .CreateEllipse is used to create an arc, .EndAngle specifies the end angle, in degrees.
.Arc	Lets you specify whether to draw the ellipse as a pie or an arc. Set to TRUE (-1) to turn the ellipse into a pie. Set to FALSE (0) to draw the ellipse as an arc.

Note

- You can use the ANGLECONVERT function to specify angle measurements

Example

```
.CreateEllipse -250000, -500000, 250000, 500000, 0, 0, 0
```

The above example creates an ellipse.

```
for count% = 1 to 4
.CreateEllipse 1500000-(250000 * count), -1200000 +( 200000* count), 750000 - ( 200000*
count), -500000+( 200000* count), 0, 0, 0
next count
```

The above example creates 4 ellipses.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CreateGridBoxes (DRAW)

.CreateGridBoxes .Top = *long*, .Left = *long*, .Bottom = *long*, .Right = *long*, .Wide = *long*, .High = *long*

This command draws a grid box.

Parameter	Description
.Top	Lets you specify the Y-coordinate of the upper-left corner of the bounding rectangle of the grid box in tenths of a micron, relative to the center of the page.
.Left	Lets you specify the X-coordinate of the upper-left corner of the bounding rectangle of the grid box in tenths of a micron, relative to the center of the page.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the bounding rectangle of the grid box in tenths of a micron, relative to the center of the page.
.Right	Lets you specify the X-coordinate of the lower-right corner of the bounding rectangle of the grid box in tenths of a micron, relative to the center of the page.
.Wide	Lets you specify the number of cells wide.
.High	Lets you specify the number of cells high.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CreateRectangle (DRAW)

.CreateRectangle .Top = *long*, .Left = *long*, .Bottom = *long*, .Right = *long*[, .CornerRadius = *long*][, .CornerRadius2 = *long*][, .CornerRadius3 = *long*][, .CornerRadius4 = *long*]

This command draws rectangles and squares.

Parameter	Description
.Top	Lets you specify the Y-coordinate of the upper-left corner of the rectangle in tenths of a micron, relative to the center of the page.
.Left	Lets you specify the X-coordinate of the upper-left corner of the rectangle in tenths of a micron, relative to the center of the page.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the rectangle in tenths of a micron, relative to the center of the page.
.Right	Lets you specify the X-coordinate of the lower-right corner of the rectangle in tenths of a micron, relative to the center of the page.
.CornerRadius (optional)	Lets you specify the radius used to create the rounded corners in tenths of a micron.
.CornerRadius2 (optional)	Lets you specify the radius used to create the rounded corners in tenths of a micron.
.CornerRadius3 (optional)	Lets you specify the radius used to create the rounded corners in tenths of a micron.
.CornerRadius4 (optional)	Lets you specify the radius used to create the rounded corners in tenths of a micron.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
```

The above example creates a rectangle.

```
FOR count% = 1 TO 8
.CreateRectangle 1500000-(250000 * count), -1200000 +( 200000* count), 750000 - ( 200000*
count), -500000+( 200000* count), 0
NEXT count
```

The above example creates 8 rectangles.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CreateSpiral (DRAW)

.CreateSpiral .Top = *long*, .Left = *long*, .Bottom = *long*, .Right = *long*, .NumRevolutions = *long*, .SpiralType = *long,* .GrowthRate = *long*

This command draws a spiral.

Parameter	Description
.Top	Lets you specify the Y-coordinate of the upper-left corner of the bounding rectangle of the spiral in tenths of a micron, relative to the center of the page.
.Left	Lets you specify the X-coordinate of the upper-left corner of the bounding rectangle of the spiral in tenths of a micron, relative to the center of the page.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the bounding rectangle of the spiral in tenths of a micron, relative to the center of the page.
.NumRevolutions	Lets you specify the number of revolutions in the spiral.
.SpiralType	Lets you specify the type of spiral. 0 = Symmetrical 1 = Logarithmic
.GrowthRate	Lets you specify the growth rate of the spiral.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

CreateSymPolygon (DRAW)

.CreateSymPolygon .Top = *long*, .Left = *long*, .Bottom = *long*, .Right = *long*, .Sides = *long*, .Subpaths = *long*, .Complexity = *long*, .Star = *boolean*, .StarComplexity = *long*, .MaxComplexity = *long*

This command creates a polygon or star.

Parameter	Description
.Top	Lets you specify the coordinate of the top of the shape in tenths of a micron, relative to the center of the page.
.Left	Lets you specify the coordinate of the left of the shape in tenths of a micron, relative to the center of the page.
.Bottom	Lets you specify the coordinate of the bottom of the shape in tenths of a micron, relative to the center of the page.
.Right	Lets you specify the coordinate of the right of the shape in tenths of a micron, relative to the center of the page.
.Sides	Lets you specify the number of sides (from 3 to 500).
.Subpaths	Lets you specify the number of subpaths in a polygon. If both the number of subpaths and the complexity are set to 1 then the polygon will be a simple polygon. If either of these values are greater than 1 then the polygon becomes a star. The relationship between the number of subpaths and the complexity is represented by the Star/Polygon button and the Sharpness slider on the Polygon Property Bar. Appropriate values for each of these parameters change depending on the number of sides of the polygon.
.Complexity	Lets you specify the complexity of a polygon. If both the number of subpaths and the complexity are set to 1 then the polygon will be a simple polygon. If either of these values are greater than 1 then the polygon becomes a star. The relationship between the number of subpaths and the complexity is represented by the Star/Polygon button and the Sharpness slider on the Polygon Property Bar. Appropriate values for each of these parameters change depending on the number of sides of the polygon. If there is only 1 subpath and the complexity is set to 2, then you will always create a simple star. For more complex stars, we recommend you experiment with the recorder to determine the appropriate values.
.Star	Set to TRUE (-1) to enable the StarComplexity parameter. Set to FALSE (0) to ignore this parameter. This parameter must be TRUE if you want to create a star-shaped polygon.
.StarComplexity	Lets you specify the distance that the start node is from the center of the shape. If this parameter is set to 0, the start node will remain at the outer edge of the shape. The closer this parameter's value is to the .lMaxComplexity value, the closer the start node is to the center of the shape. This parameter is equivalent to the Sharpness slider for polygons (as opposed to stars).
.MaxComplexity	Lets you specify the maximum value for star complexity.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

DeleteObject (DRAW)

.DeleteObject

This command deletes selected objects.

Example

```
.CreateEllipse -250000, -500000, 250000, 500000, 0, 0, 0
.CreateRectangle 750000, -750000, 0, 0, 0
.DeleteObject
```

The above example deletes the selected object. Since the rectangle is the last object created, it is selected and gets deleted.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

DistributeObjects (DRAW)

.DistributeObjects .HorizontalDistribution = *long*, .VerticalDistribution = *long*, .ObjectOrPageExtents = *long*

This command distributes selected objects.

Parameter	Description
.HorizontalDistribution	Lets you specify the type of horizontal distribution. 0 = None 1 = Right edges of object 2 = Left edges of object 3 = Center edges of object 4 = Space between objects
.VerticalDistribution	Lets you specify the type of vertical distribution. 0 = None 1 = Top edges of object 2 = Bottom edges of object 3 = Center edges of object 4 = Space between objects
.ObjectOrPageExtents	Lets you specify the type of distribution. 0 = Extent of Selection 1 = Extent of Page

Example

```
.SelectAllObjects
.DistributeObjects 3, 3, 1
```

The above example distributes the selected objects to the center of the page.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

DrawCurveClosePath (DRAW)

.DrawCurveClosePath

This command closes the path on the last node when drawing curves in Freehand mode.

Note

- The .DrawCurveClosePath command must be in a contiguous block of one or more DrawCurve commands. The first DrawCurve command in the block must be preceded by the .BeginDrawCurve command, and the last must be followed by the .EndDrawCurve command. The DrawCurve commands include:

  .DrawCurveClosePath
  .DrawCurveCurveTo
  .DrawCurveLineTo
  .DrawCurveMoveTo

Example

```
.BeginDrawCurve -500000, 1000000
.DrawCurveCurveTo 500000, 500000, 1000000 ,-500000, -500000, -500000
.DrawCurveClosePath
.EndDrawCurve
```

The above example draws an object in the shape of an uppercase "D".

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

DrawCurveCurveTo (DRAW)

.DrawCurveCurveTo .X1 = *long*, .Y1 = *long*, .X2 = *long*, .Y2 = *long*, .XEnd = *long*, .YEnd = *long*

This command sets a node in a curve drawn in Freehand mode.

Parameter	Description
.X1	Lets you specify the X-coordinate for a control point in tenths of a micron, relative to the center of the page. The control point is used with the node that was created in a preceding BeginDrawCurve or DrawCurveCurveTo command.
.Y1	Lets you specify the Y-coordinate for a control point in tenths of a micron, relative to the center of the page. The control point is used with the node that was created in a preceding BeginDrawCurve or DrawCurveCurveTo command.
.X2	Lets you specify the X-coordinate for a control point in tenths of a micron, relative to the center of the page. The control point is used with the node that is specified in the current DrawCurveCurveTo command.
.Y2	Lets you specify the Y-coordinate for a control point in tenths of a micron, relative to the center of the page. The control point is used with the node that is specified in the current DrawCurveCurveTo command.
.XEnd	Lets you specify the X-coordinate of a node of the curve in tenths of a micron, relative to the center of the page.
.YEnd	Lets you specify the X-coordinate of a node of the curve in tenths of a micron, relative to the center of the page.

Note

- The .DrawCurveCurveTo command must be in a contiguous block of one or more DrawCurve commands. The first DrawCurve command in the block must be preceded by the .BeginDrawCurve command, and the last must be followed by the .EndDrawCurve command. The DrawCurve commands include:

  .DrawCurveClosePath
  .DrawCurveCurveTo
  .DrawCurveLineTo
  .DrawCurveMoveTo

Example

```
.BeginDrawCurve -500000, 1000000
.DrawCurveCurveTo 500000, 500000, 1000000 ,-500000, -500000, -500000
.DrawCurveCurveTo 600000, 600000, 1100000 ,-600000, -600000, -600000
.EndDrawCurve
```

The above example draws a curve.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

DrawCurveLineTo (DRAW)

.DrawCurveLineTo .X = *long*, .Y = *long*

This command sets the coordinates when drawing continuous curves in Freehand mode.

Parameter	Description
.X	Lets you specify the X-coordinate of the next node of the curve in tenths of a micron, relative to the center of the page.
.Y	Lets you specify the Y-coordinate of the next node of the curve in tenths of a micron, relative to the center of the page.

Note

- The .DrawCurveLineTo command must be in a contiguous block of one or more DrawCurve commands. The first DrawCurve command in the block must be preceded by the .BeginDrawCurve command, and the last must be followed by the .EndDrawCurve command. The DrawCurve commands include:

  .DrawCurveClosePath
  .DrawCurveCurveTo
  .DrawCurveLineTo
  .DrawCurveMoveTo

Example

```
.BeginDrawCurve -500000, 1000000
.DrawCurveLineTo 500000, -500000
.EndDrawCurve
```

The above example demonstrates the DrawCurve commands.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

DrawCurveMoveTo (DRAW)

.DrawCurveMoveTo .X = *long*, .Y = *long*

This command sets the coordinates when drawing non-continuous curves in Freehand mode.

Parameter	Description
.X	Lets you specify the X-coordinate of the point to move to without drawing in tenths of a micron, relative to the center of the page.
.Y	Lets you specify the Y-coordinate of the point to move to without drawing in tenths of a micron, relative to the center of the page.

Note

- The .DrawCurveMoveTo command must be in a contiguous block of one or more DrawCurve commands. The first DrawCurve command in the block must be preceded by the .BeginDrawCurve command, and the last must be followed by the .EndDrawCurve command. The DrawCurve commands include:

  .DrawCurveClosePath
  .DrawCurveCurveTo
  .DrawCurveLineTo
  .DrawCurveMoveTo

Example

```
.BeginDrawCurve -500000, 1000000
.DrawCurveLineTo 500000, -500000
.DrawCurveMoveTo -500000, -500000
.DrawCurveLineTo 500000, 1000000
.EndDrawCurve
```

The above example demonstrates the DrawCurve commands.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

DuplicateObject (DRAW)

.DuplicateObject .XOffset = *long*, .YOffset = *long*

This command adds a copy of the selected object(s) to the current drawing. By default, the copy is placed on top of the original, offset up and to the right. It is also selected automatically.

Parameter	Description
.XOffset	Lets you specify the horizontal distance to offset the duplicate object.
.YOffset	Lets you specify the vertical distance to offset the duplicate object.

Example

```
.CreateRectangle 1082025, -333882, 272052, 500823, 0, 0, 0, 0
.StoreColor 5002, 100, 0, 100, 0, 0, 0, 0, 0
.ApplyUniformFillColor
.DuplicateObject
```

The above example creates a rectangle and fills it with a color, then duplicates the object.

Note

- The created object is created on top of the object you duplicated.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

EndDrawBezier (DRAW)

.EndDrawBezier

This command ends a set of bezier creation commands that began with the .BeginDrawBezier command.

Note

- The bezier commands include:

  .AddBezierPoint
  .MoveBezierControl

Example

```
.InitBezierTool
.BeginDrawBezier -3085992, 163280, FALSE
.MoveBezierControl -1649128, 1142960, FALSE
.AddBezierPoint 146952, -277576, FALSE, FALSE
.MoveBezierControl -326560, -1453192, FALSE
.EndDrawBezier
```

The above example draws a simple bezier curve.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

EndDrawCurve (DRAW)

.EndDrawCurve

This command ends a set of curve creation commands that began with the .BeginDrawCurve command.

Note

- The DrawCurve commands include:

  .DrawCurveClosePath
  .DrawCurveCurveTo
  .DrawCurveLineTo
  .DrawCurveMoveTo

Example

```
.BeginDrawCurve -500000, 1000000
.DrawCurveLineTo 500000, -500000
.EndDrawCurve
```

The above example demonstrates the DrawCurve commands.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

EndDrawFreehand (DRAW)

.EndDrawFreehand .StraightTightness = *long*, .CornerTightness = *long*, .CornerThreshold = *long*, .SnapTightness = *long*

This command ends a set of frehand drawing commands that began with the .BeginDrawCurve command.

Parameter	Description
.StraightTightness	Sets the amount the curve can vary from a straight path and still be treated as straight. The higher the value, the less accurate the line needs to be. The valid range is from 1 to 10 pixels.
.CurveTightness	Determines how closely the curve will match the position of each point. The lower the number, the more accurate the match. The valid range is from 1 to 10 pixels.
.CornerThreshold	Sets the limit at which each corner node is cusped (as opposed to smooth). A node is more likely to be cusped if the value is lower. The valid range is from 1 to 10 pixels.
.SnapTightness	Determines how close two end nodes must be to join automatically. The valid range is from 1 to 10 pixels.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetEllipseClockwise (DRAW)

.GetEllipseClockwise

This function returns false of the ellipse is counterclockwise.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetEllipseEndAngle (DRAW)

.GetEllipseEndAngle

This function returns the end angle of the ellipse.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetEllipseStartAngle (DRAW)

.GetEllipseStartAngle

This function returns the start angle of the ellipse.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetEllipseType (DRAW)

.GetEllipseType

This function returns the type of ellipse. (0 = Ellipse, 1 = Arc, 2 = Pie)

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetObjectCount (DRAW)

.GetObjectCount .Selection = *boolean*, .Grouped = *boolean*

This function counts the number of objects.

Parameter	Description
.Selection	Set to TRUE (-1) counts objects in the current selection.
.Grouped	Set to TRUE (-1) counts groups as one object.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetObjectID (DRAW)

.GetObjectID .Index = *long*, .Selection = *boolean,* .Grouped = *boolean*

This function returns the ID of the Index-th object.

Parameter	Description
.Index	Lets you specify the object ID. 0 J GetObjectCount-1
.Selection	Set to TRUE (-1) counts objects in the current selection.
.Grouped	Set to TRUE (-1) counts groups as one object.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetObjectsCDRStaticID (DRAW)

ReturnValue& = .GetObjectsCDRStaticID()

This function returns the CDRStaticID of the selected object. If more than one object is selected, the function returns the CDRStaticID of the last selected object.

Return Value

Returns the following value:

- the CDRStaticID of the selected object

Note

- Every object you create has a unique CDRStaticID in a document.

Example

```
.CreateRectangle 750000, -600000, 250000, -100000, 0
IDRect& = .GetObjectsCDRStaticID()
.SelectObjectOfCDRStaticID IDRect&
```

The above example demonstrates object selection using the object's CDRStaticID.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetPolygonSharpness (DRAW)

.GetPolygonSharpness

This function returns the sharpness value in a polygon.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetPolygonSides (DRAW)

.GetPolygonSides

This function returns the number of sides in a polygon.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetPolygonType (DRAW)

.GetPolygonType

This function returns the type of polygon. (0 = Polyong, 1 = Star, 2 = Polygon as star)

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

GetRectangleRadius (DRAW)

.GetRectangleRadius .Radius1 = *double*, .Radius2 = *double*, .Radius3 = *double*, .Radius 4 = *double*

This function returns the radius for each of a rectangle's corners.

Parameter	Description
.Radius1	Lets you specify the upper left radius of the rectangle.
.Radius2	Lets you specify the upper right radius of the rectangle.
.Radius3	Lets you specify the lower right radius of the rectangle.
.Radius4	Lets you specify the lower right radius of the rectangle.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

InitBezierTool (DRAW)

.InitBezierTool

This command must preceed the .BeginDrawBezier command.

Example

```
.InitBezierTool
.BeginDrawBezier -3085992, 163280, FALSE
.MoveBezierControl -1649128, 1142960, FALSE
.AddBezierPoint 146952, -277576, FALSE, FALSE
.MoveBezierControl -326560, -1453192, FALSE
.EndDrawBezier
```

The above example draws a simple bezier curve.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

MoveBezierControl (DRAW)

.MoveBezierControl .X = *long*, .Y = *long*, .Constrain = *boolean*

This command controls the position of bezier node control points created with the .BeginDrawBezier and .AddBezierPoint commands.

Parameter	Description
.X	Lets you specify the X-coordinate of the control point's position in tenths of a micron, relative to the center of the page.
.Y	Lets you specify the Y-coordinate of the control point's position in tenths of a micron, relative to the center of the page.
.Constrain	Set to TRUE (-1) to use the constrain angle when positioning the control points.

Example

```
.InitBezierTool
.BeginDrawBezier -3085992, 163280, FALSE
.MoveBezierControl -1649128, 1142960, FALSE
.AddBezierPoint 146952, -277576, FALSE, FALSE
.MoveBezierControl -326560, -1453192, FALSE
.EndDrawBezier
```

The above example draws a simple bezier curve.

{button ,AL(`OVR1 Object creation commands;',0,"Defaultoverview",)} **Related Topics**

Node editing commands

AddNode (DRAW)

.AddNode .X = *long* , .Y = *long*

This command adds a node to the current curve at a specified location.

Parameter	Description
.X	Lets you specify the X coordinate.
.Y	Lets you specify the Y coordinate.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

BeginEditObject (DRAW)

.BeginEditObject

This command initializes a block of node editing commands that includes one or more instances of the .EditObjectCommand and ends with the .EndEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

CloseCurve (DRAW)

.CloseCurve

This command closes the selected open path.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

DeleteNode (DRAW)

.DeleteNode

This command deletes the current node.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

EditAngleDimensionLabel (DRAW)

.EditAngleDimensionLabel .Dynamic = *boolean*, Precision = *short*, Units = *short*, ShowUnits = *short*, Prefix = *string*, Suffix = *string*

This command lets you change the properties of an angle dimension label.

Parameter	Description
.Dynamic	Set to TRUE the dimension text changes automatically when the objects is stretched or skewed.
.Precision	0=0
	1=0.0
	2=0.00
	3=0.000
	4=0.0000
	5=0.00000
	6=0.000000
	7=0.0000000
	8=0.00000000
	9=0.000000000
	10=0.0000000000
.Units	0=degrees
	1=radians
	2=gradians
.ShowUnits	Set to TRUE displays the units beside the dimension text.
.Prefix	Lets you specify that the prefix is attached to the dimension text.
.Suffix	Lets you specify that the suffix is attached to the dimension text.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

EditDimensionLabel (DRAW)

.EditDimensionLabel .Placement = *short*, Horizontal = *boolean*, Center = *boolean*, Dynamic = *boolean*, Style = *short*, Precision = *short*, Units = *short*, ShowUnits = *boolean*, Prefix = *string*, Suffix = *string*

This command lets you change the properties of a vertical, horizontal or slanted dimension label.

Parameter	Description
.Placement	0=Places text above the dimension line.
	1=Places text within the dimension line.
	2=Places text below the dimension line.
.Horizontal	Set to TRUE places the text horizontally.
.Center	Set to TRUE centers the text in the dimension line.
.Dynamic	Set to TRUE the dimension text changes automatically when the objects is stretched or skewed.
.Style	0=Decimal
	1=Fractional
	2=U.S. Engineering
	3=U.S. Architectural
.Precision	0=0 or 0
	1=0.0 or 01/2 or 0'-0" or 0'-0" depending on .sStyle
	2=0.00 or 01/4 or 0'-0.0" or 0'-01/4" depending on .sStyle
	3=0.000 or 01/8 or 0'-0.00" or 0'-01/8" depending on .sStyle
	4=0.0000 or 01/16 or 0'-0.000" or 0'-01/16" depending on .sStyle
	5=0.00000 or 01/32 or 0'-0.0000" or 0'-01/32" depending on .sStyle
	6=0.000000 or 01/64 or 0'-0.00000" or 0'-01/64" depending on .sStyle
	7=0.0000000 or 01/128 or 0'-0.000000" or 0'-01/128" depending on .sStyle
	8=0.00000000 or 01/256 or 0'-0.0000000" or 0'-01/256" depending on .sStyle
	9=0.000000000 or 01/512 or 0'-0.00000000" or 0'-01/512" depending on .sStyle
	10=0.0000000000 or 01/1024 or 0'-0.000000000" or 0'-01/1024" depending on .sStyle
.Units	0="
	1=in
	2=inches
	3='
	4=ft
	5=mi
	6=miles
	7=yds
	8=yards
	9=m
	10=meters
	11=km
	12=kilometers
	13=cm
	14=centimeters
	15=mm
	16=millimeters
	17=picas
	18=points
	19=ciceros
	20=didots
.ShowUnits	Set to TRUE displays the units beside the dimension text.

.Prefix	Lets you specify that the prefix is attached to the dimension text.
.Suffix	Lets you specify that the suffix is attached to the dimension text.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

EditObjectCommand (DRAW)

.EditObjectCommand .Cmd = *long*, .X = *long*, .Y = *long*, .Key = *long*, .AddToSelection = *boolean*

This command lets you change the shape of objects by editing their nodes. This command must be part of a block of commands that begins with the .BeginEditObject command and ends with the .EndEditObject command.

Parameter	Description
.Cmd	0 = This value is equivalent to releasing the mouse button. Use with the .lX parameter and the .lY parameter. 1 = This value is equivalent to pressing one of the nudge or super-nudge keys. Use the .lKey parameter to indicate which nudge key is used. 2 = This value is equivalent to pressing a key on the keyboard. Use the .lKey parameter to indicate which key is used. 3 = This value is equivalent to pressing down the mouse button. Use with the .lX parameter and the .lY parameter. 4 = This value is equivalent to the add node option. 5 = This value is equivalent to the delete node option. 6 = This value is equivalent to the join nodes option. 7 = This value is equivalent to the break path option. 8 = This value makes a node cusped. 9 = This value makes a node smooth. 10 = This value makes a segment straight. 11 = This value makes a segment curved. 12 = This value makes a node symmetrical. 13 = This value is equivalent to the auto-reduce nodes option. 14 = This value is equivalent to the extract subpath option. 16 = This value toggles the elastic option.
.X	Lets you specify the X-coordinate of a node in tenths of a micron, relative to the center of the page.
.Y	Lets you specify the Y-coordinate of a node in tenths of a micron, relative to the center of the page.
.Key	If .lCmd = 1 0 = nudge left 1 = nudge right 2 = nudge up 3 = nudge down 4 = super-nudge left 5 = super-nudge right 6 = super-nudge up 7 = super-nudge down If .lCmd = 2 0 = HOME 1 = END 2 = TAB 3 = ESC
.AddToSelection	Set to TRUE (-1) to select multiple nodes (equivalent to pressing SHIFT).

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

EndEditObject (DRAW)

.EndEditObject

This command ends a block of node editing commands that includes one or more instances of the .EditObjectCommand and begins with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetCurveClose (DRAW)

.GetCurveClose

This command returns true if the curve is closed.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetCurveFirstNodePosition (DRAW)

.GetCurveFirstNodePosition .X = *long* , .Y = *long*

This command returns the coordinates of the first node in the selected curve.

Parameter	Description
.X	Lets you specify the X coordinate.
.Y	Lets you specify the Y coordinate.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetCurveIthNodePosition (DRAW)

.GetCurveIthNodePosition .X = *long* , .Y = *long*

This command returns the coordinates of the I-th node in the selected curve.

Parameter	Description
.X	Lets you specify the I-th coordinate.
.Y	Lets you specify the I-th coordinate.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetCurveLastNodePosition (DRAW)

.GetCurveLastNodePosition .X = *long* , .Y = *long*

This command returns the coordinates of the last node in the selected curve.

Parameter	Description
.X	Lets you specify the X coordinate.
.Y	Lets you specify the Y coordinate.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetCurveLength (DRAW)

This command returns the curve length.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetCurveNodeCount (DRAW)

This command returns the number of nodes of the selected curve.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetCurveSubpathCount (DRAW)

This command returns the number of subpaths of the selected curve.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetNodeIndex (DRAW)

.GetNodeIndex .Position= *long*

This command returns the index of the current node.

Parameter	Description
.Position	Returns the position index of the current node.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetNodePosition (DRAW)

.GetNodePosition .X= *long,* .Y = *long,* .Position= *long*

This command returns the index of the current node.

Parameter	Description
.X	Lets you specify the X coordinate of the current node.
.Y	Lets you specify the Y coordinate of the current node.
.Position	Lets you specify the position of the current node.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetNodeSelectedCount (DRAW)

This command returns the number of nodes in the current selection.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetNodeType (DRAW)

Return Value = **.GetNodeType**

This command returns the type of node in the current selection.

Return Value

Returns of the following values:

- 0 Cusp
- 1 Smooth
- 2 Symmetrical

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetSegmentLength (DRAW)

.GetSegmentLength

This command returns the segment length for the current selection.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

GetSegmentType (DRAW)

Return Value = .GetSegmentType

This command returns the segment type for the current selection.

Return Value

Returns one of the following values:

- 0 Line
- 1 Curve

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

MoveNode (DRAW)

.MoveNode .DeltaX = *long,* .DeltaY = *long*

This command moves a node a specifies distance.

Parameter	Description
.DeltaX	Lets you specify the amount to move the node horizontally.
.DeltaY	Lets you specify the amount to move the node vertically.

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

SetCornerRoundness (DRAW)

.SetCornerRoundess .Roundness = *long* , .Roundness2 = *long,* .Roundness3 = *long,* .Roundness4 = *long*

This command sets the corner roundness of the selected object.

Parameter	Description
.Roundness	Lets you specify the roundness of the corners in tenths of a percent.
.Roundness2	Lets you specify the roundness of the corners in tenths of a percent.
.Roundness3	Lets you specify the roundness of the corners in tenths of a percent.
.Roundness4	Lets you specify the roundness of the corners in tenths of a percent.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

SetEllipseProperties (DRAW)

.SetEllipseProperties .Arc = *boolean*, .StartAngle = *long*, .EndAngle = *long*

This command sets the ellipse properties.

Parameter	Description
.Arc	Set to TRUE = Arc. Set to FALSE = Ellipse/Pie
.StartAngle	Lets you specify the start angle in millionths of a degree.
.EndAngle	Lets you specify the end angle in millionths of a degree.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

SetNodeType (DRAW)

.SetNodeType .Type = *long*

This command sets the type of node.

Parameter	Description
.Type	Lets you specify the type of node: 0 = Cusp 1 = Smooth 2 = Symmetrical

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

SetPolygonProperties (DRAW)

.SetPolygonProperties .Star = *boolean,* .Points = *long,* .Sharpness = *long*

This command sets the polygon properties of the selected object.

Parameter	Description
.Star	If set to TRUE = Star. If set to FALSE = Polygon.
.Points	Lets you specify the number of points/edges in a polygon.
.Sharpness	Speficies the sharpness of the star.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

SetSegmentType (DRAW)

.SetSegmentType .Type = *long*

This command sets the type of segment.

Parameter	Description
.Type	Lets you specify the type of node: 0 = Line 1 = Curve

Note

- You must precede this command with the .BeginEditObject command.

{button ,AL(`OVR1 Node editing commands;',0,"Defaultoverview",)} **Related Topics**

Symbols commands

DropSymbol (DRAW)

.DropSymbol .SymbolLibrary = *string*, .SymbolNumber = *long*, .Tile = *boolean*, .XPosOrGridSize = *long*, .YPosOrGridSize = *long*, .ProportionalSizing = *boolean*, .SymbolSize = *long*

This command positions the specified symbol at the defined position or the specified grid position.

Parameter	Description
.SymbolLibrary	Lets you specify the name of the Symbol Library. Refer to the Symbols dialog box for more details.
.SymbolNumber	Lets you specify the Symbol Index Number, which identifies the selected symbol. Refer to the Symbols dialog box for more details.
.Tile	Set to TRUE (-1) to create a pattern from the selected symbol that fills the page. Set to FALSE (0) to disable this option. Note that the tiled symbols are clones of the upper left symbol.
.XPosOrGridSize	Lets you specify the X-coordinate or grid position at which to place the symbol, in tenths of a micron.
.YPosOrGridSize	Lets you specify the Y-coordinate or grid position at which to place the symbol, in tenths of a micron.
.ProportionalSizing	Set to TRUE (-1) to enable proportional sizing of the symbol. Set to FALSE (0) to disable this option.
.SymbolSize	Lets you specify the size of the symbol in tenths of a micron. The symbol can be resized after it's been added to your drawing.

Example

```
.DropSymbol "Animals 1", 42, 0, 0, 0, 0, 1000000
```

The above example places a kangaroo symbol in the center of the page.

Arrange commands

AlignObjects (DRAW)

.AlignObjects .HorizontalAlignment = *long*, .VerticalAlignment = *long*

This command aligns the selected objects according to the last selected object.

Parameter	Description
.HorizontalAlignment	Lets you specify the type of horizontal alignment. 0 = None 1 = Right 2 = Left 3 = Center
.VerticalAlignment	Lets you specify the type of vertical alignment. 0 = None 1 = Top 2 = Bottom 3 = Center

Example

```
.SelectAllObjects
.AlignObjects 2, 0
```

The above example aligns the selected objects according to the last selected object.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

AlignToCenterOfPage (DRAW)

.AlignToCenterOfPage .HorizontalAlignment = *long*, .VerticalAlignment = *long*

This command aligns selected objects to the center of the page.

Parameter	Description
.HorizontalAlignment	Lets you specify the type of horizontal alignment. 0 = None 1 = Right 2 = Left 3 = Center
.VerticalAlignment	Lets you specify the type of vertical alignment. 0 = None 1 = Top 2 = Bottom 3 = Center

Example

```
.SelectAllObjects
.AlignToCenterOfPage 0, 3
```

The above example vertically aligns all objects to the center of the page.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

AlignToGrid (DRAW)

.AlignToGrid .HorizontalAlignment = *long*, .VerticalAlignment = *long*

This command aligns the selected objects to the gridpoint nearest to the edge of the selection.

Parameter	Description
.HorizontalAlignment	Lets you specify the type of horizontal alignment. 0 = None 1 = Right 2 = Left 3 = Center
.VerticalAlignment	Lets you specify the type of vertical alignment. 0 = None 1 = Top 2 = Bottom 3 = Center

Example

```
.SelectAllObjects
.AlignToGrid 1, 0
```

The above example horizontally aligns all objects to a gridpoint, nearest to the left edge of the selection.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

BreakApart (DRAW)

.BreakApart

This command converts an object made up of multiple subpaths into individual curve objects.

Example

```
.BreakApart
```

The above example breaks apart the selected object into individual curve objects.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

Combine (DRAW)

.Combine

This command combines the selected curve or line segments into a single object. If you use Combine on rectangles, ellipses, polygons, or text, CorelDRAW converts them to curves before converting them into a single curve object. However, when text is combined with other text it is not converted to curves; it is converted to larger blocks of text.

Example

```
for count% = 1 to 4
.CreateEllipse 1500000-(250000 * count), -1200000 +( 200000* count), 750000 - ( 200000*
count), -500000+( 200000* count), 0, 0, 0
next count
.SelectAllObjects
.Combine
.ApplyUniformFillColor 2, 0, 255, 0, 0
```

The above example creates 4 ellipses, then combines them before applying a fill.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

ConvertOutlineToObject (DRAW)

.ConvertOutlineToObject

This command converts an object's outline into an object.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

CreateGuidelineUsingAngle (DRAW)

.CreateGuidelineUsingAngle .IntersectPointX = *long*, .IntersectPointY = *long*, .Angle = *long*, .Locked = *boolean*

This command creates a guideline at a specific location using an intersection point and an angle to specify where to put the new guideline.

Parameter	Description
.IntersectPointX	Lets you specify the X-coordinate of the point in tenths of a micron, relative to the center of the page.
.IntersectPointY	Lets you specify the Y-coordinate of the point in tenths of a micron, relative to the center of the page.
.Angle	Lets you specify the angle of the guideline in millionths of a degree (e.g., 5000000 = 5 degrees).
.Locked	Set to TRUE (-1) to lock the guideline. Set to FALSE (0) to leave the guideline unlocked.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

CreateGuidelineUsingTwoPoints (DRAW)

.CreateGuidelineUsingTwoPoints .Point1X = *long*, .Point1Y = *long*, .Point2X = *long*, .Point2Y = *long*, .Locked = *boolean*

This command creates a guideline at a specific location using two sets of coordinates to place the guidleine.

Parameter	Description
.Point1X	Lets you specify the X-coordinate of the first point in tenths of a micron, relative to the center of the page.
.Point1Y	Lets you specify the Y-coordinate of the first point in tenths of a micron, relative to the center of the page.
.Point2X	Lets you specify the X-coordinate of the second point in tenths of a micron, relative to the center of the page.
.Point2Y	Lets you specify the Y-coordinate of the second point in tenths of a micron, relative to the center of the page.
.Locked	Set to TRUE (-1) to lock the guideline. Set to FALSE (0) to leave the guideline unlocked.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

DeleteGuidelineByIndex (DRAW)

.DeleteGuidelineByIndex .Index = *long*

This command deletes a specific guideline.

Parameter	Description
.Index	Lets you specify the guidline ID. Use the .GetNumberOfGuidelines command to get a guideline's ID.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

DeleteGuidelineUsingAngle (DRAW)

.DeleteGuidelineUsingAngle .IntersectX = *long*, .IntersectY = *long*, .Angle = *long*

This command deletes a specific guideline based on a set of coordinates and an angle.

Parameter	Description
.IntersectPointX	Lets you specify the X-coordinate of the point in tenths of a micron, relative to the center of the page.
.IntersectPointY	Lets you specify the Y-coordinate of the point in tenths of a micron, relative to the center of the page.
.Angle	Lets you specify the angle of the guideline in millionths of a degree (e.g., 5000000 = 5 degrees).

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

DeleteGuidelineUsingTwoPoints (DRAW)

.DeleteGuidelineUsingTwoPoints .Point1X = *long*, .Point1Y = *long*, .Point2X = *long*, .Point2Y = *long*

This command deletes a specific guideline using two sets of coordinates.

Parameter	Description
.Point1X	Lets you specify the X-coordinate of the first point in tenths of a micron, relative to the center of the page.
.Point1Y	Lets you specify the Y-coordinate of the first point in tenths of a micron, relative to the center of the page.
.Point2X	Lets you specify the X-coordinate of the second point in tenths of a micron, relative to the center of the page.
.Point2Y	Lets you specify the Y-coordinate of the second point in tenths of a micron, relative to the center of the page.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

GetGuidelineInformation (DRAW)

.GetGuidelineInformation .Index = *long*, .Point1X = *long**, .Point1Y = *long**, .Point2X = *long**, .Point2Y = *long**, .InterceptX = *long**, .InterceptY = *long**, .Angle = *long**, .Locked = *boolean**

This command retrieves all the information available about a specific guideline. The .plPoint parameters are equivalent to the coordinates used in the .CreateGuidlineUsingTwoPoints command, and the .plIntercept and .plAngle parameters are equivalent to the parameters used in the .CreateGuidelineUsingAngle command.

Parameter	Description
.Index	Lets you specify the guidline ID. Use the .GetNumberOfGuidelines command to get a guideline's ID.
.Point1X	Returns the X-coordinate of the first point in tenths of a micron, relative to the center of the page.
.Point1Y	Returns the Y-coordinate of the first point in tenths of a micron, relative to the center of the page.
.Point2X	Returns the X-coordinate of the second point in tenths of a micron, relative to the center of the page.
.Point2Y	Returns the Y-coordinate of the second point in tenths of a micron, relative to the center of the page.
.InterceptX	Returns the X-coordinate of the point that relates to .plAngle in tenths of a micron, relative to the center of the page.
.InterceptY	Returns the Y-coordinate of the point that relates to .plAngle in tenths of a micron, relative to the center of the page.
.Angle	Returns the angle of the guideline in millionths of a degree (e.g., 5000000 = 5 degrees).
.Locked	Returns TRUE (-1) if guidelines are locked. Returns FALSE (0) if the guidelines are unlocked.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

GetNumberOfGuidelines (DRAW)

ReturnValue& = .GetNumberOfGuidelines()

This function return the number of guidelines. You can use this function to determine the index of a guidline. The range of possible guideline index values is 0 to the number of guidelines minus 1.

Return Value

Returns the following value:

- the number of guidelines

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

Group (DRAW)

.Group

This command groups all selected objects together to allow them to be selected and manipulated as a single object.

Example

```
for count% = 1 to 4
.CreateEllipse 1500000-(250000 * count), -1200000 +( 200000* count), 750000 - ( 200000*
count), -500000+( 200000* count), 0, 0, 0
next count
.SelectAllObjects
.Group
.ApplyUniformFillColor 5, 0, 0, 255, 0
```

The above example groups the four ellipses together so that they are treated as one object, and applies a blue uniform fill to all four.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

Intersection (DRAW)

.Intersection .LeaveTarget = *Boolean*, .LeaveModifiers = *Boolean*

This command creates a new object using the area common to two or more overlapping objects. Intersection joins their paths at the points where they intersect. The resulting curve object assumes the fill and outline attributes of the last selected object.

Parameter	Description
.LeaveTarget	Set to TRUE (-1) leaves the target object in your document. Set to FALSE (0) removes the target object from your document.
.LeaveModifiers	Set to TRUE (-1) leaves the modifier object in your document. Set to FALSE (0) removes the modifier object from your document.

Example

```
.SelectAllObjects
.Intersection, -1, -1
```

The above example selects all objects and creates a new object(s) using the area common to overlapping objects. Both the target object and the modifier object are left in the document.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

LockGuidelineByIndex (DRAW)

.LockGuidelineByIndex .Index = *long*

This command locks a specific guideline.

Parameter	Description
.Index	Lets you specify the guidline ID. Use the .GetNumberOfGuidelines command to get a guideline's ID.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

MoveGuidelineUsingAngleByIndex (DRAW)

.MoveGuidelineUsingAngleByIndex .Index = *long*, .InterceptX = *long*, .InterceptY = *long*, .Angle = *long*, .Locked = *boolean*

This command moves a specific guideline to a location using an intersection point and an angle to specify where to put the new guideline.

Parameter	Description
.Index	Lets you specify the guidline ID. Use the .GetNumberOfGuidelines command to get a guideline's ID.
.InterseptX	Lets you specify the X-coordinate of the point in tenths of a micron, relative to the center of the page.
.InterseptY	Lets you specify the Y-coordinate of the point in tenths of a micron, relative to the center of the page.
.Angle	Lets you specify the angle of the guideline in millionths of a degree (e.g., 5000000 = 5 degrees).
.Locked	Set to TRUE (-1) to lock the guideline. Set to FALSE (0) to leave the guideline unlocked.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

MoveGuidelineUsingTwoPointsByIndex (DRAW)

.MoveGuidelineUsingTwoPointsByIndex .Index = *long*, .Point1X = *long*, .Point1Y = *long*, .Point2X = *long*, .Point2Y = *long*, .Locked = *boolean*

This command moves a specific guideline to a location using two sets of coordinates to place the guidleine.

Parameter	Description
.Index	Lets you specify the guidline ID. Use the .GetNumberOfGuidelines command to get a guideline's ID.
.Point1X	Lets you specify the X-coordinate of the first point in tenths of a micron, relative to the center of the page.
.Point1Y	Lets you specify the Y-coordinate of the first point in tenths of a micron, relative to the center of the page.
.Point2X	Lets you specify the X-coordinate of the second point in tenths of a micron, relative to the center of the page.
.Point2Y	Lets you specify the Y-coordinate of the second point in tenths of a micron, relative to the center of the page.
.Locked	Set to TRUE (-1) to lock the guideline. Set to FALSE (0) to leave the guideline unlocked.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

OrderBackOne (DRAW)

.OrderBackOne

This command rearranges the stacking order by moving the selected object back one position.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.ApplyUniformFillColor 5,255,0,0,0
.CreateEllipse -450000, -700000, 450000, 700000, 0, 0, 0
.ApplyUniformFillColor 5,0,0,250,0
.OrderBackOne
```

The above example creates a rectangle and then creates an ellipse on top of the rectangle. The ellipse, still selected, is ordered back one position in the drawing.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

OrderForwardOne (DRAW)

.OrderForwardOne

This command rearranges the stacking order by moving the selected object up one position.

Example

```
.SelectObjectOfCDRStaticID Six&
.OrderForwardOne
```

The above example orders the selected object forward one position.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

OrderReverseOrder (DRAW)

.OrderReverseOrder

This command reverses the stacking order of the selected object(s).

Example

```
.SelectAllObjects
.OrderReverseOrder
```

The above example reverses the order of all the objects.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

OrderToBack (DRAW)

.OrderToBack

This command rearranges the stacking order by moving the selected object to the back of the screen. Areas of the object overlapped by other objects with fills are "knocked out" so that they will not print.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.ApplyUniformFillColor 5,255,0,0,0
.CreateEllipse -450000, -700000, 450000, 700000, 0, 0, 0
.ApplyUniformFillColor 5,0,0,250,0
.OrderToBack
```

The above example creates a rectangle and then creates an ellipse on top of the rectangle. The ellipse, still selected, is ordered to the back of the drawing.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

OrderToFront (DRAW)

.OrderToFront

This command rearranges the stacking order by moving the selected object to the front of the layer.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.ApplyUniformFillColor 5,255,0,0,0
.CreateEllipse -450000, -700000, 450000, 700000, 0, 0, 0
.ApplyUniformFillColor 5,0,0,250,0
.SelectPreviousObject 0
.OrderToFront
```

The above example creates a rectangle and then creates an ellipse on top of the rectangle. The rectangle is then selected and ordered to the front of the drawing.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

RemoveAllGuidelines (DRAW)

.RemoveAllGuidelines

This command removes all guidelines.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

Separate (DRAW)

.Separate

This command separates original objects from intermediate shapes.

Example

```
.Separate
```

The above example separates a combined object into its individual component object(s).

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

Trim (DRAW)

.Trim .LeaveTarget = *Boolean*, .LeaveModifiers = *Boolean*

This command lets you trim selected objects. Trimming two or more overlapping objects reshapes the last object selected. Trimming separates the paths at points where the objects overlap. Initially, the trimmed object may appear no different than it did before trimming. However, closer inspection will show that new nodes appear where the object was trimmed. Move the trimmed objects apart to see the full effect of the trim.

Parameter	Description
.LeaveTarget	Set to TRUE (-1) leaves the target object in your document. Set to FALSE (0) removes the target object from your document.
.LeaveModifiers	Set to TRUE (-1) leaves the modifier object in your document. Set to FALSE (0) removes the modifier object from your document.

Example

```
.SelectAllObjects
.Trim ,-1, -1
```

The above example trims the selected objects. Both the target object and the modifier object are left in the document.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

Ungroup (DRAW)

.Ungroup

This command breaks up the selected group into its individual objects. If you have more than one sublevel of grouping, Ungroup breaks up one level of grouping at a time.

Example

```
.Ungroup
```

The above example breaks up the grouped object into its individual object components.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

UngroupAll (DRAW)

.UngroupAll

This command breaks up all the groups into its individual objects.

Example

```
.UngroupAll
```

The above example breaks up all the grouped objects into its individual object components.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

UnlockGuidelineByIndex (DRAW)

.UnlockGuidelineByIndex .Index = *long*

This command unlocks a specific guideline.

Parameter	Description
.Index	Lets you specify the guidline ID. Use the .GetNumberOfGuidelines command to get a guideline's ID.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

Weld (DRAW)

.Weld . LeaveTarget = *Boolean*, LeaveModifiers = *Boolean*

This command joins overlapping objects at points where their paths intersect. Although not necessarily apparent in editable preview, welding also removes sections of the path between those intersect points. The resulting curve object assumes the fill and outline attributes of the bottom object of the selected group of objects. If you marquee-select the objects, CorelDRAW will outline and fill the welded object with the attributes of the most recently created object.

Parameter	Description
.LeaveTarget	Set to TRUE (-1) leaves the target object in your document. Set to FALSE (0) removes the target object from your document.
.LeaveModifiers	Set to TRUE (-1) leaves the modifier object in your document. Set to FALSE (0) removes the modifier object from your document.

Example

```
.SelectAllObjects
.Weld -1, -1
```

The above example welds the selected object group. Both the target object and the modifier object are left in the document.

{button ,AL(`OVR1 Arrange commands;',0,"Defaultoverview",)} **Related Topics**

Transformation commands

GetPosition (DRAW)

.GetPosition .XPos = *long**, .Ypos = *long**

This function returns the position coordinates of a selected object's reference point. If more than one object is selected, the function returns the position coordinates of the last selected object.

Parameter	Description
.XPos	Returns the X-coordinate of the selected object's reference point in tenths of a micron, relative to the center of the page. An object's default reference point is the upper-left corner.
.YPos	Returns the Y-coordinate of the selected object's reference point in tenths of a micron, relative to the center of the page. An object's default reference point is the upper-left corner.

Example

```
.CreateRectangle 1000000, 750000, 500000, 100005, 0
id& = .GetObjectsCDRStaticID()
.GetPosition XPos&, YPos&
MESSAGE "Horizontal"+STR(XPos&)
MESSAGE "Vertical"+STR(YPos&)
```

The above example creates a rectangle then displays the coordinates of the upper-left corner in message boxes.

```
.CreateRectangle 1000000, 750000, 500000, 100005, 0
id& = .GetObjectsCDRStaticID()
.SetReferencePoint 3
.GetPosition XPos&, YPos&
MESSAGE "Horizontal"+STR(XPos&)
MESSAGE "Vertical"+STR(YPos&)
```

The above example creates a rectangle then displays the coordinates of the upper-right corner in message boxes. The upper-right coordinates are used because the selected object's reference point was changed with the .SetReferencePoint command.

{button ,AL(`OVR1 Transformation commands;',0,"Defaultoverview",)} **Related Topics**

GetSize (DRAW)

.GetSize .XSize = *long**, .YSize = *long**

This function returns the size attributes of a selected object. If more than one object is selected, the function returns the size attributes of the last selected object.

Parameter	Description
.XSize	Returns the horizontal size of the selected object, in tenths of a micron.
.YSize	Returns the vertical size of the selected object, in tenths of a micron.

Example

```
.CreateRectangle 1000000, 750000, 450000, 100000, 0
id& = .GetObjectsCDRStaticID()
.GetSize XSize&, YSize&
MESSAGE "Horizontal"+STR(XSize&)
MESSAGE "Vertical"+STR(YSize&)
```

The above example returns the size of the selected rectangle and displays the width and height (in tenths of a micron) in message boxes.

{button ,AL(`OVR1 Transformation commands;',0,"Defaultoverview",)} **Related Topics**

MoveCenter (DRAW)

.MoveCenter .AnchorID = *long*, .XOffset = *long*, .YOffset = *long*

This command moves the selected object's center of rotation.

Parameter	Description
.AnchorID	Lets you specify the reference point of the object to be skewed. -1 = Use .XOffset and YOffset 1 = Upper-right 2 = Upper-middle 3 = Upper-left 4 = Middle-left 5 = Lower-left 6 = Lower-middle 7 = Lower-right 8 = Middle-right 9 = Center
.XOffset	If .lAnchorID = -1, then .XOffset specifies the distance, in tenths of a micron, that the center of rotation is horizontally offset from the object's center.
.YOffset	If .lAnchorID = -1, then .YOffset specifies the distance, in tenths of a micron, that the center of rotation is vertically offset from the object's center.

{button ,AL(`OVR1 Transformation commands;',0,"Defaultoverview",)} **Related Topics**

MoveObject (DRAW)

.MoveObject .XDelta = *long*, .YDelta = *long*

This command repositions the selected object to the specified location.

Parameter	Description
.XDelta	Lets you specify the distance the object is to be moved along the X-axis in tenths of a micron.
.YDelta	Lets you specify the distance the object is to be moved along the Y-axis in tenths of a micron.

Example

```
.SetPageSize 2159000, 2794000
.CreateRectangle 500000, -750000, -500000, 750000, 0
.MoveObject 250000, -750000
```

The above example creates a rectangle, then moves it to the bottom right corner of an 8.5 by 11 inch page.

{button ,AL(`OVR1 Transformation commands;',0,"Defaultoverview",)} **Related Topics**

ResetTransfo (DRAW)

.ResetTransfo

This command clears all tranformations.

{button ,AL(`OVR1 Transformation commands;',0,"Defaultoverview",)} **Related Topics**

RotateObject (DRAW)

.RotateObject .Angle = *long*, .UseObjectsCenter = *boolean*, .XCenter = *long*, .YCenter = *long*

This command lets you rotate the selected object.

Parameter	Description
.Angle	Lets you specify the angle of rotation of the selected object, expressed in millionths of degrees. Negative values rotate the object clockwise from its current position; positive values rotate it counterclockwise. e.g., 45 degrees clockwise = -45000000
.UseObjectsCenter	Set to TRUE (-1) to enable rotation around the center of the object. Set to FALSE (0) to disable this option.
.XCenter	Lets you specify the logical X-coordinate of the center of the object to be rotated in tenths of a micron, relative to the center of the page.
.YCenter	Lets you specify the logical Y-coordinate of the center of the object to be rotated in tenths of a micron, relative to the center of the page.

Note

- You can use the ANGLECONVERT function to specify angle measurements.

Example

```
.CreateRectangle 500000, -750000, -500000, 750000, 0
.RotateObject 45000000, -1, 0,0
```

The above example rotates the rectangle 45 degrees counter clockwise.

```
.CreateRectangle 500000, -750000, -500000, 750000, 0
.RotateObject -45000000, 0, -500000, 500000
```

The above example rotates the rectangle 45 degrees clockwise about the specified point.

{button ,AL(`OVR1 Transformation commands;',0,"Defaultoverview",)} **Related Topics**

SkewObject (DRAW)

.SkewObject .XAngle = *long*, .YAngle = *long*, .Reference = *long*

This command lets you skew the selected object.

Parameter	Description
.XAngle	Lets you specify the amount of horizontal skew (skew along the X-axis), in millionths of degrees. Positive angles result in counter-clockwise skew. Negative angles result in clockwise skew.
.YAngle	Lets you specify the amount of vertical skew (skew along the Y-axis), in millionths of degrees. Positive angles result in counter-clockwise skew. Negative angles result in clockwise skew.
.Reference	Lets you specify the reference point of the object to be skewed. 1 = Upper-right 2 = Upper-middle 3 = Upper-left 4 = Middle-left 5 = Lower-left 6 = Lower-middle 7 = Lower-right 8 = Middle-right 9 = Center

Note

- You can use the ANGLECONVERT function to specify angle measurements.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.SkewObject -15000000, 20000000, 3
```

The above example creates a rectangle, horizontally skews it 15 degrees clockwise and vertically skews it 20 degrees counterclockwise. The reference point for skewing is the upper-left position.

{button ,AL(`OVR1 Transformation commands;',0,"Defaultoverview",)} **Related Topics**

StretchObject (DRAW)

.StretchObject .XScaleNumerator = *long*, .XScaleDenominator = *long*, .YScaleNumerator = *long*, .YScaleDenominator = *long*, .bHMirror = *boolean*, .bVMirror = *boolean*, .ReferenceNum = *long*

This command stretches or mirrors the selected object.

Parameter	Description
.XScaleNumerator	Lets you specify the amount that the selected object is stretched along the X-axis. The final stretch value is determined by dividing this number by .XScaleDenominator. If the final value is less than 1, the object becomes smaller. If the final value is greater than 1, the object becomes larger.
.XScaleDenominator	Lets you specify the amount that the selected object is stretched along the X-axis. The final stretch value is determined by dividing .XScaleNumerator by this number. If the final value is less than 1, the object becomes smaller. If the final value is greater than 1, the object becomes larger.
.YScaleNumerator	Lets you specify the amount that the selected object is stretched along the Y-axis. The final stretch value is determined by dividing this number by .YScaleDenominator. If the final value is less than 1, the object becomes smaller. If the final value is greater than 1, the object becomes larger.
.YScaleDenominator	Lets you specify the amount that the selected object is stretched along the Y-axis. The final stretch value is determined by dividing .YScaleNumerator by this number. If the final value is less than 1, the object becomes smaller. If the final value is greater than 1, the object becomes larger.
.HMirror	Set to TRUE (-1) to horizontally mirror the selected object.
.VMirror	Set to TRUE (-1) to vertically mirror the selected object.
.ReferenceNum	Lets you specify the reference point of the object to be skewed. 1 = Upper-right 2 = Upper-middle 3 = Upper-left 4 = Middle-left 5 = Lower-left 6 = Lower-middle 7 = Lower-right 8 = Middle-right 9 = Center

{button ,AL(`OVR1 Transformation commands;',0,"Defaultoverview",)} **Related Topics**

Text commands

AddTabStop (DRAW)

.AddTabStop .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .TabStop = *long*

This command adds tab stops to text.

Parameter	Description
.FirstSelectedChar	Lets you specify the starting character of the selected text. Note: The first character in a string is equal to 0, not 1.
.LastSelectedChar	Lets you specify the ending character of the selected text.
.TabStop	Lets you specify the distance at which to apply tabs, in tenths of a micron.

Note

- The .CreateTextString and .SelectObjectsInRect functions must be called before this command.
- You can use the LENGTHCONVERT function, or one of the FROM... or TO... functions to specify length measurements.

Example

```
.CreateTextString 1000000, -1000000, -1000000, 1000000, "Lets you specify the type of
underline. 0 = None 1 = Single thin 2 = Single thin words 3 = Single thick 4 = Single thick
words 5 = Double thin 6 = Double thin words"
.SelectObjectsInRect 1000000, -1000000, -1000000, 1000000, 0
.AddTabStop 0, 0, 1270000
```

The above example adds a tab stop every 0.5 inch.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

AlignTextToBaseline (DRAW)

.AlignTextToBaseline .FirstSelectedChar = *long*, .LastSelectedChar = *long*

This command aligns text to the baseline.

Parameter	Description
.FirstSelectedChar	Lets you specify the starting character of the selected text. Note: The first character in a string is equal to 0, not 1.
.LastSelectedChar	Lets you specify the ending character of the selected text.

Note

- The .CreateTextString and .SelectObjectsInRect functions must be called before this command.

Example

```
.CreateTextString 1000000, -1000000, -1000000, 1000000, "Lets you specify the type of
underline. 0 = None 1 = Single thin 2 = Single thin words 3 = Single thick 4 = Single thick
words 5 = Double thin 6 = Double thin words"
.SelectObjectsInRect 1000000, -1000000, -1000000, 1000000, 0
.AlignTextToBaseline 0, 0
```

The above example aligns the text to the baseline.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

CreateArtisticText (DRAW)

.CreateArtisticText .NewText = *string*, .Left = *long*, .Top = *long*

This command allows you to create a text string with default text settings as Artistic Text. The left-most character of the Artistic Text is placed on the center of the page. Text created with the CreateArtisticText command can be modified using the SetArtisticText command.

Parameter	Description
.NewText	Lets you specify the name of the new text to create.
.Left	Sets the left edge of the artistic text's position in tenths of a micron, relative to the center of the page.
.Top	Sets the top edge of the artistic text's position in tenths of a micron, relative to the center of the page.

Example

```
.CreateArtisticText "CorelDRAW", 1000, 1000
```

The above example displays the text string "CorelDRAW" 100 microns to the left of the center of the page, and 100 microns above the center of the page.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

CreateTextString (DRAW)

.CreateTextString .Top = *long*, .Left = *long*, .Bottom = *long*, .Right = *long*, .Text = *string*

This command creates the text.

Parameter	Description
.Top	Lets you specify the Y-coordinate of the upper-left corner of the text's bounding box in tenths of a micron, relative to the center of the page.
.Left	Lets you specify the X-coordinate of the upper-left corner of the text's bounding box in tenths of a micron, relative to the center of the page.
.Bottom	Lets you specify the Y-coordinate of the lower-right corner of the text's bounding box in tenths of a micron, relative to the center of the page.
.Right	Lets you specify the X-coordinate of the lower-right corner of the text's bounding box in tenths of a micron, relative to the center of the page.
.Text	Lets you specify the text. Maximum string length is 255 characters.

Note

- This function must be called first to create the text before any of the functions which manipulate the text.

Example

```
.CreateTextString 250000, -300000, -250000, 1100000, "COREL"
```

The above example creates the text "COREL".

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

ExtractText (DRAW)

.ExtractText .DestinationFile = *string*

This command extracts the Artistic Text to a text file which can then be edited in any text editor and merged back into the document with MergeTextBack.

Parameter	Description
.DestinationFile	Lets you specify the name of the destination file.

Example

```
.CreateArtisticText "COREL DRAW"
.ExtractText "C:\TEXTFILE.TXT"
```

The above example extracts the text "COREL DRAW" to a text file named "TEXTFILE.TXT".

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

FitTextToPath (DRAW)

.FitTextToPath .TextOrientation = *long*, .VertAlign = *long*, .HorizAlign = *long*, .CurveSideToFit = *long*, .FitOtherSide = *boolean*, .HorizOffset = *long*, .DistFromPath = *long*

This command fits selected artistic text to the selected path.

Parameter	Description
.TextOrientation	0 = Rotates individual characters to follow the contours of the path. 1 = The characters are not changed. 2 = Vertically skews each character, creating the impression that the text is standing upright on the path. 3 = Horizontally skews each character, creating the impression that the text is turning in toward the screen.
.VertAlign	If you specify a distamce from the path, then this parameter has no effect. 0 = Variable. Allows you to move the text off the path by dragging with the mouse. 1 = Bottom. Aligns the descender line of the text with the path. 2 = Top. Aligns the ascender line of the text with the path. 3 = Center. Centers the text vertically on the path. 4 = Baseline. Aligns the baseline of the text with the path.
.HorizAlign	1 = Aligns the text with the start node of the line or curve. 2 = Aligns the text with the end point of the line or curve. 3 = Centers the text on the path.
.CurveSideToFit	1 = Aligns the text to the top of a closed object. 2 = Aligns the text to the left of a closed object. 3 = Aligns the text to the bottom of a closed object. 4 = Aligns the text to the right of a closed object.
.FitOtherSide	Set to TRUE (-1) to place the text on the other side of the path.
.HorizOffset	Lets you specify the distance the text is offset from the start node.
.DistFromPath	Lets you specify the distance the text is from path.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

GetTextFontName (DRAW)

This command returns the font name of the selected text.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

GetTextFontSize (DRAW)

This command returns the font size of the selected text.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

GetTextString (DRAW)

This command returns the actual text string.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

GetTextWordCount (DRAW)

This command returns the number of words in your document.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

MergeBackText (DRAW)

.MergeBackText .SourceFile = *string*

This command merges the extracted text back into the CorelDRAW document.

Parameter	Description
.SourceFile	Lets you specify the name of the source file to merge.

Example

```
.CreateArtisticText "COREL DRAW"
.ExtractText "C:\TEXTFILE.TXT"
.MergeBackText "C:\TEXTFILE.TXT"
```

The above example merges the extracted text from the file "TEXTFILE.TXT" back into the DRAW document.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

SetArtisticText (DRAW)

.SetArtisticText .NewText = *string*

This command allows you to change selected Artistic Text text strings.

Parameter	Description
.NewText	Lets you specify the name of the new text to set.

Example

```
.FileNew
.CreateArtisticText "1"
.FilePrint
FOR i%=2 TO 10 STEP 1
.SetArtisticText i%
.FilePrint
NEXT i%
```

The above example creates the string "1" as Artistic text and then prints the document. Within the FOR...NEXT loop, the Artistic text is changed from the numbers 2 to 10. After each change in the Artistic text, the document is printed.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

SetBullet (DRAW)

.SetBullet .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .SymbolLibrary = *string*, .SymbolNumber = *long*, .PointSize = *long*, .BulletIndent = *long*, .VerticalShift = *long*

This command sets bullets for text.

Parameter	Description
.FirstSelectedChar	Lets you specify the starting character of the selected text. Note: The first character in a string is equal to 0, not 1.
.LastSelectedChar	Lets you specify the ending character of the selected text.
.SymbolLibrary	Lets you specify the name of the symbol library. Refer to the Effects tab of the Paragraph dialog box for more details.
.SymbolNumber	Lets you specify the selected symbol number. Refer to the Effects tab of the Paragraph dialog box for more details.
.PointSize	Lets you specify the point size in tenths of a point.
.BulletIndent	Lets you specify the size of the bullet indentation in tenths of a micron.
.VerticalShift	Lets you specify the amount of baseline shift in tenths of a micron.

Note

- The .CreateTextString and .SelectObjectsInRect functions must be called before this command.

Example

```
.CreateTextString 1000000, -1000000, -1000000, 1000000, "Lets you specify the type of
underline. 0 = None 1 = Single thin 2 = Single thin words 3 = Single thick 4 = Single thick
words 5 = Double thin 6 = Double thin words"
.SelectObjectsInRect 1000000, -1000000, -1000000, 1000000, 0
.SetBullet 32, 123, "Animals 1", 55, 480, 400000, 0
```

The above inserts a camel bullet, indented 1.57 inches.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

SetCharacterAttributes (DRAW)

.SetCharacterAttributes .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .FontName = *string*, .FontStyle = *long*, .PointSize = *long*, .Underline = *long*, .Overline = *long*, .StrikeOut = *long*, .Placement = *long*, .CharacterSpacing = *long*, .WordSpacing = *long*, .LineSpacing = *long*, .Alignment = *long*

This command sets the text character attributes.

Parameter	Description
.FirstSelectedChar	Lets you specify the starting character of the selected text. Note: The first character in a string is equal to 0, not 1.
.LastSelectedChar	Lets you specify the ending character of the selected text.
.FontName	Lets you specify the font name.
.FontStyle	Lets you specify the style of the selected font. 7 = Normal 8 = Normal/Italic 13 = Bold 14 = Bold/Italic
.PointSize	Lets you specify the size of the selected font in tenths of a point.
.Underline	Lets you specify the type of underline. 0 = None 1 = Single thin 2 = Single thin words 3 = Single thick 4 = Single thick words 5 = Double thin 6 = Double thin words
.Overline	Lets you specify the type of overline. 0 = None 1 = Single thin 2 = Single thin words 3 = Single thick 4 = Single thick words 5 = Double thin 6 = Double thin words
.StrikeOut	Lets you specify the type of strikeout. 0 = None 1 = Single thin 2 = Single thin words 3 = Single thick 4 = Single thick words 5 = Double thin 6 = Double thin words
.Placement	Lets you specify the placement of the font. 0 = Normal 1 = Superscript 2 = Subscript
.CharacterSpacing	Lets you specify the character spacing in tenths of a percent.
.WordSpacing	Lets you specify the word spacing in tenths of a percent.
.LineSpacing	Lets you specify the line spacing in tenths of a percent.
.Alignment	Lets you specify the alignment. 0 = None 1 = Left 2 = Center 3 = Right 4 = Full justify 5 = Force justify

Note

- You can include the SCPCONST.CSI and DRWCONST.SCI files in your script. These files define contants for all referenced parameters used with the .SetCharacterAttributes command.

Example

```
.CreateTextString 250000, -300000, -250000, 1100000, "COREL"
```

```
.SelectObjectsInRect 250000, -300000, -250000, 1100000, 0
.SetCharacterAttributes 0, 4, "Arial", 13, 900, 0, 0, 0, 0, 0, 0, 0, 1
```

The above example creates the text "COREL", then sets the font to Arial, the font type to Bold, and the point size to 90.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

SetFrameColumn (DRAW)

.SetFrameColumn .ColumnNumber = *long*, .Width = *long*, .GutterWidth = *long*

This command formats columns for text.

Parameter	Description
.ColumnNumber	Lets you specify the column number.
.Width	Lets you specify the width of the column in tenths of a micron.
.GutterWidth	Lets you specify the width of the gutter in tenths of a micron.

Note

- The .CreateTextString and .SelectObjectsInRect functions must be called before this command. This command must be called twice.
- You can use the LENGTHCONVERT function, or one of the FROM... or TO... functions to specify length measurements.

Example

```
.CreateTextString 1000000, -1000000, -1000000, 1000000, "Lets you specify the type of
underline. 0 = None 1 = Single thin 2 = Single thin words 3 = Single thick 4 = Single thick
words 5 = Double thin 6 = Double thin words"
.SelectObjectsInRect 1000000, -1000000, -1000000, 1000000, 0
.SetFrameColumn 0, 500000, 50000
.SetFrameColumn 1, 500000, 50000
```

The above example formats the text into two columns, each 2 inches wide.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

SetIndents (DRAW)

.SetIndents .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .FirstLine = *long*, .RestOfLines = *long*, .RightMargin = *long*

This command sets indents for text.

Parameter	Description
.FirstSelectedChar	Lets you specify the starting character of the selected text. Note: The first character in a string is equal to 0, not 1.
.LastSelectedChar	Lets you specify the ending character of the selected text.
.FirstLine	Lets you specify the size of the first line indentation, in tenths of a micron.
.RestOfLines	Lets you specify the size of the remaining line indentation, in tenths of a micron.
.RightMargin	Lets you specify the size of the right margin indentation, in tenths of a micron.

Note

- The .CreateTextString and .SelectObjectsInRect functions must be called before this command.
- You can use the LENGTHCONVERT function, or one of the FROM... or TO... functions to specify length measurements.

Example

```
.CreateTextString 1000000, -1000000, -1000000, 1000000, "Lets you specify the type of
underline. 0 = None 1 = Single thin 2 = Single thin words 3 = Single thick 4 = Single thick
words 5 = Double thin 6 = Double thin words"
.SelectObjectsInRect 1000000, -1000000, -1000000, 1000000, 0
.SetIndents 0, 0, 0, 400000, 0
```

The above example indents all lines except the first by 1.57 inches.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

SetParagraphSpacing (DRAW)

.SetParagraphSpacing .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .CharacterSpacing = *long*, .WordSpacing = *long*, .LineSpacing = *long*, .BeforeParagraph = *long*, .AfterParagraph = *long*, .Alignment = *long*, .AutoHyphenation = *boolean*, .HyphenHotZone = *long*

This command sets paragraph spacing.

Parameter	Description
.FirstSelectedChar	Lets you specify the starting character of the selected text. Note: The first character in a string is equal to 0, not 1.
.LastSelectedChar	Lets you specify the ending character of the selected text.
.CharacterSpacing	Lets you specify the character spacing in tenths of a percent.
.WordSpacing	Lets you specify the word spacing in tenths of a percent.
.LineSpacing	Lets you specify the line spacing in tenths of a percent.
.BeforeParagraph	Lets you specify the spacing before paragraphs in tenths of a percent.
.AfterParagraph	Lets you specify the spacing after paragraphs in tenths of a percent.
.Alignment	Lets you specify the alignment. 0 = None 1 = Left 2 = Center 3 = Right 4 = Full justify 5 = Force justify
.AutoHyphenation	Set to TRUE (-1) to enable automatic hyphenation. Set to FALSE (0) to disable this option.
.HyphenHotZone	Lets you specify the size of the hyphen hot zone in tenths of a micron.

Note

- The .CreateTextString and .SelectObjectsInRect functions must be called before this command.

Example

```
.CreateTextString 1000000, -1000000, -1000000, 1000000, "Lets you specify the type of
underline. 0 = None 1 = Single thin 2 = Single thin words 3 = Single thick 4 = Single thick
words 5 = Double thin 6 = Double thin words"
.SelectObjectsInRect 1000000, -1000000, -1000000, 1000000, 0
.SetParagraphSpacing 0, 0, 900, 900, 900, 200, 200, 1, 0, 0
```

The above example creates a text string, selects the entire text and applies paragraph spacing to it.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

SetTextString (DRAW)

.SetTextString .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .Text = *string*

This command changes the text in a selected text object.

Parameter	Description
.FirstSelectedChar	Lets you specify the starting character of the selected text. Note: The first character in a string is equal to 0, not 1.
.LastSelectedChar	Lets you specify the ending character of the selected text.
.Text	Lets you specify the text. Maximum string length is 255 characters.

Note

- The .CreateTextString and .SelectObjectsInRect functions must be called before this command.

Example

```
.CreateTextString 250000, -300000, -250000, 1100000, "COREL"
.SelectObjectsInRect 250000, -300000, -250000, 1100000, 0
.SetCharacterAttributes 0, 4, "Arial", 13, 900, 0, 0, 0, 0, 0, 0, 0, 1
.SetTextString -1, -1, "RT"
.SetCharacterAttributes 5, 6, "Arial", 8, 900, 0, 0, 0, 1, 0, 0, 0, 0
```

The above example creates the text string "COREL", then appends a second text string "RT" to it. The appended string is italic and superscript.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

StraightenText (DRAW)

.StraightenText .FirstSelectedChar = *long*, .LastSelectedChar = *long*

This command resets the kerning angle to 0 for selected text.

Parameter	Description
.FirstSelectedChar	Lets you specify the starting character of the selected text. Note: The first character in a string is equal to 0, not 1.
.LastSelectedChar	Lets you specify the ending character of the selected text.

Example

```
.StraightenText 0, 0
```

The above example straightens the first character of selected paragraph text.

{button ,AL(`OVR1 Text commands;',0,"Defaultoverview",)} **Related Topics**

Fill and outline commands

AddArrowPoint (DRAW)

.AddArrowPoint .X = *long*, .Y = *long*, .Enabled = *boolean*, .Letter = *boolean*, .User = *boolean*, .Closed = *boolean*, .Continuity = *long*, .NodeType = *long*

This command adds a node that is part of an arrowhead. This command must be part of a block of commands that begins with the .BeginDrawArrow command and ends with the .EndDrawArrow command. To add the arrowhead to a path, use the .SetOutlineArrow command.

Parameter	Description
.X	Lets you specify the X-coordinate of the node in tenths of a micron, relative to the center of the arrowhead area.
.Y	Lets you specify the Y-coordinate of the node in tenths of a micron, relative to the center of the arrowhead area.
.Enabled	Always set to FALSE (0)
.Letter	Always set to FALSE (0)
.User	Set to TRUE (-1) to make this node selectable.
.Closed	Set to TRUE (-1) to make the arrowhead closed.
.Continuity	0 = cusp node 1 = smooth node 2 = symmetrical node
.NodeType	1 = straight line segment 2 = curved line segment

Example

```
.BeginDrawArrow TRUE, 0, 7
.AddArrowPoint -1016000, 0, FALSE, FALSE, TRUE, TRUE, 0, 0
.AddArrowPoint -481838, -271272, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 571500, -271272, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 31750, 0, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 571500, 299974, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint -481838, 299974, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint -1016000, 0, FALSE, FALSE, FALSE, TRUE, 0, 1
.EndDrawArrow
.SetOutlineArrow 0
```

The above example adds an arrowhead to the selected line.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ApplyFountainFill (DRAW)

.ApplyFountainFill .Type = *long*, .CenterX = *long*, .CenterY = *long*, .Angle = *long*, .Steps = *long*, .Padding = *long*, .Blend = *long*, .Rate = *long*

This command lets you apply a Fountain Fill to the selected object. If the Blend was Custom, then all intermediate colors will be lost unless the Blend applied is again Custom. If the existing fill is not fountain, the start color will be CMYK Black and the end color CMYK white.

Parameter	Description
.Type	Lets you specify the type of Fountain Fill to apply: 0 = Linear (default) 1 = Radial 2 = Conical 3 = Square
.CenterX	Lets you specify the horizontal offset of the center of the fill. Valid values range from -100 to 100 percent. A value of -50% will place the center on the left edge of your object; a value of 50% will place it on the right edge.
.CenterY	Lets you specify the vertical offset of the center of the fill. Valid values range from -100 to 100 percent. A value of -50% will place the center on the bottom edge of your object; a value of 50% will place it on the top edge.
.Angle	Lets you specify the angle at which the fill is applied in tenths of degrees. Positive values will rotate the fill counter-clockwise, negative values will rotate it clockwise.
.Steps	Lets you specify the number of steps you want. Lower values produce coarser fountains on screen which take less time to redraw.
.Padding	Lets you specify the amount of padding to apply to the fill. Ignored for type 2. Valid values range from 0 to 45 percent.
.Blend	Lets you specify the type of blending to apply to the fill. 0 = Direct (default) 1 = Rainbow CW 2 = Rainbow CCW 3 = Custom
.Rate	Lets you specify the mid-point used to apply the fill. Valid values range from 1 to 99.

Note

- The Horizontal and Vertical Offset options are not available for linear fountain fills; set parameters to 0.
- The Angle option is not available for circular fountain fills; set parameter to 0.
- You can use the ANGLECONVERT function to specify angle measurements
- You can include the SCPCONST.CSI and DRWCONST.SCI files in your script. These files define contants for all referenced parameters used with the .ApplyFountainFill command.

Example

```
.CreateEllipse -250000, -500000, 250000, 500000, 0, 0, 0
.ApplyFountainFill 1, -50, -50, 900, 20, 20, 2, 1
```

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ApplyFullColorFill (DRAW)

.ApplyFullColorFill .FileName = *string*, .TileWidth = *long*, .TileHeight = *long*, .FirstTileOffsetX = *long*, .FirstTileOffsetY = *long*, .RowOffset = *boolean*, .RowColumnOffset = *long*, .SeamlessTiling = *boolean*, .ScaleWithObject = *boolean*[, .VectorBBoxTop = *long*][, .VectorBBoxBottom = *long*][, .VectorBBoxLeft = *long*][, .VectorBBoxRight = *long*][, .RotationAngle = *long*][, SkewAngle = *long*]

This command lets you apply a Full-Color fill to the selected object.

Parameter	Description
.FileName	Lets you specify the name of the Fill file.
.TileWidth	If less than 500 and ScaleWithObject is set, is a percentage of the object width, otherwise is in tenths of a micron.
.TileHeight	If less than 500 and ScaleWithObject is set, is a percentage of the object height, otherwise is in tenths of a micron.
.FirstTileOffsetX	X offset from center of object in the same units used by width & height.
.FirstTileOffsetY	Y offset from center of object in the same units used by widht & height.
.RowOffset	Set to TRUE (-1) to enable row offset. Set to FALSE (0) to enable column offset.
.RowColumnOffset	Lets you specify the amount of row or column offsets. Valid values range from 0 to 100.
.SeamlessTiling	Set to TRUE (-1) to enable seamless tiling. Set to FALSE (0) to disable this option.
.ScaleWithObject	Set to TRUE (-1) to scale the pattern with the object. Set to FALSE (0) to disable this option.
.VectorBBoxTop (optional)	Lets you specify the top coordinate of the fill's bounding box in tenths of a micron, relative to the center of the page.
.VectorBBoxBottom (optional)	Lets you specify the bottom coordinate of the fill's bounding box in tenths of a micron, relative to the center of the page.
.VectroBBoxLeft (optional)	Lets you specify the left coordinate of the fill's bounding box in tenths of a micron, relative to the center of the page.
.VectorBBoxRight (optional)	Lets you specify the right coordinate of the fill's bounding box in tenths of a micron, relative to the center of the page.
.Rotation Angle (optional)	Lets you specify the amount in which the tile is rotated in millionths of degrees.
.SkewAngle (optional)	Lets you specify the amount in which the tile is skewed in millionths of degrees.

Note

- You can use the LENGTHCONVERT function, or one of the FROM... or TO... functions to specify length measurements.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.ApplyFullColorFill "C:\MONTEMP.BMP", 500000, 500000, 100, 100, 0, 100, 0, 0
```

The above example applies a full color fill to a rectangle.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ApplyNoFill (DRAW)

.ApplyNoFill

This command removes the fill from the selected object, allowing objects behind it to show through.

Example

```
.SelectAllObjects
.ApplyNoFill
```

The above example removes the fill from all objects.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} Related Topics

ApplyOutline (DRAW)

.ApplyOutline .Width = *long*, .Type = *long*, .EndCaps = *long*, .JoinType = *long*, .Aspect = *long*, .Angle = *long*, .DotDash = *long*, .RightArrow = *long*, .LeftArrow = *long*, .BehindFill = *boolean*, .OutlineType = *long*, .Preset = *long*, .ScalePen = *boolean*

This command lets you apply an Outline to the selected object.

Parameter	Description
.Width	Lets you specify the width of the outline to apply, in tenths of a micron.
.Type	Lets you specify the outline type: 0 = None 1 = Solid 2 = Dot - Dash
.EndCaps	Lets you specify the end caps to be applied to the outline: 0 = Butt 1 = Round 2 = Square
.JoinType	Lets you specify the outline join types: 0 = Miter 1 = Round 2 = Bevel
.Aspect	Lets you specify the stretch field which adjusts the width of the nib. Valid values range from 1 to 100 percent.
.Angle	Lets you specify the angle of the nib's edge, in tenths of degrees.
.DotDash	Lets you specify the type of dot/dash line. Dot/dash line types are listed in the Style drop-down list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.RightArrow	Lets you specify the style of right-arrow. Right-arrow types are listed in the right arrow drop-down list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left-to-right) the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.LeftArrow	Lets you specify the style of left-arrow. Left-arrow types are listed in the left arrow drop-down list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left-to-right) the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.BehindFill	Set to TRUE (-1) to position the outline behind the fill. Set to FALSE (0) to position the outline in front of the fill.
.OutlineType	Lets you specify the type of preset outline. 0 = Pen 1 = Outline 2 = PenOutline
.Preset	Lets you specify the tint of the outline. Values range from 1 (0%) to 11 (100%). A value of 0 has no effect on the outline. This parameter is only used if the selected outline type supports preset tints.
.ScalePen	Set to TRUE (-1) to scale the outline when the object is scaled.

Note

- You can use the ANGLECONVERT function to specify angle measurements
- You can include the SCPCONST.CSI and DRWCONST.SCI files in your script. These files define contants for all referenced parameters used with the .ApplyOutline command.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.ApplyOutline 50000, 2, 0, 1, 50, 250, 2, 0, 0, 0
```

The above example applies a dashed outline 50000 microns wide, with round corners to the rectangle.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ApplyPostscriptFill (DRAW)

.ApplyPostscriptFill .PSFill = *string*, .NumParms = *long*, .Parm1 = *long*, .Parm2 = *long*, .Parm3 = *long*, .Parm4 = *long*, .Parm5 = *long*

This command lets you apply a PostScript Fill to a selected object.

Parameter	Description
.PSFill	Lets you specify the name of the postscript fill. The name must be preceded by an F/. For a listing of Postscript fills available, see the PostScript Texture dialog box. If you create custom PostScript fills, their definitions are placed in the USERPROC.PS file in the Custom folder of your Corel folder. The PostScript fills definitions supplied with DRAW are also in this file.
.NumParms	Lets you specify the number of parameters used for the selected PostScript Fill, an integer value between 1 and 5, inclusive. Refer to the PostScript Texture dialog box to determine the number of parameters for a full fill. Set to 2 for spot fills.
.Parm1	Lets you specify the first parameter for the selected PostScript Fill. This parameter varies depending on the Fill selected. Refer to the PostScript Texture dialog box for full fill parameter specifics. If you're using a spot fill, set Parm1 to a value between -1 and 1.
.Parm2	Lets you specify the second parameter for the selected PostScript Fill. This parameter varies depending on the Fill selected. Refer to the PostScript Texture dialog box for full fill parameter specifics. If the parameter is not used with the fill you selected, set it to 0. If you're using a spot fill, set Parm2 to a value between -1 and 1.
.Parm3	Lets you specify the third parameter for the selected PostScript Fill. This parameter varies depending on the Fill selected. Refer to the PostScript Texture dialog box for full fill parameter specifics. If the parameter is not used with the fill you selected, set it to 0. If you're using a spot fill, set this parameter to 0.
.Parm4	Lets you specify the fourth parameter for the selected PostScript Fill. This parameter varies depending on the Fill selected. Refer to the PostScript Texture dialog box for full fill parameter specifics. If the parameter is not used with the fill you selected, set it to 0. If you're using a spot fill, set this parameter to 0.
.Parm5	Lets you specify the fifth parameter for the selected PostScript Fill. This parameter varies depending on the Fill selected. Refer to the PostScript Texture dialog box for full fill parameter specifics. If the parameter is not used with the fill you selected, set it to 0. If you're using a spot fill, set this parameter to 0.

Note

- If you create custom PostScript fills, their definitions are placed in the USERPROC.PS file in the Custom folder of your Corel folder. The PostScript fills definitions supplied with CorelDRAW are also in this file.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.ApplyPostScriptFill "F/StoneWall", 4, 15, 100,0, 5, 0
```

The above example applies the StoneWall PostScript fill to the selected rectangle.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ApplyPreset (DRAW)

.ApplyPreset .PresetFileName = *string*, .PresetName = *string*

This command lets you load and apply a Preset.

Parameter	Description
.PresetFileName	Lets you specify the name of the Preset File.
.PresetName	Lets you specify the name of the Preset.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.ApplyPreset "C:\CORELDRW.PST", "Button Blue"
```

The above example applies the specified preset fill to the rectangle.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ApplyTextureFill (DRAW)

.ApplyTextureFill .TextureLibrary = *string*, .TextureName = *string*, .TextureStyle = *string,* .TextureWidth = *long*, .TextureHeight = *long*, .TextureOffsetX = *long*, .TextureOffsetY = *long*, .RowOffset = *boolean*, .RowColumnOffset = *long*, .ScaleWithObject = *boolean*, .RotationAngle = *long*, .SkewAngle = *long*

This command lets you apply one of the texture fills included in CorelDRAW.

Parameter	Description
.TextureLibrary	Lets you specify the name of the Texture Library.
.TextureName	Lets you specify the name of the texture.
.TextureStyle	Lets you specify the name of the style. If you set .TextureLibrary to "Samples 5", the style name must be preceded by "CDR5:". For example, "CDR5:Blue Valley".
.TextureWidth	If less than 500 and ScaleWithObject is set, is a percentage of the object width, otherwise is in tenths of a micron.
.TextureHeight	If less than 500 and ScaleWithObject is set, is a percentage of the object height, otherwise is in tenths of a micron.
.TextureOffsetX	X offset from center of object in the same units used by width & height.
.TextureOffsetY	Y offset from center of object in the same units used by widht & height.
.RowOffset	Set to TRUE enables the row offset, Set to FALSE enables the column offset.
.RowColumnOffset	Lets you specify the amount that the row or columns is offset as a percentage.
.ScaleWithObject	Set to TRUE will transform the fill with the object.
.RotationAngle	Lets you specify the amount that the texture is rotated in millionths of degrees.
.SkewAngle	Lets you specify the amount that the texture is skewed in millionths of degrees.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.ApplyTextureFill "Styles", "Satellite Photography", "Satellite Photography"
```

The above example creates a rectangle, then applies the satellite photography fill to it.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ApplyTwoColorFill (DRAW)

.ApplyTwoColorFill .FileName = *string*, .TileWidth = *long*, .TileHeight = *long*, .FirstTileOffsetX = *long*, .FirstTileOffsetY = *long*, .RowOffset = *boolean*, .RowColumnOffset = *long*, .SeamlessTiling = *boolean*, .ScaleWithObject = *boolean,* .RotationAngle = *long,* .SkewAngle = *long*

This command lets you apply a Two-Color fill to the selected object.

Parameter	Description
.FileName	Lets you specify the name of the two color fill file to use. See the Two-Color Bitmap Pattern dialog box for a list of valid file formats.
.TileWidth	If less than 500 and ScaleWithObject is set, is a percentage of the object width, otherwise is in tenths of a micron.
.TileHeight	If less than 500 and ScaleWithObject is set, is a percentage of the object height, otherwise is in tenths of a micron.
.FirstTileOffsetX	X offset from center of object in the same units used by width & height.
.FirstTileOffsetY	Y offset from center of object in the same units used by widht & height.
.RowOffset	Set to TRUE (-1) to enable row offset. Set to FALSE (0) to enable column offset.
.RowColumnOffset	Lets you specify the amount of row or column offsets. Valid values range from 0 to 100.
.SeamlessTiling	Set to TRUE (-1) to enable seamless tiling. Set to FALSE (0) to disable this option.
.ScaleWithObject	Set to TRUE (-1) to enable seamless tiling. Set to FALSE (0) to disable this option.
.Rotation Angle	Lets you specify the amount in which the tile is rotated in millionths of degrees.
.SkewAngle	Lets you specify the amount in which the tile is skewed in millionths of degrees.

Note

- You can use the LENGTHCONVERT function, or one of the FROM... or TO... functions to specify length measurements.

Example

```
.CreateRectangle 1000000, -500000, -1000000, 500000, 0
.ApplyTwoColorFill "mybitmap.bmp", 5, 255, 0, 0, 0, 5, 0, 0, 0, 0, 500000, 500000, 100, 100,
0, 100, 0, 0, 0
```

The above example applies a two-color bitmap fill from the MYBITMAP.BMP file to the rectangle.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ApplyUniformFillColor (DRAW)

.ApplyUniformFillColor

This command lets you apply a Uniform Fill Color to a selected object.

Example

```
.CreateEllipse -250000, -500000, 250000, 500000, 0, 0, 0
.StoreColor DRAW_COLORMODEL_CMYK, 100, 100, 0, 0, 0
.ApplyUniformFillColor
```

The above example creates an ellipse and uniformly fills it with cyan.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

BeginDrawArrow (DRAW)

.BeginDrawArrow .LeftArrow = *boolean*, .LineOffset = *long*, .NumOfPoints = *long*

This command initializes a block of arrowhead creation commands. This block must include one or more instances of the .AddArrowPoint command and this block must end with the .EndDrawArrow command. To add the arrowhead to a path, use the .SetOutlineArrow command.

Parameter	Description
.LeftArrow	Set to TRUE (-1) to create a left-facing arrowhead. Set to FALSE (0) to create a right-facing arrowhead.
.LineOffset	Lets you specify the distance between the end of the path and the arrowhead.
.NumOfPoints	Lets you specify the number of nodes in your arrowhead.

Example

```
.BeginDrawArrow TRUE, 0, 7
.AddArrowPoint -1016000, 0, FALSE, FALSE, TRUE, TRUE, 0, 0
.AddArrowPoint -481838, -271272, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 571500, -271272, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 31750, 0, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 571500, 299974, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint -481838, 299974, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint -1016000, 0, FALSE, FALSE, FALSE, TRUE, 0, 1
.EndDrawArrow
.SetOutlineArrow 0
```

The above example adds an arrowhead to the selected line.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ConvertColor (DRAW)

.ConvertColor .ToColorModel = *long*, .V1 = *long*, .V2 = *long*, .V3 = *long*, .V4 = *long*, .V5 = *long*, .V6 = *long*

This command converts one color model to another color model.

Parameter	Description
.ToColorModel	Lets you specify the destination folder.
.V1	Returns the first color component.
.V2	Returns the second color component.
.V3	Returns the third color component.
.V4	Returns the fourth color component.
.V5	Returns the fifth color component.
.V6	Returns the sixth color component.
.V7	Returns the color density from 0 to 100

Example

```
.GetUniformFillColor InModel, In1, In2, In3, In4, In5, In6, In7
.StoreColor InModel, In1, In2, In3, In4, In5, In6, In7
.ConvertColor DRAW_COLORMODEL_RGB, RedVal, GREENVAL, BLUEVAL, 0, 0, 0, 0
```

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

CreatePaletteFromDocument (DRAW)

.CreatePaletteFromDocument .Filename = *string*, .Overwrite = *boolean*

This command creates a color palette from the current document.

Parameter	Description
.Filename	Lets you specify the file name for the color palette.
.Overwrite	Set to TRUE (-1) overwrites the specifief file name.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

CreatePaletteFromSelection (DRAW)

.CreatePaletteFromSelection .Filename = *string*, .Overwrite = *boolean*

This command creates a color palette from the current selection.

Parameter	Description
.Filename	Lets you specify the file name for the color palette.
.Overwrite	Set to TRUE (-1) overwrites the specifief file name.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

DeletePaletteColor (DRAW)

.DeletePaletteColor .Index = *long*

This command deletes the palette color from the default palette.

Parameter	Description
.Index	Lets you specify the color index that you want to delete.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

EndDrawArrow (DRAW)

.EndDrawArrow

This command ends a block of arrowhead creation commands. This block must include one or more instances of the .AddArrowPoint command and this block must begin with the .BeginDrawArrow command.

Example

```
.BeginDrawArrow TRUE, 0, 7
.AddArrowPoint -1016000, 0, FALSE, FALSE, TRUE, TRUE, 0, 0
.AddArrowPoint -481838, -271272, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 571500, -271272, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 31750, 0, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 571500, 299974, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint -481838, 299974, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint -1016000, 0, FALSE, FALSE, FALSE, TRUE, 0, 1
.EndDrawArrow
.SetOutlineArrow 0
```

The above example adds an arrowhead to the selected line.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetColor (DRAW)

.GetColor .StoreColor = *boolean*[, .ColorModel = *long*][, .V1 = *long*][, .V2 = *long*][, .V3 = *long*][, .V4 = *long*][, .V5 = *long*][, .V6 = *long*][, .V7 = *long*]

This function gets a color from the color dialog box.

Parameter	Description
.StoreColor	-1 stores the color 0 does not store the color
.ColorModel (optional)	Lets you specify the Color Model to use: 1 = Pantone 2 = CMYK100 3 = CMYK255 4 = CMY 5 = RGB 6 = HSB 7 = HLS 8 = BW 9 = Gray 11 = YIQ255 12 = LAB 13=Index 14=Pantone Hex 15=Hexachrome To use a color palette with the color model, multiply the palette value (listed below) by 1000 and add it to the color model value. Color palettes: 1 = TRUMATCH 2 = PANTONE PROCESS 3 = PANTONE SPOT 4 = IMAGE 5 = USER 6 = CUSTOMFIXED 7 = RGBSTANDARD 8 = FOCOLTONE 9 = DUPONT 10 = TOYO 11 = DIC 12 = PANTONE HEX 13 = LAB 14 = NETSCAPE 15 = EXPLORER 16 = USERINKS
.Color1 (optional)	Lets you specify the first color component for .ColorModel. For example, Hue is the first color component for HSB. Click + for valid value ranges.
.Color2 (optional)	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.Color3 (optional)	Lets you specify the third color component for .ColorModel. For example, Saturation is the third color component for HLS. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.Color4 (optional)	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click + for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.V5 (optional)	Lets you specify the fifth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.V6 (optional)	Lets you specify the sixth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.V7 (optional)	Lets you specify the seventh color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetCurrentPaletteName (DRAW)

.GetCurrentPaletteName

This function returns the name of the current color palette.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetFillType (DRAW)

ReturnValue& = .GetFillType()

This function returns the Fill Type of a selected object. If more than one object is selected, the function returns the Fill Type of the last selected object.

Return Value

Returns one of the following values:

- 0 None
- 1 Uniform
- 2 Fountain
- 6 PostScript
- 7 Two-color
- 9 ColorBitmap
- 10 Vector
- 11 Texture

Note

- You can include the SCPCONST.CSI and DRWCONST.SCI files in your script. These files define contants for all referenced parameters used with the .GetFillType command.

Example

```
.SelectObjectOfCDRStaticID IDRect&
fillType& = .GetFillType()
Message fillType&
```

The above example displays a number corresponding to the fill type of the selected object in a message box.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetFountainFill (DRAW)

.GetFountainFill .Type = *long**, .CenterX = *long**, .CenterY = *long**, .Angle = *long**, .Steps = *long**, .Padding = *long**, .Blend = *long**, .Rate = *long**, .NumColors = *long**

This command returns the Fountain Fill attributes of a selected object. If more than one object is selected, the function returns the Fountain Fill attributes of the last selected object.

Parameter	Description
.Type	Returns the type of Fountain Fill: 0 = Linear (default) 1 = Radial 2 = Conical 3 = Square
.CenterX	Returns the Horizontal Offset of the center of the fill. Valid values range from -100 to +100 percent. A value of -50% will place the center on the left edge of your object; a value of 50% will place it on the right edge.
.CenterY	Returns the Horizontal Offset of the center of the fill. Valid values range from -100 to +100 percent. A value of -50% will place the center on the bottom edge of your object; a value of 50% will place it on the top edge.
.Angle	Returns the angle at which the fill is applied in degrees. Positive values will rotate the fill counter-clockwise, negative values will rotate it clockwise.
.Steps	Returns the number of stripes you want. Lower values produce coarser fountains on screen which take less time to redraw. Valid values range from 2 to 256.
.Padding	Returns the amount of padding to apply to the fill. Ignored for type 2. Valid values range from 0 to 45 percent.
.Blend	Returns the type of blending to apply to the fill. 0 = Direct (default) 1 = Rainbow CW 2 = Rainbow CCW 3 = Custom
.Rate	Returns the rate method used to apply the fill.
.NumColors	Returns the number of colors.

Note

- You can use the ANGLECONVERT function to specify angle measurements

Example

```
.GetFountainFill fillType&, CX&, CY&, Angle&, Steps&, Pad&, Blend&, Rate&, Num&
MESSAGE fillType&
```

The above example returns Fountain Fill attributes and displays a number corresponding to the fill type in a message box.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetFountainFillColor (DRAW)

.GetFountainFillColor .Index = *long*, .Position = *long**, .ColorModel = *long**, .V1 = *long**, .V2 = *long**, .V3 = *long**, .V4 = *long*,* .V5 = *long*,* .V6 = *long*,* Density = *long**

This command Retrieves a color from a fountain fill.

Parameter	Description
.Index	The index number of the color you want to get. Use the .GetFountainFill command to find out how many colors are in a fountain fill. Valid index numbers will range from 0 to the number of colors minus 1. If you use the value 100, you will always get the end color.
.Position	Return the position of the color within the fill.
.ColorModel	Returns the Color Model to use: 1 = Pantone 2 = CMYK100 3 = CMYK255 4 = CMY 5 = RGB 6 = HSB 7 = HLS 8 = BW 9 = Gray 11 = YIQ255 12 = LAB 13=Index 14=Pantone Hex 15=Hexachrome
.V1	Returns the first color component for .ColorModel. For example, Hue is the first color component for HSB. Click J for valid value ranges.
.V2	Returns the second color component for .ColorModel. For example, Green is the second color component for RGB. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.V3	Returns the third color component for .ColorModel. For example, Saturation is the third color component for HLS. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.V4	Returns the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.V5	Returns the fifth color component for .plColorModel.
.V6	Returns the sixth color component for .plColorModel.
.Density	Returns the color density.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetOutline (DRAW)

.GetOutline .Width = *long**, .Type = *long**, .EndCaps = *long**, .JoinType = *long**, .Aspect = *long**, .Angle = *long**, .DotDash = *long**, .RightArrow = *long**, .LeftArrow = *long**, .BehindFill = *boolean**, .ScalePen = *boolean**

This function returns the outline attributes of the selected object. If more than one object is selected, the function returns the outline attributes of the last selected object.

Parameter	Description
.Width	Returns the width of the outline, in tenths of a micron.
.Type	Returns the outline type: 0 = None 1 = Solid 2 = Dot - Dash
.EndCaps	Returns the End Caps applied to the outline: 0 = Butt 1 = Round 2 = Square
.JoinType	Returns the outline join types: 0 = Miter 1 = Round 2 = Bevel
.Aspect	Returns the stretch field which adjusts the width of the nib.
.Angle	Returns the angle of the nib's edge, in tenths of degrees.
.DotDash	Returns the type of dot/dash line. Refer to the Outline Pen dialog box for more details.
.RightArrow	Returns the style of right-arrow. Refer to the Outline Pen dialog box for more details.
.LeftArrow	Returns the style of left-arrow. Refer to the Outline Pen dialog box for more details.
.BehindFill	Returns the position of the outline fill. TRUE (-1) = Outline behind fill FALSE (0) = Outline in front of fill
.ScalePen	Returns the the scale pen setting. TRUE (-1) = Outline is scaled when object is scaled FALSE (0) = Outline is not scaled when object is scaled

Note

- You can use the ANGLECONVERT function to specify angle measurements

Example

```
.GetOutline Width&, outlineType&, EndCaps&, JoinType&, Aspect&, Angle&, DotDash&, RArrow&,
LArrow&, BehindFill&
```

The above example returns the outline attributes of the selected object.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetOutlineColor (DRAW)

.GetOutlineColor .ColorModel = *long**, .Color1 = *long**, .Color2 = *long**, .Color3 = *long**, .Color4 = *long**, .Color5 = *long**, .Color6 = *long*,* Density = *long*

This function returns the Outline Color attributes of a selected object. If more than one object is selected, the function returns the Outline Color attributes of the last selected object.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 1 = Pantone 2 = CMYK100 3 = CMYK255 4 = CMY 5 = RGB 6 = HSB 7 = HLS 8 = BW 9 = Gray 11 = YIQ255 12 = LAB 13=Index 14=Pantone Hex 15=Hexachrome To use a color palette with the color model, multiply the palette value (listed below) by 1000 and add it to the color model value. Color palettes: 1 = TRUMATCH 2 = PANTONE PROCESS 3 = PANTONE SPOT 4 = IMAGE 5 = USER 6 = CUSTOMFIXED 7 = RGBSTANDARD 8 = FOCOLTONE 9 = DUPONT 10 = TOYO 11 = DIC 12 = PANTONE HEX 13 = LAB 14 = NETSCAPE 15 = EXPLORER 16 = USERINKS
.Color1	Returns the first color component for .ColorModel. For example, Hue is the first color component for HSB. Click J for valid value ranges.
.Color2	Returns the second color component for .ColorModel. For example, Green is the second color component for RGB. Click J for valid value ranges.
.Color3	Returns the third color component for .ColorModel. For example, Saturation is the third color component for HLS. Click J for valid value ranges.
.Color4	Returns the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click J for valid value ranges.
.Color5	Lets you specify the fifth color component for .ColorModel.
.Color6	Lets you specify the sixth color component for .ColorModel.
.Density	Lets you specify the color density from 0-100.

Example

```
.GetOutlineColor Model&, C1&, C2&, C3&, C4&, C5&, C6&, C7&
MESSAGE Model&
```

The above example determines the outline color attributes of the selected object and displays a number corresponding to the color model in a message box.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetPaletteColor (DRAW)

.GetPaletteColor .Index = *long*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*, .Color5 = *long*, .Color6 = *long,* .Color7 = *long*

This function returns the palette color attributes of a selected object.

Parameter	Description
.Index	Lets you specify the color index.
.ColorModel	Lets you specify the Color Model to use: 1 = Pantone 2 = CMYK100 3 = CMYK255 4 = CMY 5 = RGB 6 = HSB 7 = HLS 8 = BW 9 = Gray 11 = YIQ255 12 = LAB 13=Index 14=Pantone Hex 15=Hexachrome To use a color palette with the color model, multiply the palette value (listed below) by 1000 and add it to the color model value. Color palettes: 1 = TRUMATCH 2 = PANTONE PROCESS 3 = PANTONE SPOT 4 = IMAGE 5 = USER 6 = CUSTOMFIXED 7 = RGBSTANDARD 8 = FOCOLTONE 9 = DUPONT 10 = TOYO 11 = DIC 12 = PANTONE HEX 13 = LAB 14 = NETSCAPE 15 = EXPLORER 16 = USERINKS
.Color1	Lets you specify the first color component for .ColorModel. For example, Hue is the first color component for HSB. Click J for valid value ranges.

.Color2 Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3 Lets you specify the third color component for .ColorModel. For example, Saturation is the third color component for HLS. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color5 Lets you specify the fifth color component for .ColorModel.

.Color6 Lets you specify the sixth color component for .ColorModel.

.Color7 Lets you specify the seventh color component (density)

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetPaletteColorCount (DRAW)

.GetPaletteColorCount

This function returns the number of colors in the default Color Palette.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetPaletteColorName (DRAW)

.GetPaletteColorName .Index = *long*

This command retrieves the Color name for a specific index.

Parameter	Description
.Index	Lets you specify the index of the color for which you want to retrieve the name.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

GetUniformFillColor (DRAW)

.GetUniformFillColor .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*, .Color5 = *long*, .Color6 = *long,* .Color7 = *long*

This function returns the Uniform Fill color attributes of a selected object. If more than one object is selected, the function returns the Uniform Fill color of the last selected object.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 1 = Pantone 2 = CMYK100 3 = CMYK255 4 = CMY 5 = RGB 6 = HSB 7 = HLS 8 = BW 9 = Gray 11 = YIQ255 12 = LAB 13=Index 14=Pantone Hex 15=Hexachrome To use a color palette with the color model, multiply the palette value (listed below) by 1000 and add it to the color model value. Color palettes: 1 = TRUMATCH 2 = PANTONE PROCESS 3 = PANTONE SPOT 4 = IMAGE 5 = USER 6 = CUSTOMFIXED 7 = RGBSTANDARD 8 = FOCOLTONE 9 = DUPONT 10 = TOYO 11 = DIC 12 = PANTONE HEX 13 = LAB 14 = NETSCAPE 15 = EXPLORER 16 = USERINKS
.Color1	Lets you specify the first color component for .ColorModel. For example, Hue is the first color component for HSB. Click J for valid value ranges.
.Color2	Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.Color3	Lets you specify the third color component for .ColorModel. For example, Saturation is the third color component for HLS. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.
.Color5	Lets you specify the fifth color component for .ColorModel.
.Color6	Lets you specify the sixth color component for .ColorModel.
.Color7	Lets you specify the seventh color component (density)

Example

```
.GetUniformFillColor Model&, C1&, C2&, C3&, C4&, C5&, C6&, C7&
MESSAGE Model&
```

The above example determines the uniform fill color of the selected object and displays a number corresponding to the color model in a message box.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

InsertPaletteColor (DRAW)

.InsertPaletteColor .Index = *long*, .Name = *string*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*, .Color5 = *long*, .Color6 = *long,* .Color7 = *long*

This command inserts a color into the default palette.

Parameter	Description
.Index	Lets you specify the color index.
.Name	Lets you specify the color name.
.ColorModel	Lets you specify the Color Model to use: 1 = Pantone 2 = CMYK100 3 = CMYK255 4 = CMY 5 = RGB 6 = HSB 7 = HLS 8 = BW 9 = Gray 11 = YIQ255 12 = LAB 13=Index 14=Pantone Hex 15=Hexachrome To use a color palette with the color model, multiply the palette value (listed below) by 1000 and add it to the color model value. Color palettes: 1 = TRUMATCH 2 = PANTONE PROCESS 3 = PANTONE SPOT 4 = IMAGE 5 = USER 6 = CUSTOMFIXED 7 = RGBSTANDARD 8 = FOCOLTONE 9 = DUPONT 10 = TOYO 11 = DIC 12 = PANTONE HEX 13 = LAB 14 = NETSCAPE 15 = EXPLORER 16 = USERINKS
.Color1	Lets you specify the first color component for .ColorModel. For example, Hue is the first color component for HSB. Click J for valid value ranges.

.Color2 Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3 Lets you specify the third color component for .ColorModel. For example, Saturation is the third color component for HLS. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color5 Lets you specify the fifth color component for .ColorModel.

.Color6 Lets you specify the sixth color component for .ColorModel.

.Color7 Lets you specify the seventh color component (density)

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

LoadPalette (DRAW)

.LoadPalette .Filename = *string*

This command loads the specifies color palette in to the current document.

Parameter	Description
.Filename	Lets you specify the file name of the color palette you want to load in to your document.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

OverPrintFill (DRAW)

.OverPrintFill

This command overprints the fill of the selected object.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

OverPrintOutline (DRAW)

.OverPrintOutline

This command overprints the outline of the selected object.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

RemoveFountainFillColor (DRAW)

.RemoveFountainFillColor .Position = *long*

This command removes the currently selected Fountain Fill Color.

Parameter	Description
.Position	Lets you specify the position of the color to be removed. 0 and 100 are invalid values. For any other value, the color at that position is removed, if one exists. Existing fill must be a Fountain and Blend must be custom.

Example

```
.ApplyFountainFill 2, -50, -50, 900, 20, 20, 2, 0
.RemoveFountainFillColor 75
```

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

ReplacePaletteColor (DRAW)

.ReplacePaletteColor .Index = *long*, .Name = *string*, .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long*, .Color5 = *long*, .Color6 = *long,* .Color7 = *long*

This command replaces the color in the default palette.

Parameter	Description
.Index	Lets you specify the color index.
.Name	Lets you specify the color name.
.ColorModel	Lets you specify the Color Model to use: 1 = Pantone 2 = CMYK100 3 = CMYK255 4 = CMY 5 = RGB 6 = HSB 7 = HLS 8 = BW 9 = Gray 11 = YIQ255 12 = LAB 13=Index 14=Pantone Hex 15=Hexachrome To use a color palette with the color model, multiply the palette value (listed below) by 1000 and add it to the color model value. Color palettes: 1 = TRUMATCH 2 = PANTONE PROCESS 3 = PANTONE SPOT 4 = IMAGE 5 = USER 6 = CUSTOMFIXED 7 = RGBSTANDARD 8 = FOCOLTONE 9 = DUPONT 10 = TOYO 11 = DIC 12 = PANTONE HEX 13 = LAB 14 = NETSCAPE 15 = EXPLORER 16 = USERINKS
.Color1	Lets you specify the first color component for .ColorModel. For example, Hue is the first color component for HSB. Click J for valid value ranges.

.Color2 Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3 Lets you specify the third color component for .ColorModel. For example, Saturation is the third color component for HLS. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color5 Lets you specify the fifth color component for .ColorModel.

.Color6 Lets you specify the sixth color component for .ColorModel.

.Color7 Lets you specify the seventh color component (density)

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

SavePalette (DRAW)

.SavePalette .Filename = *string*, .Overwrite = *boolean*

This command saves the current color palette.

Parameter	Description
.Filename	Lets you specify the file name for the color palette.
.Overwrite	Set to TRUE (-1) overwrites the existing palette.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

SetOutlineArrow (DRAW)

.SetOutlineArrow .ArrowType = *long*

This command changes the arrowhead of the outline of the selected object. This command follows a block of arrowhead creation commands (see example).

Parameter	Description
.ArrowType	Lets you specify the arrow position. 0 = left arrow 1 = right arrow 2 = both arrows

Example

```
.BeginDrawArrow TRUE, 0, 7
.AddArrowPoint -1016000, 0, FALSE, FALSE, TRUE, TRUE, 0, 0
.AddArrowPoint -481838, -271272, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 571500, -271272, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 31750, 0, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint 571500, 299974, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint -481838, 299974, FALSE, FALSE, TRUE, FALSE, 0, 1
.AddArrowPoint -1016000, 0, FALSE, FALSE, FALSE, TRUE, 0, 1
.EndDrawArrow
.SetOutlineArrow 0
```

The above example adds an arrowhead to the selected line.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

SetOutlineColor (DRAW)

.SetOutlineColor

This command sets the color to be applied to the outline.

Example

```
.StoreColor DRAW_COLOR_CMYK100, 0, 0, 255, 0
.SetOutlineColor
```

The above example sets the outline color to yellow.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

SetOutlineMiscProperties (DRAW)

.SetOutlineMiscProperties .Type = *long,*.Style = *long,* .Corners = *long,* .LineCaps = *long,* .Aspect = *long,* .Angle = *long,* .BehindFill = *Boolean,* .ScalePen = *boolean*

This command sets the outline properties of the selected object.

Parameter	Description
.Type	Lets you specify the outline type: 0 = None 1 = Solid 2 = Dot - Dash
.Style	Calls .DotDash from the ApplyOutline command.
.Corners	0 = Mitered
	1 = Beveled
	2 = Rounded
.LineCaps	0 = Square
	1 = Rounded
	2 = Extended Square
.Aspect	Lets you specify the stretch field which adjusts the width of the nib. Valid values range from 1 to 100 percent.
.Angle	Lets you specify the angle of the nib's edge, in tenths of degrees.
.BehindFill	Set to TRUE (-1) to position the outline behind the fill. Set to FALSE (0) to position the outline in front of the fill.
.ScalePen	Set to TRUE (-1) to scale the outline when the object is scaled.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

SetOutlineWidth (DRAW)

.SetOutlineWidth .Width = *long*

This command changes the width of the outline of the selected object.

Parameter	Description
.Width	Lets you specify the width of the outline.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

StoreColor (DRAW)

.StoreColor .ColorModel = *long*, .Color1 = *long*, .Color2 = *long*, .Color3 = *long*, .Color4 = *long,*.Color5 = *long,* .Color6 = *long,* .Color7 = *long*

This command sets the color that will be applied to the selected object.

Parameter	Description
.ColorModel	Lets you specify the Color Model to use: 1 = Pantone 2 = CMYK100 3 = CMYK255 4 = CMY 5 = RGB 6 = HSB 7 = HLS 8 = BW 9 = Gray 11 = YIQ255 12 = LAB 13=Index 14=Pantone Hex 15=Hexachrome To use a color palette with the color model, multiply the palette value (listed below) by 1000 and add it to the color model value. Color palettes: 1 = TRUMATCH 2 = PANTONE PROCESS 3 = PANTONE SPOT 4 = IMAGE 5 = USER 6 = CUSTOMFIXED 7 = RGBSTANDARD 8 = FOCOLTONE 9 = DUPONT 10 = TOYO 11 = DIC 12 = PANTONE HEX 13 = LAB 14 = NETSCAPE 15 = EXPLORER 16 = USERINKS
.Color1	Lets you specify the first color component for .ColorModel. For example, Hue is the first color component for HSB. Click J for valid value ranges.

.Color2 Lets you specify the second color component for .ColorModel. For example, Green is the second color component for RGB. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color3 Lets you specify the third color component for .ColorModel. For example, Saturation is the third color component for HLS. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

.Color4	Lets you specify the fourth color component for .ColorModel. For example, Black is the fourth color component for CMYK. Click J for valid value ranges. If this parameter is not available in the Color Model specified, set it to 0.

Example

```
.StoreColor DRAW_COLORMODEL_CMYK255, 0, 255, 0, 0, 0, 0, 0
.SetOutlineColor
```

The above example stores the Magenta color for use in the SetOutlineColor command.

{button ,AL(`OVR1 Fill and outline commands;',0,"Defaultoverview",)} **Related Topics**

Special effects commands

AddEnvelopeEffect (DRAW)

.AddEnvelopeEffect .PresetNumber = *long*, .MappingMode = *long*, .KeepLines = *boolean*

This command adds an envelope to the selected object.

Parameter	Description
.PresetNumber	Lets you specify the envelope preset number. Choose from 1 to 39.
.MappingMode	Lets you specify the mapping mode. 0 = Horinzontal 1 = Original 2 = Putty 3 = Vertical
.KeepLines	Set to TRUE (-1) keeps the lines of the source object. Set to FALSE (0) excludes the lines of the source object.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyBitmapEffect (DRAW)

.ApplyBitmapEffect .EffectID = *long*

This command applies the bitmap effect to the selected image.

Parameter	Description
.EffectID	Lets you specify effect ID.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyBlend (DRAW)

.ApplyBlend .Steps = *boolean*, .NoOfSteps = *long*, .AngleOfRotation = *long*, .Loop = *boolean*, .PathObjectID = *long*, .FullPath = *boolean*, .RotateAll = *boolean*, .ColorWheelMode = *long*, .MapNodeStartObject = *long*, .MapNodeEndObject = *long*, .LinearBlend = *boolean*, .LinearSpacing = *boolean*, .LinkAcceleration = *boolean*, .AccelShapes = *boolean*, .BendLogBase = *long*, .SpacingLogBase = *long*, .BlendID = *long*, .BlendType = *long*

This command applies a blend to two selected objects. The parameters below correspond to the controls in the Blend Roll-Up.

Parameter	Description
.Steps	Set to TRUE (-1) to set the number of steps. Set to FALSE (0) if the blend is on a path and you want to use fixed spacing along that path.
.NoOfSteps	Lets you specify the number of intermediate steps.
.AngleOfRotation	Lets you specify the rotation of the intermediate steps in millionths of a degree (e.g., 5000000 = 5 degrees).
.Loop	Set to TRUE (-1) to enable the loop option. Set to FALSE (0) to disable the loop option.
.PathObjectID	Lets you specify the object ID of the path object. Use .GetObjectsCDRStaticID to get an object's ID.
.FullPath	Set to TRUE (-1) to enable the blend along full path option. Set to FALSE (0) to disable the blend along full path option.
.RotateAll	Set to TRUE (-1) to enable the rotate all objects option. Set to FALSE (0) to disable the rotate all objects option.
.ColorWheelMode	0 = straight 1 = clockwise 2 = counter-clockwise
.MapNodeStartObject	Lets you specify a node on the start object to map to a specific node on the end object. The value can range from 0 (the first node) to the number of nodes minus 1.
.MapNodeEndObject	Lets you specify a node on the end object to map to a specific node on the start object. The value can range from 0 (the first node) to the number of nodes minus 1.
.LinearBlend	Set to TRUE (-1) to enable the rotate all objects option. Set to FALSE (0) to disable the rotate all objects option.
.LinearSpacing	Set to TRUE (-1) to enable the rotate all objects option. Set to FALSE (0) to disable the rotate all objects option.
.LinkAcceleration	Set to TRUE (-1) to link the blend acceleration options.
.AccelShapes	Set to TRUE (-1) to accelerate the change in size between the start and end objects.
.BendLogBase	Lets you specify the rate of color acceleration.
.SpacingLogBase	Lets you specify the rate of spacing acceleration.
.BlendID	Reserved for future use.
.BlendType	Reserved for future use.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyContour (DRAW)

.ApplyContour .ContourType = *long*, .Offset = *long*, .Steps = *long*, .ColorWheelDirection = *long*

This command applies a contour to the selected object.

Parameter	Description
.ContourType	0 = To Center 1 = Inside 2 = Outside
.Offset	Lets you specify the distance between contours in tenths of a micron.
.Steps	Lets you specify the number of steps.
.ColorWheelDirection	0 = straight 1 = clockwise 2 = counter-clockwise

Note

- You can include the SCPCONST.CSI and DRWCONST.SCI files in your script. These files define contants for all referenced parameters used with the .ApplyContour command.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyDistortion (DRAW)

.ApplyDistortion .New= *boolean*, .DistortionType = *long*, .Amplitude = *long*, .Frequency = *long*, .Seed = *long*, .Angle = *long*, .CenterX = *long*, .CenterY = *long*, Flag = *long*

This command applies a contour to the selected object.

Parameter	Description
.New	Set to TRUE will add to existing distortion. Set to FALSE clears the existing distortion.
.DistortionType	1=Push & Pull
	2=Zipper
	3=Twister
.Amplitude	1=-200-200
	2=0-100
	3=N/A
.Frequency	1=N/A
	2=Peaks/Curve
	3=N/A
.Seed	If the .Random flag is set then .Seed sets the number of seeds for the zipper distortion.
.Angle	1=N/A
	2=N/A
	3=the maximum twist angle
.CenterX	Lets you specify the offset in X from the center of the distortion.
.CenterY	Lets you specify the offset in Y from the center of the distortion.
.Flag	1=the smoothness of the zipper
	2=the random amplitude of the zipper
	4=set the acceleration of the distortion towards the center of the zipper.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyDropShadow (DRAW)

.ApplyDropShadow .HorizOffset = *long*, .VertOffset = *long*, .Opacity = *long*, .Feather = *long*, .FeatherType = *long*, .FeatherEdge = *long*

This command applies a drop shadow to the selected object.

Parameter	Description
.HorizOffset	Lets you specify the horizontal offset in tenths of a micron.
.VertOffset	Lets you specify the vertical offset in tenths of a micron.
.Opacity	Lets you specify the shadow opacity from 0 to 100.
.Feather	Lets you specify the shadow feathering from 0 to 100.
.FeatherType	0=Inside
	1=Middle
	2=Outside
	3=Average
.FeatherEdge	0=Linear
	1=Square
	2=Flat
	3=Inverse square

Example

```
.StoreColor DRAW_COLORMODEL_RGB&, 0, 0, 0 'Make a black drop shadow
```

```
.ApplyDropShadow LENGTHCONVERT(LC_INCHES, LC_TENTHS_OFA_MICRON, 1), \\LENGHTCONVERT(LC_INCHES, LC_TENTHS_OFA_MICRON, 1), 50, 100, 0, 0
```

The above example applies a drop shadow.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyEnvelopeFrom (DRAW)

.ApplyEnvelopeFrom .ObjectID = *long*, .Mappingmode = *long*, .KeepLines = *boolean*

This command applies an envelope to the selected object from the shape of another object.

Parameter	Description
.ObjectID	Lets you specify the object ID of the source object. Use .GetObjectsCDRStaticID to get an object's ID.
.Mappingmode	0 = Horizontal 1 = Original 2 = Putty 3 = Vertical
.KeepLines	Set to TRUE (-1) to keep the lines of the source object. Set to FALSE (0) to exclude these lines.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyExtrude (DRAW)

.ApplyExtrude .ExtrudeType = *long*, .VPProperties = *long*, .CopyObjectID = *long*, .Depth = *long*, .VPHorizPos = *long*, .VPVertPos = *long*, .PageOrigin = *boolean*, .Light1Pos = *long*, .Light1Intensity = *long*, .Light2Pos = *long*, .Light2Intensity = *long*, .Light3Pos = *long*, .Light3Intensity = *long*, .FillType = *long,* .DrapeFill = *long*

This command extrudes the selected object.

Parameter	Description
.ExtrudeType	0 = Small Back 1 = Small Front 2 = Big Back 3 = Big Front 4 = Back Parallel 5 = Front Parallel
.VPProperties	0 = Vanishing Point locked to object 1 = Vanishing Point locked to page 2 = Copy VP from object (specified with .lCopyObjectID) 3 = Shared VP (specified with .lCopyObjectID)
.CopyObjectID	Lets you specify the object ID of the source object for shared and copied vanishing points. Use .GetObjectsCDRStaticID to get an object's ID.
.Depth	Lets you specify the depth of the extrusion.
.VPHorizPos	Lets you specify the X-coordinate of the vanishing point in tenths of a micron, relative to the center of the page or the object depending on .bPageOrigin.
.VPVertPos	Lets you specify the Y-coordinate of the vanishing point in tenths of a micron, relative to the center of the page or the object depending on .bPageOrigin.
.PageOrigin	Set to TRUE (-1) to position the vanishing point using absolute page coordinates. Set to FALSE (0) to position the vanishing point relative to the object's position.
.Light1Pos	Lets you specify the position of the light source. Valid values range from 0 to 16.
.Light1Intensity	Lets you specify the intensity of the light source. Valid values range from 0 to 100.
.Light2Pos	Lets you specify the position of the light source. Valid values range from 0 to 16.
.Light2Intensity	Lets you specify the intensity of the light source. Valid values range from 0 to 100.
.Light3Pos	Lets you specify the position of the light source. Valid values range from 0 to 16.
.Light3Intensity	Lets you specify the intensity of the light source. Valid values range from 0 to 100.
.FillType	0 = Object fill 1 = Solid fill 2 = Shade
.DrapeFill	Set to TRUE will drape the fill over the object.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyFountainBitmapLens (DRAW)

.ApplyFountainBitmapLens .Type = *long*,.CenterX = *long*, .CenterY = *long*, .Angle = *long*, .Steps = *long*, .Padding = *long*, .Blend = *long*, .Rate = *long*, .Starting = *long*, .Operation = *long*,.Freeze = *boolean*

This command applies a fountain bitmap lens to the selected object.

Parameter	Description
.Type	Lets you specify the fill type: 0 = Linear 1 = Radial 2 = Conical 3 = Square
.CenterX	Lets you specify the horizontal offset to the center of the fill. Choose between -100 to 100.
.CenterY	Lets you specify the vertical offset to the center of the fill. Choose between -100 to 100.
.Angle	Lets you specify the angle of the fill in tenths of a degree.
.Steps	Lets you specify the number of steps in the fill.
.Padding	Lets you specify the amount of padding to apply to the fill.
.Blend	Lets you specify the blending type: 0 = Direct 1 = Rainbow clockwise 2 = Rainbow counterclocwise 3 = Custom
.Rate	Lets you specify the mid-point to apply between the fill colors. Choose from 0 to 100.
.Starting	Lets you specify the starting transparency.
.Operation	Lets you specify which operation to perform on the selected object.
.Freeze	Set to TRUE (-1) freezes the object.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyLensEffect (DRAW)

.ApplyLensEffect .LensType = *long*, .Frozen = *boolean*, .RemoveFace = *boolean*, .ViewPoint = *boolean*, .VpX = *long*, .VpY = *long*, .Param1 = *long*

This command adds a lens to the selected object.

Parameter	Description
.LensType	0 = No Lens Effect 1 = Brighten 2 = Color Add 3 = Color Limit 4 = Custom Color Map 5 = Fish Eye 6 = Heat Map 7 = Invert 8 = Magnify 9 = Tinted Grayscale 10 = Transparancy 11 = Wireframe
.Frozen	Set to TRUE (-1) to enable the Frozen option.
.RemoveFace	Set to TRUE (-1) to enable the Remove Face option.
.ViewPoint	Set to TRUE (-1) to enable the View Point option.
.VpX	Lets you specify the X-coordinate of the view point in tenths of a micron.
.VpY	Lets you specify the Y-coordinate of the view point in tenths of a micron.
.Param1	This value will vary depending on the selected lens. Refer to the Lens Roll-Up for more information

Note

- You can include the SCPCONST.CSI and DRWCONST.SCI files in your script. These files define contants for all referenced parameters used with the .ApplyLensEffect command.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyPerspectiveEffect (DRAW)

.ApplyPerspectiveEffect .Handle = *long*, .PosX = *long*, .PosY = *long*

This command adds a perspective effect to the selected object.

Parameter	Description
.Handle	Lets you specify the handle of the object that is repositioned to create the perspective effect. Valid values range from 1 to 5.
.PosX	Lets you specify the X-coordinate of the new position of the handle, in tenths of a micron
.PosY	Lets you specify the Y-coordinate of the new position of the handle, in tenths of a micron

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyPresetEnvelope (DRAW)

.ApplyPresetEnvelope .PresetNumber = *long*, .Mappingmode = *long*, .KeepLines = *boolean*

This command applies a preset envelope to the selected object.

Parameter	Description
.PresetNumber	Lets you specify the preset envelope to use. Refer to the Preset Roll-Up to see which presets are available. Valid values range from 1 to 39.
.Mappingmode	0 = Horizontal 1 = Original 2 = Putty 3 = Vertical
.KeepLines	Set to TRUE (-1) to keep the lines of the envelope. Set to FALSE (0) to exclude these lines.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyRotatedExtrude (DRAW)

.ApplyRotatedExtrude .ExtrudeType = *long*, .Depth = *long*, .XRotation = *long*, .YRotation = *long*, .ZRotation = *long*, .VPHorizPos = *long*, .VPVertPos = *long*, .PageOrigin = *boolean*, .Light1Pos = *long*, .Light1Intensity = *long*, .Light2Pos = *long*, .Light2Intensity = *long*, .Light3Pos = *long*, .Light3Intensity = *long*, .FillType = *long,* .DrapeFill = *long*

This command applies a rotated extrusion to the selected object.

Parameter	Description
.ExtrudeType	0 = Small Back 1 = Small Front 2 = Big Back 3 = Big Front 4 = Back Parallel 5 = Front Parallel
.Depth	Lets you specify the depth of the extrusion.
.XRotation	Lets you specify the rotation value for the X-axis. Valid values range from 0 to 100.
.YRotation	Lets you specify the rotation value for the Y-axis. Valid values range from 0 to 100.
.ZRotation	Lets you specify the rotation value for the Z-axis. Valid values range from 0 to 100.
.VPHorizPos	Lets you specify the X-coordinate of the vanishing point, in tenths of a micron, relative to the center of the page or the object depending on .bPageOrigin.
.VPVertPos	Lets you specify the Y-coordinate of the vanishing point, in tenths of a micron, relative to the center of the page or the object depending on .bPageOrigin.
.PageOrigin	Set to TRUE (-1) to position the vanishing point using absolute page coordinates. Set to FALSE (0) to position the vanishing point relative to the object's position.
.Light1Pos	Lets you specify the position of the light source. Valid values range from 0 to 16.
.Light1Intensity	Lets you specify the intensity of the light source. Valid values range from 0 to 100.
.Light2Pos	Lets you specify the position of the light source. Valid values range from 0 to 16.
.Light2Intensity	Lets you specify the intensity of the light source. Valid values range from 0 to 100.
.Light3Pos	Lets you specify the position of the light source. Valid values range from 0 to 16.
.Light3Intensity	Lets you specify the intensity of the light source. Valid values range from 0 to 100.
.FillType	0 = Object fill 1 = Solid fill 2 = Shade
.DrapeFill	Set to TRUE will drape the fill over the object.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyTextureBitmapLens (DRAW)

.ApplyTextureBitmapLens .TextureLibrary = *string*,.TextureName = *string*, .TextureStyle = *string*, TextureWidth = *long*, .TextureHeight = *long*, .TextureOffsetX = *long*, .TextureOffsetY = *long*, .RowOffset = *boolean*, .RowColumnOffset = *long*, .ScaleWithObject = boolean, .RotationAngle = *long*, .SkewAngle = *long,* .Starting = *long,* .Ending = *long,* .Operation = *long,* .Freeze = *long*

This command applies a texture bitmap lens to the selected object.

Parameter	Description
.TextureLibrary	Lets you specify the name of the texture library.
.TextureName	Lets you specify the name of the texture.
.TextureStyle	Lets you specify the name of the texture style.
.TextureWidth	Lets you specify the width of the tile in tenths of a micron.
.TextureHeight	Lets you specify the height of the tile in tenths of a micron.
.TextureOffsetX	Lets you specify the amount to offset the texture horizontally.
.TextureOffsetY	Lets you specify the amount to offset the texure vertically.
.RowOffset	Set to TRUE (-1) enables the row offset. Set to FALSE (0) enables the column offset.
.RowColumnOffset	Lets you specify the amount to offset the row or column in tenths of a micron.
.ScaleWithObject	Set to TRUE (-1) transforms the fill when the object is transformed.
.RotationAngle	Lets you specify the amount that the tile is rotated in millionths of degrees.
.SkewAngle	Lets you specify the amount that the tile is skewed in millionths of degrees.
.Starting	Lets you specify the starting transparency.
.Operation	Lets you specify which operation to perform on the selected object.
.Freeze	Set to TRUE (-1) freezes the object.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyTwoColorBitmapLens (DRAW)

.ApplyTwoColorBitmapLens .FileName = *filename*,.TileWidth = *long*, .TileHeight = *long*, .FirstTileOffsetX = *long*, .FirstTileOffsetY = *long*, .RowOffset = *boolean*, .RowColumnOffset = *long*, .SeamlessTiling = *boolean*, .ScaleWithObject = boolean, .RotationAngle = *long*, .SkewAngle = *long,* .Starting = *long,* .Ending = *long,* .Operation = *long,* .Freeze = *long*

This command applies a two-color bitmap lens to the selected object.

Parameter	Description
.FileName	Lets you specify the path and filenameof the two-color fill file to use.
.TileWidth	Lets you specify the width of the tile in units specified by ScaleWithObject command.
.TileHeight	Lets you specify the height of the tile in units specifies by the ScaleWithObject command.
.FirstTileOffsetX	Lets you specify the amount to offset the first tile horizontally.
.FirstTileOffsetY	Lets you specify the amount to offset the first tile vertically.
.RowOffset	Set to TRUE (-1) enables the row offset. Set to FALSE (0) enables the column offset.
.RowColumnOffset	Lets you specify the amount to offset the row or column. Choose between 0 and 100.
.Seamless Tiling	Set to TRUE (-1) enables seamless tiling.
.ScaleWithObject	Set to TRUE (-1) sets the height/width units to a tenth of a micron. Set to FALSE (0) set the measurement as a percentage.
.RotationAngle	Lets you specify the amount that the tile is rotated in millionths of degrees.
.SkewAngle	Lets you specify the amount that the tile is skewed in millionths of degrees.
.Starting	Lets you specify the starting transparency.
.Operation	Lets you specify which operation to perform on the selected object.
.Freeze	Set to TRUE (-1) freezes the object.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ApplyUniformBitmapLens (DRAW)

.ApplyUniformBitmapLens .Starting = *long*, .Operation = *long*, .Freeze = *boolean*

This command applies a uniform bitmap lens to the selected object.

Parameter	Description
.Starting	Lets you specify the starting transparency.
.Operation	Lets you specify which operation to perform on the selected object.
.Freeze	Set to TRUE (-1) freezes the object.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ClearEffect (DRAW)

.ClearEffect

This command removes a special effect from the selected object.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ConvertBitmapTo (DRAW)

.ConvertBitmapTo .EffectID = *long*

This command converts the selected bitmap to another bit depth.

Parameter	Description
.EffectID	Lets you specify ID.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ConvertToBitmap (DRAW)

.ConvertToBitmap .BitDepth = *long*, .Grayscale = *boolean*, .Dithered = *boolean*, .TransparentBG = *boolean*, .Resolution = *long*, .AntiAliasing = *long*, .UseColorProfile = *boolean*

This command converts a vector object to a bitmap.

Parameter	Description
.BitDepth	Lets you specify bit depth.
.Grayscale	Set to TRUE (-1) convert to grayscale.
.Dithered	Set to TRUE (-1) enables dithering.
.TransparentBG	Set to TRUE (-1) enables transparent background.
.Resolution	Lets you specify the resolution.
.AntiAliasing	Lets you specify the anti aliasing: 0 = None 1 = Normal 2 = Super Sampling
.UseColorProfile	Set to TRUE (-1) use the color profile.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

CopyPowerClip (DRAW)

.CopyPowerClip .Index = *long*, .XDisp = *long*, .YDisp = *long*

This command copies the PowerClip of the object to the current selection.

Parameter	Description
.Index	Lets you specify the object ID.
.XDisp	Lets you specify the X coordinate of the PowerClip in the object to copy.
.YDisp	Lets you specify the Y coordinate of the PowerClip in the object to copy.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

CopyEffectFrom (DRAW)

.CopyEffectFrom .Clone = *boolean*, .SourceObjectID = *long*

This command copies an effect from a specific object to the selected object.

Parameter	Description
.Clone	Set to TRUE (-1) to clone the effect instead of copying it. This creates a link between the two objects. When the effect is changed for one object, the other object also changes.
.SourceObjectID	Lets you specify the object ID of the source object. Use .GetObjectsCDRStaticID to get an object's ID.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

DetachBlendPath (DRAW)

.DetachBlendPath

This command detaches a path from a blend group.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ExtractContents (DRAW)

.ExtractContents

This command extracts the contents of the selected PowerClip.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

FuseBlend (DRAW)

.FuseBlend .End = *boolean*, .PositionX = *long*, .PositionY = *long*

This command fuses a split blend group.

Parameter	Description
.End	Set to TRUE (-1) to fuse the top of the blend. Set to FALSE (0) to fuse the bottom of the blend.
.PositionX	Lets you specify the X-coordinate of the point where you want the blend to be fused, in tenths of a micron, relative to the center of the page.
.PositionY	Lets you specify the Y-coordinate of the point where you want the blend to be fused, in tenths of a micron, relative to the center of the page.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

GetBitmapResolution (DRAW)

.GetBitmapResolution .XRes = *long*, .YRes = *long*

This command returns the bitmap resolution.

Parameter	Description
.XRes	Lets you specify the horizontal resolution.
.YRes	Lets you specify the vertical resolution.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

GetBitmapSize (DRAW)

.GetBitmapSize .Width = *long*, .Height = *long*

This command returns the bitmap size.

Parameter	Description
.Width	Lets you specify the bitmap width.
.Height	Lets you specify the bitmap height.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

InflateBitmap (DRAW)

.InflateBitmap .Width = *long*, .Height = *long*, .InflationType = *long*

This command inflates a bitmap.

Parameter	Description
.Width	Lets you specify the bitmap width.
.Height	Lets you specify the bitmap height.
.InflationType	Lets you specify the inflation type: 0 = Size 1 = incrementation 2 = Percentage

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

IsBitmapExternallyLink (DRAW)

.IsBitmapExternallyLink

This command returns true if the bitmap is externally linked.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

PlaceInside (DRAW)

.PlaceInside .Index = *long*, .XDisp = *long*, .YDisp = *long,* .ForceCenter = *boolean,* .Center = *boolean*

This command will PowerClip the current selection.

Parameter	Description
.Index	Lets you specify the object ID.
.XDisp	Lets you specify the X coordinate in the object to PowerClip.
.YDisp	Lets you specify the Y coordinate in the object to PowerClip.
.ForceCenter	Set to TRUE (-1) will use the Center parameter which centers the object inside the container. Set to FALSE (0) Uses the workspace preferences.
.Center	Set to TRUE (-1) center the objet inside the container.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ResolveAllBitmapsLink (DRAW)

.ResolveAllBitmapsLink

This command resolves all external links.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

ResolveBitmapLink (DRAW)

.ResolveBitmapLink

This command resolves any external links.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

SplitBlend (DRAW)

.SplitBlend .PositionX = *long*, .PositionY = *long*

This command splits the selected blend group.

Parameter	Description
.PositionX	Lets you specify the X-coordinate of the point where you want the blend to be split, in tenths of a micron, relative to the center of the page.
.PositionY	Lets you specify the Y-coordinate of the point where you want the blend to be split, in tenths of a micron, relative to the center of the page.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

StartEditContents (DRAW)

.StartEditContents

This command start to edit the contents of the selected PowerClip.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

StopEditContents (DRAW)

.StopEditContents

This command stops to edit the contents of the selected PowerClip.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

UpdateBitmapLink (DRAW)

.UpdateBitmapLink

This command updates any external links.

{button ,AL(`OVR1 Special effects commands;',0,"Defaultoverview",)} **Related Topics**

Object Data Manager commands

AddObjectDataField (DRAW)

.AddObjectDataField.FieldName = *string*

This function adds an object data field. The new object data field must be followed by its definition. See DefineObjectDataField.

Parameter	Description
.FieldName	Lets you specify the name of object data field.

Example

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

ClearAllObjectData (DRAW)

.ClearAllObjectData

This function clears all the fields and data from the target object.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

ClearObjectData (DRAW)

.ClearObjectData .FieldName = *string*

This function clears a field and data from the selected object.

Parameter	Description
.FieldName	Lets you specify the name of the objects data field.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

CopyObjectDataFields (DRAW)

.CopyObjectDataFields.Index = *long*

This function copies all data fields from the target object.

Parameter	Description
.Index	Lets you specify the object ID.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

CreateObjectDataField (DRAW)

.CreateObjectDataField .FieldName = *string,* .Format = *string,* .Order = *long,* .ColumnWidth = *long,* .DrawDefault = *boolean,* .DocDefault = *boolean,* .SummarizeGroup = *boolean*

This function creates a new object data field.

Parameter	Description
.FieldName	Lets you specify the name of object data field.
.Format	Lets you specify the format for the field.
.Order	Lets you specify the position in the field.
.ColumnDefault	Lets you specify the width of the column.
.DrawDefault	Set to TRUE (-1) object is saved in Coreldrw.ini.
.DocDefault	Set to TRUE (-1) object is added to all objects.
.SummarizeGroup	Set to TRUE (-1) to summarize groups.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

DefineObjectDataField (DRAW)

.DefineObjectDataField .FieldName = *string,*.Format = *string,*.DrawDefault = *boolean,*.DocDefault = *boolean,*.SummarizeGroup = *boolean*

This function defines the properties of an object data field.

Parameter	Description
.FieldName	Lets you specify the name of object data field.
.Format	Lets you specify the name of the field format.
.DrawDefault	Set to TRUE (-1) object is saved in Coreldrw.ini.
.DocDefault	Set to TRUE (-1) object is added to all objects.
.SummarizeGroup	Set to TRUE (-1) to summarize groups.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

DeleteObjectDataField (DRAW)

.DefineObjectDataField .FieldName = *string*

This function deletes the object data field.

Parameter	Description
.FieldName	Lets you specify the name of object data field.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

GetObjectData (DRAW)

ReturnString$ = .GetObjectData(.FieldName = *string*)

This function returns a specified object data field of a selected object. If more than one object is selected, the function returns the specified object-data field of the last selected object.

Parameter	Description
ReturnString$	Returns the object data of the selected object.
.FieldName	Lets you specify the name of object data field.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

GetUserDataField (DRAW)

ReturnString$ = .GetUserDataField(.FieldName = *string*)

This function returns a specified user-data field of a selected object. If more than one object is selected, the function returns the specified user-data field of the last selected object.

Parameter	Description
ReturnString$	Returns the user data field of the selected object.
.FieldName	Lets you specify the name of an object's user data field.

Example

```
u_d_f$="CDRStaticID"
data_field1$=.GetUserDataField (u_d_f)
data_field2$=.GetUserDataField ("Name")
```

The above example returns the value for the CDRStaticID and Name field of a selected object.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

OrderObjectDataFields (DRAW)

.OrderObjectDataFields .FieldToMove = *string,* .NewPosition = *long*

This function changes the order of the object data fields.

Parameter	Description
.FieldToMove	Lets you specify the name of the field to move.
.NewPosition	Lets you specify the position to place the field.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

PasteObjectData (DRAW)

.PasteObjectData .Index = *long,* .FieldName = *string*

This function pastes the contents from the Clipboard into a field.

Parameter	Description
.Index	Lets you specify the object ID.
.FieldName	Lets you specify the name of the object data field.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

RenameObjectDataField (DRAW)

.RenameObjectDataField .OldFieldName = *string*, .NewFieldName = *string*

This command lets you rename an object data field.

Parameter	Description
.OldFieldName	Lets you specify the old name of the object data field.
.NewFieldName	Lets you specify the new name of the object data field.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

SetObjectData (DRAW)

.SetObjectData .FieldName = *string*, .FieldValue = *string*

This command lets you set object data values for selected objects.

Parameter	Description
.FieldName	Lets you specify the name of the object data field to set.
.FieldValue	Lets you specify the value of the object data field to set.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

SetUserDataField (DRAW)

.SetUserDataField .FieldName = *string*, .FieldValue = *string*

This command lets you set user data field values for selected objects.

Parameter	Description
.FieldName	Lets you specify the name of the user data field to set.
.FieldValue	Lets you specify the value of the user data field to set.

Example

```
.CreateRectangle 1000000, 750000, 500000, 100000, 0
.SetUserDataField "Name", "MyObject"
```

The above example creates a rectangle and while it is still selected, sets its object name to "MyObject". Other common data fields for objects include cost and comments.

{button ,AL(`OVR1 Object Data Manager commands;',0,"Defaultoverview",)} **Related Topics**

Recorder commands

RecorderBeginEditText (DRAW)

.RecorderBeginEditText

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEndEditText (DRAW)

.RecorderEndEditText

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEditTextCharAttributes (DRAW)

.RecorderEditTextCharAttributes .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .FontName = *string*, .FontStyle = *long*, .PointSize = *long*, .Underline = *long*, .Overline = *long*, .StrikeOut = *long*, .Placement = *long*, .CharacterSpacing = *long*, .WordSpacing = *long*, .LineSpacing = *long*, .Alignment = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEditTextChangeCase (DRAW)

.RecorderEditTextChangeCase .CaseID = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEditReplaceText (DRAW)

.RecorderEditTextReplaceText .NewText = *string*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderBeginEditParaText (DRAW)

.RecorderBeginEditParaText

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEndEditParaText (DRAW)

.RecorderEndEditParaText

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEditParaTextChangeCase (DRAW)

.RecorderEditParaTextChangeCase .CaseID = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEditParaTextSpacing (DRAW)

.RecorderEditParaTextSpacing .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .CharacterSpacing = *long*, .WordSpacing = *long*, .LineSpacing = *long*, .BeforeParagraph = *long*, .AfterParagraph = *long*, .Alignment = *long*, .AutoHyphenation = *boolean*, .HyphenHotZone = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEditParaTextIndents (DRAW)

.RecorderEditParaTextIndents .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .FirstLine = *long*, .RestOfLines = *long*, .RightMargin = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEditParaTextCharAttributes (DRAW)

.RecorderEditParaTextCharAttributes .FirstSelectedChar = *long*, .LastSelectedChar = *long*, .FontName = *string*, .FontStyle = *long*, .PointSize = *long*, .Underline = *long*, .Overline = *long*, .StrikeOut = *long*, .Placement = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderEditParaReplaceText (DRAW)

.RecorderEditParaTextReplaceText .NewText = *string*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderApplyPerspective (DRAW)

.RecorderApplyPerspective .Type = *long*, .Flags = *long*, .Box0X = *long*, .Box0Y = *long*, .Box1X = *long*, .Box1Y = *long*, .Box2X = *long*, .Box2Y = *long*, .Box3X = *long*, .Box3Y = *long*, .VPHorizRef = *long*, .VPHorizX = *long*, .VPHorizY = *long*, .VPVertRef = *long*, .VPVertX = *long*, .VPVertY = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderObjectScaleInfo (DRAW)

.RecorderObjectScaleInfo .ScaledSizeX = *long*, .ScaledSizeY = *long*, .DisplacementX = *long*, .DisplacementY = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderSelectObjectByIndex (DRAW)

.RecorderSelectObjectByIndex .ClearFirst = *boolean*, .IIndex = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderSelectObjectsByIndex (DRAW)

.RecorderSelectObjectsByIndex .ClearFirst = *boolean*, .Index1 = *long*, .Index2 = *long*, .Index3 = *long*, .Index4 = *long*, .Index5 = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderSelectPreselectedObjects (DRAW)

.RecorderSelectPreselectedObjects .ClearFirst = *boolean*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

RecorderStorePreselectedObjects (DRAW)

.RecorderStorePreselectedObjects .ConvertingPreset = *boolean*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

StartOfRecording (DRAW)

.StartOfRecording

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

EndOfRecording (DRAW)

.EndOfRecording

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

MenuCommand (DRAW)

.MenuCommand .MenuID = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

ClickedDialogButton (DRAW)

.ClickedDialogButton .DialogID = *long*, .ItemD = *long*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

ShareExtrudeVP (DRAW)

.ShareExtrudeVP .ExtrudeIndex = *long*, .VPToShareIndex = *long*, .SharedVP = *boolean*

This command is used by the Script And Preset Manager. You do not need to include this command in your script.

{button ,AL(`OVR1 Recorder commands;',0,"Defaultoverview",)} **Related Topics**

ID	Color Model	Color 1	Color 2	Color 3	Color 4
1	Pantone	Pantone ID number	Tint (0 - 100)	Ignored	Ignored
2	CMYK100	Cyan (0 - 100)	Magenta (0 - 100)	Yellow (0 - 100)	Black (0 - 100)
3	CMYK255	Cyan (0 - 255)	Magenta (0 - 255)	Yellow (0 - 255)	Black (0 - 255)
4	CMY	Cyan (0 - 255)	Magenta (0 - 255)	Yellow (0 - 255)	Ignored
5	RGB	Red (0 - 255)	Green (0 - 255)	Blue (0 - 255)	Ignored
6	HSB	Hue (0 - 360)	Saturation (0 - 255)	Brightness (0 - 255)	Ignored
7	HLS	Hue (0 - 360)	Lightness (0 - 255)	Saturation (0 - 255)	Ignored
8	Black and White	Black (0) or White (1)	Ignored	Ignored	Ignored
9	Grayscale	Black % (0-255)	Ignored	Ignored	Ignored
10	YIQ255	Y-luminance (0 - 255)	I-chromaticity (0 - 255)	Q-chromaticity (0 - 255)	Ignored
11	L*a*b*	L*-lightness (0 - 255)	a*-green to red (0 - 255)	b*-blue to yellow (0 - 255)	Ignored