Notice

This specification is provided under the Microsoft Open Specification Promise. For further details on the Microsoft Open Specification Promise, please refer to: http://www.microsoft.com/interop/osp/default.mspx. You are free to copy, display and perform this specification, to make derivative works of this specification, and to distribute the specification, however distribution rights are limited to unmodified copies of the original specification and any redistributed copies of the specification must retain its attribution of Microsoft’s rights in the copyright of the specification, this full notice, and the URL to the webpage containing the most current version of the specification as provided by Microsoft.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in these materials. Except as expressly provided in the Microsoft Open Specification Promise and this notice, the furnishing of these materials does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The information contained in this document represents the point-in-time view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of authoring.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

©2007 Microsoft Corporation. All rights reserved.
Microsoft, Windows, Windows NT, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
Contents
Microsoft Office 97-2007 ......................................................................................... 1
Introduction .................................................................................................................. 7
Object Container Hierarchy .......................................................................................... 7
Use by the Host Application ....................................................................................... 8
Records ......................................................................................................................... 8
Common Header ......................................................................................................... 8
Notes for Implementers .............................................................................................. 9
Pointers ......................................................................................................................... 9
Drawing Group Container ........................................................................................... 13
Drawing Group Record ............................................................................................... 13
Class ID Record ........................................................................................................... 13
Default Property Table Records ............................................................................... 13
Color MRU Record ....................................................................................................... 13
Split Menu Colors Record ......................................................................................... 14
BStore Container ........................................................................................................ 14
BLIP Store Entry Record ............................................................................................. 14
Drawing Container ..................................................................................................... 17
Drawing Record .......................................................................................................... 18
Regroup ......................................................................................................................... 18
Group Container ......................................................................................................... 19
Shape Container ......................................................................................................... 19
Group Shape Record .................................................................................................. 20
Shape Record ............................................................................................................... 20
Property Table Records ............................................................................................. 20
Anchor Record .............................................................................................................. 21
Child Anchor Record .................................................................................................. 21
Textbox Record .......................................................................................................... 21
OLE Object Record ..................................................................................................... 21
Deleted PSPL Record ................................................................................................. 21
Solver Container .......................................................................................................... 22
Connector Rule Record ............................................................................................... 22
Align Rule Record ....................................................................................................... 22
Arc Rule Record .......................................................................................................... 22
Interpreting the “main” property

Appendix A: Change History

Appendix B: Colors

Group Shape 2

Ink

Clip

Web Component

Line Top Style

Line Column Style

Line Bottom Style

Line Right Style

Line Left Style

Unknown HTML

Relative Transform

Shape

Callout

Group Shape

Transform

Protection

Text

GeoText

Blip

Geometry

Fill Style

Line Style

Shadow Style

Perspective Style

3D Object

3D Style

Shape Properties

Callout Rule Record

Microsoft Office Drawing 97-2007 Binary Format Specification
Interpreting "extended" properties ................................. 54
Appendix C: Shape Types .................................................. 57
Appendix D: AutoShapes .................................................... 60
Appendix E: Animation Records ........................................... 143
F123  msofbtTimeNodeContainer ........................................... 143
F124  msofbtTimeConditionList ............................................ 143
F125  msofbtTimeConditionContainer .................................... 143
F126  msofbtTimeModifierList ............................................. 144
F127  msofbtTimeNode (FTN) ................................................ 144
F128  msofbtTimeCondition (FTC) ......................................... 145
F129  msofbtTimeModifier (FTM) .......................................... 146
F12A  msofbtTimeBehaviorContainer ..................................... 146
F12B  msofbtTimeAnimateBehaviorContainer .......................... 147
F12C  msofbtTimeColorBehaviorContainer .............................. 147
F12D  msofbtTimeEffectBehaviorContainer ............................... 147
F12E  msofbtTimeMotionBehaviorContainer ............................ 148
F12F  msofbtTimeRotationBehaviorContainer .......................... 148
F130  msofbtTimeScaleBehaviorContainer ............................... 148
F131  msofbtTimeSetBehaviorContainer ................................ 149
F132  msofbtTimeCommandBehaviorContainer .......................... 149
F133  msofbtTimeBehavior (FTB) ........................................... 149
F134  msofbtTimeAnimateBehavior (FTBA) ................................. 150
F135  msofbtTimeColorBehavior (FTBC) ................................... 151
F136  msofbtTimeEffectBehavior (FTBE) .................................. 152
F137  msofbtTimeMotionBehavior (FTBM) ................................. 152
F138  msofbtTimeRotationBehavior (FTBR) ............................... 153
F139  msofbtTimeScaleBehavior (FTBS) .................................... 154
F13A  msofbtTimeSetBehavior (FTBSet) ................................... 154
F13B  msofbtTimeCommandBehavior (FTBCom) ........................ 155
F13C  msofbtClientVisualElement .......................................... 155
F13D  msofbtTimePropertyList ............................................. 155

A list of TimeVariant objects, defined by msofbtTimeVariant records. The instance field for each TimeVariant record gives you the property ID for the property .................................. 155
F13E  msofbtTimeVariantList ............................................... 156
F13F  msofbtTimeAnimationValueList ..................................... 156
F140  msofbtTimeIterateData FTID ........................................ 156
F141  msofbtTimeSequenceData FTSD ................................................................. 157
F142  msofbtTimeVariant ................................................................. 158
F143  msofbtTimeAnimationValue ................................................................. 158
F144  msofbtExtTimeNodeContainer ................................................................. 159
F145  msofbtSubNodeContainer ................................................................. 159
Appendix F: Shape XML Data ........................................................................ 160
Appendix G: Miscellaneous Enumerated Types and Structures .................. 165
Introduction

This document describes the file format of the Office Drawing Layer (Escher). With this document and knowledge of the host application’s file format, the reader should be able to construct and interpret an Escher file stream.

Object Container Hierarchy

Escher has an object containership hierarchy similar to other drawing programs. At the root of the hierarchy is a drawing group object. There is one drawing group per client document. Drawing groups contain drawings. Drawings in turn contain shapes that are the objects that actually mark a page. Next to the drawings in a drawing group is a collection that contains the images and pictures used by the drawings. Escher keeps pictures in a separate collection to be able to incrementally load and save them.

A few other points are worth noting.

Associated with each shape is a piece of client data that keeps the shape’s anchor, text and OLE data, as well as host specific properties. The format of this structure is host-defined.

Shapes store their properties in a separate structure called a property table. The property table is basically a sorted list of property id-value pairs.

Each drawing group has a shape property table that stores the defaults for new shapes.

Each drawing has a collection of rules that govern the shapes in the drawing.

This object hierarchy does not exactly correspond to the record hierarchy in the file format. In the file format, drawings are not saved inside drawing groups, but in separate top-level containers. In this scheme, hosts can save drawing group information with per-document information, and save drawing information with per-sheet, per-slide, or per-page information.
Use by the Host Application

Escher is part of the Office DLL and is used by PowerPoint, Word, FrontPage, Publisher, and Excel. FrontPage does not save the Office Drawing file format into files; it emits and consumes the Office Drawing file format only when interacting with the clipboard. The other client applications use the Office Drawing file format when saving data into their binary file format. When serializing Escher data into a file, each client application provides an OLE IStream interface to Escher. This document describes the records Escher writes to this interface. However, it doesn’t describe how the actual bytes are saved to disk. The client controls this format, since Escher writes to a client provided interface. Excel, for example, places the bytes of the Escher stream into BIFF records.

Records

The Escher file stream is a series of records that share a common header structure. Records can be categorized into two groups.

- **Atoms**  Records that contain information about an Escher object and are kept inside containers.
- **Containers**  Records that keep atoms and other containers in a logical and organized way.

Each record, whether it's an atom or a container, has a common header. Container records are just the common header, while the atom records are the common header followed by some record specific data. Escher uses the structure of container records containing atom records and other container records throughout its stream.

Common Header

The common record header is an 8-byte structure defined as follows:

```c
typedef struct MSOFBH
{
 struct
 {
     ULONG ver : 4;
     ULONG inst: 12;
     ULONG fbt : 16;
     
 };

     ULONG cbLength;
 } MSOFBH;
```

The fields are:

- **Record Type**  (fbt) Indicates the signature or type of the record. Each record has a symbolic and a numeric signature. Escher uses values from 0xF000 to 0xFFFF. Clients may define their own records in other ranges. A description of each of the different types can be found in the following sections.

- **Record Instance**  (inst) Differentiates atoms. Depending on the instance a record’s contents it can have different meanings. For example a list container can store a list of slides or a list of fonts, and its instance would vary accordingly. The instance of a record is useful for differentiating atoms when there is more than one atom of the same type in a particular container.

- **Record Version**  (ver) Indicates the version if the record is an atom. If the record is a container, this field has a value of 0xFFFF.
Record Length (cbLength) Stores the length of the record in bytes. If the record is an atom, it refers to the length of the atom excluding the header. If the record is a container, it refers to the sum of the lengths of the atoms inside it, plus the length of the record headers.

Notes for Implementers

The common header specifies the length of each record. Consequently, it is possible to parse the Escher record stream without knowledge of the actual contents of each record. The Escher team intends to take advantage of this fact in future versions. As new features are added, Escher will define new record types. Readers of the Escher file format should skip over record types unknown to the reader. In addition, readers should not expect a record to come in a certain order in a container. They can, however, expect that the containership hierarchy will not change. For example, readers do not need to handle the case of a shape record containing a drawing record.

When Escher writes to a client file, it stores client-specific records in its stream to preserve the client features and behaviors. On the other hand, when Escher writes to a clipboard stream, it uses a client-independent form of the file format to allow interchange between applications.

Escher saves records in Intel byte-order even on the Macintosh. The Macintosh version of Escher byte-swaps the records as they are loaded and as they are saved. Records are tightly packed, without alignment. The LONG type is 32 bits in length.

Pointers

The general problem of saving pointers to objects in the file format is solved in ordinary fashion by giving objects unique identifiers, which are saved in the file format in place of the pointer values. At load time, these IDs are converted back into pointers.

The most common instances of this are pointers to shapes, which are saved as shape IDs, or SPIDs. SPIDs are unique per drawing group, and are parceled out by the drawing group to individual drawings in blocks of 1024. The drawing group keeps a table recording which drawing owns which block of SPIDs, so that, given a SPID, it is easy to determine which drawing the shape is in. That table makes up the bulk of the msofbtDgg record, and is the only place where pointers to drawings are saved (as DGIDs).
<table>
<thead>
<tr>
<th>Record Name</th>
<th>Word</th>
<th>Excel</th>
<th>PowerPoint</th>
<th>FBT Value</th>
<th>Version</th>
<th>Instance</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>msofbtDggContainer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F000</td>
<td></td>
<td></td>
<td>per-document data</td>
</tr>
<tr>
<td>msofbtDgg</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F006</td>
<td>0</td>
<td></td>
<td>an FDGG and several FIDCLs</td>
</tr>
<tr>
<td>msofbtCLSID</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>F016</td>
<td>0</td>
<td></td>
<td>the CLSID of the application that put the data on the clipboard</td>
</tr>
<tr>
<td>msofbtOPT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F00B</td>
<td>3</td>
<td></td>
<td>the document-wide default shape properties (Block 1)</td>
</tr>
<tr>
<td>msofbtTertiaryOPT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F122</td>
<td>3</td>
<td></td>
<td>the document-wide default shape properties (Block 3)</td>
</tr>
<tr>
<td>msofbtColorMRU</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F11A</td>
<td>0</td>
<td></td>
<td>the colors in the MRU swatch</td>
</tr>
<tr>
<td>msofbtSplitMenuColors</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F11E</td>
<td>0</td>
<td></td>
<td>the colors in the top-level split menus</td>
</tr>
<tr>
<td>msofbtBstoreContainer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F001</td>
<td></td>
<td></td>
<td>all images in the document (JPEGs, metafiles, etc.)</td>
</tr>
<tr>
<td>msofbtBSE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F007</td>
<td>2</td>
<td></td>
<td>an FBSE (one per BLIP)</td>
</tr>
<tr>
<td>msofbtBlip***</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F018 - F117</td>
<td></td>
<td></td>
<td>range of fbts reserved for various kinds of BLIPs</td>
</tr>
<tr>
<td>msofbtDggContainer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F002</td>
<td></td>
<td></td>
<td>per-sheet/page/slide data</td>
</tr>
<tr>
<td>msofbtDg</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F008</td>
<td>0</td>
<td></td>
<td>an FDG</td>
</tr>
<tr>
<td>msofbtRegroupItems</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F118</td>
<td>0</td>
<td></td>
<td>several FRITs</td>
</tr>
<tr>
<td>msofbtColorScheme</td>
<td>C</td>
<td>C</td>
<td>F120</td>
<td></td>
<td></td>
<td></td>
<td>the colors of the source host's color scheme</td>
</tr>
<tr>
<td>msofbtSpgrContainer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F003</td>
<td></td>
<td></td>
<td>several SpContainers, the first of which is the group shape itself</td>
</tr>
<tr>
<td>msofbtSpContainer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F004</td>
<td></td>
<td></td>
<td>a shape</td>
</tr>
<tr>
<td>msofbtSpgr</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F009</td>
<td>1</td>
<td></td>
<td>an FSPSPGR; only present if the shape is a group shape</td>
</tr>
<tr>
<td>msofbtSp</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F00A</td>
<td>2</td>
<td>shape type</td>
<td>an FSP</td>
</tr>
<tr>
<td>msofbtOPT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F00B</td>
<td>3</td>
<td></td>
<td>a shape property table (Block 1)</td>
</tr>
<tr>
<td>msofbtSecondaryOPT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F121</td>
<td>3</td>
<td></td>
<td>a shape property table (Block 2)</td>
</tr>
<tr>
<td>msofbtTertiaryOPT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F122</td>
<td>3</td>
<td></td>
<td>a shape property table (Block 3)</td>
</tr>
<tr>
<td>msofbtTextbox</td>
<td>C C C #00C 0</td>
<td>RTF text</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtClientTextbox</td>
<td>✓ ✓ ✓ #00D</td>
<td>host-defined</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>the text in the textbox, in a host-defined format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtAnchor</td>
<td>C C C #00E 0</td>
<td>a RECT, in 100000ths of an inch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtChildAnchor</td>
<td>✓ ✓ ✓ #00F 0</td>
<td>a RECT, in units relative to the parent group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtClientAnchor</td>
<td>✓ ✓ ✓ #010</td>
<td>host-defined</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>the location of the shape, in a host-defined format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtClientData</td>
<td>✓ ✓ ✓ #011</td>
<td>host-defined</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>host-specific data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtOleObject</td>
<td>C C C #11F 0</td>
<td>a serialized IStorage for an OLE object</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtDeletedPspl</td>
<td>✓ #11D 0</td>
<td>an FPSPL; only present in top-level deleted shapes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtSolverContainer</td>
<td>✓ ✓ ✓ #005</td>
<td>count of rules</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>the rules governing shapes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtConnectorRule</td>
<td>✓ ✓ #012 1</td>
<td>an FConnectorRule</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtAlignRule</td>
<td>✓ ✓ ✓ #013 0</td>
<td>an FAlignRule</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtArcRule</td>
<td>✓ ✓ ✓ #014 0</td>
<td>an FARCRU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtClientRule</td>
<td>#015 host-defined</td>
<td>host-defined</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtCalloutRule</td>
<td>✓ ✓ ✓ #017 0</td>
<td>an FCORU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msofbtSelection</td>
<td>✓ #119 0</td>
<td>an FDGSL followed by the SPIDs of the shapes in the selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Drawing Group Container

msofbtDggContainer

Drawing Group Record

msofbtDgg

The drawing group record is a variable length record consisting of a fixed part followed by an array. The fixed part is defined as follows.

// FDGG - File DGG
typedef struct _FDGG
{
  MSOSPID spidMax;   // The current maximum shape ID
  ULONG cidcl;       // The number of ID clusters (FIDCLs)
  ULONG cspSaved;    // The total number of shapes saved
  // (including deleted shapes, if undo
  // information was saved)
  ULONG cdgSaved;    // The total number of drawings saved
} FDGG;

The fixed part is followed by an array of ID clusters. The ID clusters are used internally for the translation of shape ids (SPIDs) to shape handles (MSOHSPs).

// File ID Cluster - used to save IDCLs
typedef struct _FIDCL
{
  MSODGID dgid;      // DG owning the SPIDs in this cluster
  ULONG cspidCur;    // number of SPIDs used so far
} FIDCL;

Class ID Record

msofbtCLSID

The class ID record is only present in the clipboard format. It just contains an OLE CLSID record from the source application, and is used by the destination application to check where the clipboard data originated.

Default Property Table Records

msofbtOPT, msofbtTertiaryOPT

This describes the default properties of newly created shapes. Only the properties that differ from the per-property defaults are saved. The format of the record is the same as that of the property table in a shape, except that Block 2 (msofbtSecondaryOPT) is not allowed in the defaults. A discussion of that format is in the Shape Properties section.

Color MRU Record

msofbtColorMRU

The Color MRU record contains the colors in the most-recently-used-colors swatch that appears at the bottom of color dropdowns. The instance field contains the number of colors; the data of the record contains the colors in order from left to right.
Split Menu Colors Record

`msofbtSplitMenuColors`

The single MRU colors of the top-level Fill Color, Line Color, Shadow Color, and 3D Color split menus are saved to a `SplitMenuColors` record in that order, with the number of colors (currently always four) in the instance field.

BStore Container

`msofbtBstoreContainer`

The images and pictures in a drawing can dominate the size of a drawing. Consequently, Escher handles these objects in a special way. As an abstraction, Escher names these objects BLIPs for Big Large Image or Picture. In Office 2003, there are eight types of blips supported in Escher: Windows Metafiles, Enhanced Metafiles, JPEG Interchange Format, Device Independent Bitmap (DIB,) Tag Image File Format (TIFF,) Portable Network Graphics (PNG,) Graphic interchange format (GIF,) and Macintosh PICT. Implementers should note that some of these types cannot be processed by Office 2000, and even more cannot be processed by Office 97. Additional types may be permitted in future versions.

Escher stores all the BLIPs in a document in a separate container called the BStore. It reference counts the BLIPs, so that if a picture is inserted multiple times in a document it is only stored once in the BStore but is multiply referenced by different shapes.

The host may choose to store the blip data in a separate delay stream. If a delay stream is used, Escher can incrementally load the blips as they are displayed, not when the document is loaded. (As of Office 97, Word and PowerPoint use a delay stream, and Excel does not.)

The BStore container is just an array of Blip Store Entry (BSE) records. Each shape stores indices into the array for the BLIPs they use. BLIPs are used not only for inserted pictures, but also for the textured and pictures fills of the shape.

BLIP Store Entry Record

`msofbtBSE`

Each BLIP in the BStore is serialized to a File BLIP Store Entry (FBSE) record. The instance field encodes the type of the blip. A fixed size header contains the rest of the common information about the BLIP. If the cbName field in the FBSE is nonzero, a null-terminated Unicode string is written immediately after the FBSE in the file.

```c
// FBSE - File Blip Store Entry
typedef struct _FBSE
{
    BYTE btWin32;    // Required type on Win32
    BYTE btMacOS;    // Required type on Mac
    BYTE rgbUid[16]; // Identifier of blip
    WORD tag;        // currently unused
    ULONG size;      // Blip size in stream
    ULONG cRef;      // Reference count on the blip
    MSOFO foDelay;   // File offset in the delay stream
    BYTE usage;      // How this blip is used (MSOBLIPUSAGE)
    BYTE cbName;     // length of the blip name
    BYTE unused2;    // for the future
    BYTE unused3;    // for the future
} FBSE;
```
typedef enum
{
    msoblipUsageDefault,  // All non-texture fill blips get this.
    msoblipUsageTexture,
    msoblipUsageMax = 255 // Since this is stored in a byte
} MSOBLIPUSAGE;

typedef enum
{
    // GEL provided types...
    msoblipERROR = 0,          // An error occurred during loading
    msoblipUNKNOWN,            // An unknown blip type
    msoblipEMF,                // Windows Enhanced Metafile
    msoblipWMF,                // Windows Metafile
    msoblipPICT,               // Macintosh PICT
    msoblipJPEG,               // JPEG
    msoblipPNG,                // PNG or GIF
    msoblipDIB,                // Windows DIB
    msoblipTIFF = 17,          // TIFF
    msoblipCMYKJPEG = 18,      // JPEG data in YCCK or CMYK color space
    msoblipFirstClient = 32,   // First client defined blip type
    msoblipLastClient = 255    // Last client defined blip type
} MSOBLIPTYPE;

typedef enum
{
    msobiUNKNOWN = 0,
    msobiWMF = 0x216,  // Metafile header then compressed WMF
    msobiEMF = 0x3D4,  // Metafile header then compressed EMF
    msobiPICT = 0x542, // Metafile header then compressed PICT
    msobiPNG = 0x6E0,  // One byte tag then PNG data
    msobiJFIF = 0x46A, // One byte tag then JFIF data
    msobiJPEG = msobiJFIF,
    msobiDIB = 0x7A8,  // One byte tag then DIB data
    msobiCMYKJPEG = 0x6E2, // One byte tag then CMYK/YCCK JPEG data
    msobiTIFF = 0x6e4, // One byte tag then TIFF data
    msobiClient = 0x800, // Clients should set this bit
} MSOBI;             // Blip signature as encoded in the MSOFBH.inst

The btWin32 and btMacOS fields store the MSOBLIPTYPE for the respective operating systems. When the OS blip type doesn't match the blip type of stored, Escher will attempt to convert the blip. For example, a PICT will be stored as a msoblipPICT with a btWin32 field of msoblipWMF and a btMacOS field of msoblipPICT. When the PICT blip is loaded on Windows, the stored field will not match the OS field, so PICTtoWMF filter will be called to create a msoblipWMF BLIP.

A few additional facts are worth noting. Clients can define their own BLIP types. When loading client defined blip types Escher calls the clients to load the blips. Each BSE contains a 16-byte checksum that is used to quickly compare a BLIP with other BLIPs in the store. Any algorithm could be used for this checksum. Escher uses the *RSA Data Security, Inc. MD4 Message-Digest Algorithm* for the checksums of its BLIP types. Finally, the cRef field can be 0, indicating an empty slot in the BStore.

If a delay stream is not used, then the BLIP data follows the BSE header in a separate record. (If a delay stream *is* being used, the BLIP's record header and data are both written there instead.)
Here is the format of the BLIP data. The FBT (MSOFBH::fbt) of the BLIP record is the MSOBLIPTYPE plus msofbtBlipFirst (0xF018). The instance (MSOFBH::inst) contains a signature that varies by blip type (see the Metafile/PICT/Bitmap Blip sections below.) The data that follows the file block header varies by blip type (again, see the Metafile/PICT/Bitmap Blip sections below.)

**Metafile/PICT Blips**

Those blips have one of the following values from the MSOBI enumeration in MSOFBH::inst: msobiEMF, msobiWMF, or msobiPICT. They are normally stored in a compressed format using the LZ compression algorithm in the format used by GNU Zip deflate/inflate with a 32k window. The format is zlib format 1. The only metafile compression version number currently defined identifies this format and is analogous to the PNG compression type value in the PNG file format. The filter values (MSOBLIPFILTER) define pre-filtering of metafile data to give better compression. Currently no pre-filtering is done (it is likely that filtering on a per-record basis will give substantially better compression in the future).

However, if there is an exception due to out-of-memory or out-of-disk space when saving those blips, the compression operation is skipped and the blips are then saved in a non-compressed format in this case the compressed bits are simply the original metafile data. When the blips are loaded back in memory, a check is performed based on a “compression status” flag (MSOBLIPCOMPRESSION) that follows the blip header encoded as follows:

```c
typedef enum
{
    msocompressionDeflate = 0,
    msocompressionNone = 254,    // Used only if compression fails
    msocompressionTest = 255,    // For testing only
} MSOBLIPCOMPRESSION;

typedef enum
{
    msofilterAdaptive = 0,       // PNG type - not used/supported for metafile

1 The formal documentation is as follows (note that this will almost certainly not be of interest, see the comment about code resources below.)

JFIF: JPEG File Interchange Format version 1.02 (September 1992) by Eric Hamilton

Sorry, but for some reason, I can’t add comments. Please check the “gz” at the end of the previous link—It appears to me to not belong, but I can’t be sure.

JPEG: ISO/IEC 10918-1
msofilterNone = 254,
msofilterTest = 255, // For testing only
}
MSOBLIPFILTER;

/* The secondary, or data, UID - should always be set. */
BYTE m_rgbUid[16];

/* The primary UID - this defaults to 0, in which case the primary ID is that of the internal data. NOTE!: The primary UID is only saved to disk if (blip_instance ^ blip_signature == 1). Blip_instance is MSOFBH.finst and blip_signature is one of the values defined in MSOBI */
BYTE m_rgbUidPrimary[16]; // optional based on the above check

/* Metafile Blip overhead = 34 bytes. m_cb gives the number of bytes required to store an uncompressed version of the file, m_cbSave is the compressed size. m_mfBounds gives the boundary of all the drawing calls within the metafile (this may just be the bounding box or it may allow some whitespace, for a WMF this comes from the SetWindowOrg and SetWindowExt records of the metafile). */
int m_cb; // Cache of the metafile size
RECT m_rcBounds; // Boundary of metafile drawing commands
POINT m_ptSize; // Size of metafile in EMUs
int m_cbSave; // Cache of saved size (size of m_pvBits)
BYTE m_fCompression; // MSOBLIPCOMPRESSION
BYTE m_fFilter; // always msofilterNone
void *m_pvBits; // Compressed bits of metafile.

**Bitmap Blips**

Those blips have one of the following values from the MSOBI enumeration in MSOFBH::inst: msobiJPEG, msobiPNG, msobiCMYKJPEG, msobiTIFF, or msobiDIB. They have the same UID header as described in the Metafile Blip case. The data after the header is just a single BYTE "tag" value and is followed by the compressed data of the bitmap in the relevant format (JFIF, TIFF, GIF or PNG, bytes as would be stored in a file). For the msobiDIB format, the data is in the standard DIB format as a BITMAPINFOHEADER, BITMAPCOREHEADER or BITMAPV4HEADER followed by the color map (DIB_RGB_COLORS) and the bits. This data is not compressed (the format is used for very small DIB bitmaps only).

To determine where the bits are located, refer to the following header:

/* The secondary, or data, UID - should always be set. */
BYTE m_rgbUid[16];

/* The primary UID - this defaults to 0, in which case the primary ID is that of the internal data. NOTE!: The primary UID is only saved to disk if (blip_instance ^ blip_signature == 1). Blip_instance is MSOFBH.finst and blip_signature is one of the values defined in MSOBI */
BYTE m_rgbUidPrimary[16]; // optional based on the above check
BYTE m_bTag;
void *m_pvBits; // raster bits of the blip.

**Drawing Container**

`msofbtDgContainer`
The drawing container contains all per-slide/sheet types of information, including the shapes themselves. With a few exceptions, shapes are stored hierarchically according to how they’ve been grouped (through use of the Draw/Group command). For normal shapes, there is a special parent group shape called the patriarch that contains all of the top-level shapes (which in turn may contain other shapes). The patriarch is always the first msofbtSpgrContainer in the drawing container.

A few kinds of shapes are stored separately from the patriarch. The background shape, if there is one, is saved in its own msofbtSpContainer after the patriarch and its children. Additionally, if undo information is being saved and there are deleted shapes that could be brought back via Undo, the deleted shapes are saved. Note that there is no patriarch for the deleted shapes, so the top-level deleted shapes are saved separately into the drawing container. (Deleted groups still contain their children, though.)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Condition</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>msofbtDg</td>
<td>Always.</td>
<td>Basic drawing information.</td>
</tr>
<tr>
<td>msofbtRegroupItems</td>
<td>Shapes have been ungrouped.</td>
<td>Mappings to reconstitute groups.</td>
</tr>
<tr>
<td>msofbtSpgrContainer</td>
<td>Always.</td>
<td>Patriarch shape, with all non-background non-deleted shapes inside it.</td>
</tr>
<tr>
<td>msofbtSpContainer</td>
<td>Application uses a background shape (currently Word and PowerPoint only).</td>
<td>Special shape used as background of the document, e.g. the background texture of a Web page.</td>
</tr>
<tr>
<td>msofbtSpContainer</td>
<td>Undo is being saved, and there are deleted shapes in the drawing.</td>
<td>Shapes that have been deleted but that could be brought back via Undo.</td>
</tr>
<tr>
<td>msofbtSolverContainer</td>
<td>There are rules in the drawing.</td>
<td>Rules governing shapes in the drawing.</td>
</tr>
<tr>
<td>msofbtColorScheme</td>
<td>The application uses a color scheme.</td>
<td>Only present in the clipboard format.</td>
</tr>
</tbody>
</table>

**Drawing Record**

*msofbtDg*

The drawing record is very simple, with just a count and MSOSPID seed. The attentive reader may expect to find the size of the drawing recorded here, but that information is stored elsewhere by the host application.

```c
// FDG - File DG
typedef struct _FDG
{
    ULONG csp;          // The number of shapes in this drawing
    MSOSPID spidCur;    // The last MSOSPID given to an SP in this DG
} FDG;
```

**Regroup**

*msofbtRegroupItems*
Each shape in a drawing has a regroup ID (separate from the shape ID), so that the regroup command can find shapes that were once grouped. In order to handle nested cases (e.g. ungroup, ungroup, ungroup, regroup, regroup, regroup), there is a table logging changes to regroup IDs. Each entry has an old ID and a new ID and records the change of all instances of the old ID to the new ID.

The instance of an msofbtRegroupItems record contains the number of entries, and the record itself is just that many FRITs (File Regroup Items).

type def struct _FRIT // File Regroup item
    {
        FRID fridNew;
        FRID fridOld;
    } FRIT;

Group Container

*msofbtSpgrContainer*

A group is a collection of other shapes. The contained shapes are placed in the coordinate system of the group. The group container contains a variable number of shapes (msofbtSpContainer) and other groups (msofbtSpgrContainer, for nested groups). The group itself is a shape, and always appears as the first msofbtSpContainer in the group container.

Shape Container

*msofbtSpContainer*

A shape is the elemental object that composes a drawing. All graphical figures on a drawing are shapes. Each shape has a list of properties, which is stored in an array. A shape container contains the following records:

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Condition</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>msofbtSpgr</td>
<td>Shape is a group shape.</td>
<td>Group-shape-specific information.</td>
</tr>
<tr>
<td>msofbtSp</td>
<td>Always.</td>
<td>A shape atom record.</td>
</tr>
<tr>
<td>msofbtOPT</td>
<td>Always.</td>
<td>Those properties of a shape that are stored in Block 1.</td>
</tr>
<tr>
<td>msofbtSecondaryOPT</td>
<td>Shape has properties from Block 2.</td>
<td>The Block 2 properties of a shape.</td>
</tr>
<tr>
<td>msofbtTertiaryOPT</td>
<td>Shape has properties from Block 3.</td>
<td>The Block 3 properties of a shape.</td>
</tr>
<tr>
<td>msofbtAnchor or msofbtChildAnchor or msofbtClientAnchor</td>
<td>Always, except for the background shape.</td>
<td>The anchor or location of the shape. If the shape is saved to a clipboard, a msofbtAnchor record is used. If the shape is a child of a group shape, a msofbtChildAnchor is used. Otherwise, for top-level shapes, a host anchor record is present.</td>
</tr>
<tr>
<td>msofbtClientData</td>
<td>Always.</td>
<td>A client data record, the content of which is up to the host.</td>
</tr>
<tr>
<td>msofbtClientTextbox or msofbtTextbox</td>
<td>Shape has attached text.</td>
<td>If the shape has text, a text record is written. For clipboard streams, a msofbtTextbox record is used. Otherwise, a msofbtClientTextbox record is used, the content of which is up to the host.</td>
</tr>
</tbody>
</table>
msofbtOleObject  Shape is an OLE object.  Used only in the clipboard format.
msofbtDeletedPspl Shape is deleted.  Link to previous spot of object.

Group Shape Record

msofbtSpgr
This record is present only in group shapes (not shapes in groups, shapes that are groups). The group shape record defines the coordinate system of the shape, which the anchors of the child shape are expressed in. All other information is stored in the shape records that follow.

typedef struct _FSPGR
{
   RECT rcgBounds;
} FSPGR;

Shape Record

msofbtSp
The instance field of the record header contains the shape type; the record itself contains the shape ID and a group of persistent flags:

typedef struct _FSP
{
   MSOSPID spid;   // The shape id
   ULONG grfPersistent;
} FSP;

The flags for the shape are:

typedef struct
{
   ULONG fGroup : 1;   // This shape is a group shape
   ULONG fChild : 1;   // Not a top-level shape
   ULONG fPatriarch : 1;   // This is the topmost group shape.
      // Exactly one of these per drawing.
   ULONG fDeleted : 1;   // The shape has been deleted
   ULONG fOleShape : 1;   // The shape is an OLE object
   ULONG fHaveMaster : 1;   // Shape has a hspMaster property
   ULONG fFlipH : 1;      // Shape is flipped horizontally
   ULONG fFlipV : 1;      // Shape is flipped vertically
   ULONG fConnector : 1;  // Connector type of shape
   ULONG fHaveAnchor : 1; // Shape has an anchor of some kind
   ULONG fBackground : 1; // Background shape
   ULONG fHaveSpt : 1;    // Shape has a shape type property
   ULONG reserved : 20;   // Not yet used
   }

Property Table Records

msofbtOPT, msofbtSecondaryOPT, msofbtTertiaryOPT
A shape’s properties are stored in a sorted array of property id-value pairs. Only the properties that differ from the per-shape-type defaults or the per-property defaults are saved. (Note that the per-property defaults are unrelated to the default property table stored in the drawing group container).

The format of a property table record is in the Shape Properties section of this document.

Anchor Record

msofbtAnchor

An anchor record is used for top-level shapes when the shape streamed to the clipboard. The content of the record is simply a RECT with a coordinate system of 100,000 units per inch and origin in the top-left of the drawing.

Child Anchor Record

msofbtChildAnchor

A child anchor record is used for all shapes that belong to a group. The content of the record is simply a RECT in the coordinate system of the parent group shape.

Textbox Record

msofbtTextbox

A textbox record is used when a shape with attached text is written to a clipboard stream. It just contains an RTF string.

OLE Object Record

msofbtOleObject

An OLE object record is present when a shape that is an OLE object is saved to the clipboard. It contains the OLE object’s storage, serialized using OleConvertIStorageToOLESTREAM.

Deleted PSPL Record

msofbtDeletedPspl

Top-level deleted shapes save a pointer back into their former position in the shape tree, so that if they are undeleted via undo they can be easily put back into the main shape tree.

The record consists of a single FPSPL:

```c
// FPSPL - File PSPL
typedef struct _FPSPL
{
    union
    {
        ULONG lAll;
        struct
        {
            ULONG spid : 30; // The SPID of the shape PSPL points at.
            ULONG fFirst : 1; // Is this a pointer to the m_splFirst?
            ULONG fLast : 1; // Is this a pointer to the m_splLast?
        };
    };
};
```
Solver Container

*msofbtSolverContainer*

Rules give special behaviors to shapes. Rules can govern a single shape, like in the case of a callout shape, or multiple shapes, as in the case of connectors. Each drawing can have a list of rules associated with it.

Connector Rule Record

*msofbtConnectorRule*

 Governs a connector shape.

typedef struct _FConnectorRule
{
    ULONG ruid; // rule ID
    MSOSPID spidA; // SPID of shape A
    MSOSPID spidB; // SPID of shape B
    MSOSPID spidC; // SPID of connector shape
    ULONG cptiA; // Connection site Index of shape A
    ULONG cptiB; // Connection site Index of shape B
} FConnectorRule;

Align Rule Record

*msofbtAlignRule*

Aligns shapes. The FAlignRule record is followed by the SPIDs of the proxy shapes.

// FAlignRule
typedef struct _FAlignRule
{
    ULONG ruid; // rule ID
    ULONG align; // alignment  – see below
    ULONG cProxies; // number of shapes governed by rule
} FAlignRule;

// ALIGN == Shape alignment (Horz and vert can be or'ed together)
#define alignHorz 0x000F // mask for horizontal component
#define alignLeft 0x0001 // left edges
#define alignCenter 0x0002 // horizontal center
#define alignRight 0x0003 // right edges
#define alignVert 0x00F0 // mask for vertical component
#define alignTop 0x0010 // top edges
#define alignMiddle 0x0020 // vertical center
#define alignBottom 0x0030 // bottom edges
#define alignRelative 0x0100 // Relative to the page

Arc Rule Record

*msofbtArcRule*

One Arc rule per elliptical arc shape.
// FARCRU -- Arc Rule
typedef struct _FARCRU
{
  ULONG  ruid; // rule ID
  MSOSPID spid; // spid of arc shape
} FARCRU;

Callout Rule Record

**msofbtCalloutRule**

One callout rule per callout shape.

// FCORU -- Callout Rule
typedef struct _FCORU
{
  ULONG  ruid; // rule ID
  MSOSPID spid; // spid of callout shape
} FCORU;

Color Scheme

**msofbtColorScheme**

Hosts may define their own color scheme and store colors in shape properties that are an index to that scheme plus an indicating flag (see Appendix B). Since hosts' color schemes are independent of each other, the color scheme is saved when rendering to the clipboard. If the clipboard data is pasted back into the same application, the color scheme block is ignored; otherwise, it is used to translate scheme colors in properties into RGB values.

The data in the block is an array of RGB values, saved as LONGs in order of color scheme index.

Selections

**msofbtSelection**

Selections of shapes are saved as top-level file blocks; they are never placed in a container. (Note: As of Office 97, only Excel saves shape selections; Word and PowerPoint do not.) The selection record consists of an FDGSL followed by the SPIDs of the shapes in the selection.

// FDGSL - File Drawing Selection
typedef struct _FDGSL
{
  ULONG  cpsp; // number of shapes in the selection
  ULONG  dgslk; // kind of selection (an MSODGSLK)
  MSOSPID spidFocus; // SPID of the focus shape
} FDGSL;

// DGSLK = DrawinG SeLection Kind.
typedef enum
{
  msodgslkNormal, // Normal Selection Mode.
  msodgslkRotate, // Rotate selection mode
  msodgslkReshape, // Reshape Selection Mode.
  msodgslkUnused,
  msodgslkWrapPolygon, // Display and edit of wrap polygons.
}
Shape Properties

*msofbtOPT, msofbtSecondaryOPT, msofbtTertiaryOPT*

The first part of an OPT record is an array of FOPTEs, consisting of ID-value pairs tightly packed:

```c
typedef struct _FOPTE
{
    struct
    {
        USHORT pid : 14; // Property ID
        USHORT fBid : 1; // value is a blip ID - only valid if fComplex is FALSE
        USHORT fComplex : 1; // complex property, value is length
    }
    ULONG op; // Value
} FOPTE;
```

The FOPTE array is sorted by property ID.

Some property values, such as Unicode strings, don’t fit in 32 bits. For these properties, the fComplex bit is set in the FOPTE, and the length of the data is saved in the value slot. The data of the complex properties follows the FOPTE array in the file record (sorted by property ID).

BLIPs are usually saved in the BLIP Store, so, in most cases, BLIP properties just store a BLIP ID (basically an index into an array in the BLIP Store). This is signaled by the fBid flag; note however that this flag is only valid if fComplex is FALSE.

Boolean properties are grouped in bitfields by property set; note that the Boolean properties in each property set below are contiguous. They are saved under the property ID of the *last* Boolean property in the set, and are placed in the value field in reverse order starting with the last property in the low bit.

Notes on types and units:

MSOHSP properties are basically just pointers to shapes, and they are therefore saved as SPIDs.

WCHAR* properties are Unicode strings; char* properties are ASCII strings.

IMsoArray properties are arrays. They are always complex when non-NULL. The complex-data part is saved as three shorts (16 bits each) followed by the array data. The first short is the number of elements in the array; the second short is the number of elements allocated for the array in memory (always greater than or equal to the first short); and the third short is the size of each array element.

Absolute distances are specified in English Metric Units (EMUs), occasionally referred to as A units; there are 360000 EMUs per centimeter, 914400 EMUs per inch, 12700 EMUs per point.

A coordinate space relative to the size of the shape is specified with the geoLeft, geoTop, geoRight, and geoBottom properties; coordinates in this space are said to be in G units.

Many quantities are specified as fixed-point 16.16 numbers; that is, the quantity fits in a LONG, where the high word specifies the integer part and the low word specifies the fractional part. In this system, 1<<16 signifies 1, 1<<17 signifies 2, and 1<<15 signifies ½.

The property listings below contain only those properties which are saved in OPT records in the file. Properties which are never saved, or which appear elsewhere in the file format, have been omitted.
The "Ver" column in the property listings specifies the version of Office in which the property was added. The property is known to that version and all subsequent versions, but not the ones preceding that version:

<table>
<thead>
<tr>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>Office 97 for Windows</td>
</tr>
<tr>
<td>98</td>
<td>Office 98 for Macintosh</td>
</tr>
<tr>
<td>2000</td>
<td>Office 2000 for Windows</td>
</tr>
<tr>
<td>XP</td>
<td>Office XP for Windows</td>
</tr>
<tr>
<td>2003</td>
<td>Office System 2003 for Windows</td>
</tr>
<tr>
<td>2007</td>
<td>2007 Microsoft Office System for Windows</td>
</tr>
</tbody>
</table>

"Block 1, Block 2, Block 3"

The property table for a single shape or default property table may be split into as many as three records, with each individual record in the OPT format described above. Block 1 properties go into a record of type msofbtOPT, Block 2 properties into msofbtSecondaryOPT, and Block 3 properties into msofbtTertiaryOPT.

Using the "Ver" column, the implementer must infer which "block" a property should be written in as follows: if "Ver" is 97 or 98, the property must appear in block 1 unless its property ID is 274 (movie,) in which case it must appear in block 2. If "Ver" is anything else, the property must appear in block 3.

When emitting the Office Drawing file format, the implementer must partition property table entries among the three record types as defined above. However, at read time an implementation must be prepared to handle any property in a block of any of the three types.

**Transform**

Position, size, rotation, and flipping of the shape.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>left</td>
<td>0</td>
<td>LONG</td>
<td>0</td>
<td>97 Bounds of the unrotated shape expressed as top left and bottom right in drawing units.</td>
</tr>
<tr>
<td>top</td>
<td>1</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
</tr>
<tr>
<td>right</td>
<td>2</td>
<td>LONG</td>
<td>1</td>
<td>97</td>
</tr>
<tr>
<td>bottom</td>
<td>3</td>
<td>LONG</td>
<td>1</td>
<td>97</td>
</tr>
<tr>
<td>rotation</td>
<td>4</td>
<td>LONG</td>
<td>0</td>
<td>97 Rotation is about the top left. Fixed point: 16.16 degrees</td>
</tr>
<tr>
<td>gvPage</td>
<td>5</td>
<td>MSOGV</td>
<td>0</td>
<td>97</td>
</tr>
<tr>
<td>fChangePage</td>
<td>61</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
</tr>
<tr>
<td>fFlipV</td>
<td>62</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97 Flip vertically</td>
</tr>
<tr>
<td>fFlipH</td>
<td>63</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97 Flip horizontally</td>
</tr>
</tbody>
</table>

**Protection**

Changes the behavior of a shape by restricting direct manipulation.
### Property

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fLockAgainstUngrouping</td>
<td>118</td>
<td>BOOL</td>
<td>FALSE</td>
<td>XP</td>
<td>Do not ungroup this shape</td>
</tr>
<tr>
<td>fLockRotation</td>
<td>119</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>No rotation</td>
</tr>
<tr>
<td>fLockAspectRatio</td>
<td>120</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Don’t allow changes in aspect ratio</td>
</tr>
<tr>
<td>fLockPosition</td>
<td>121</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Don’t allow the shape to be moved</td>
</tr>
<tr>
<td>fLockAgainstSelect</td>
<td>122</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Shape may not be selected</td>
</tr>
<tr>
<td>fLockCropping</td>
<td>123</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>No cropping this shape</td>
</tr>
<tr>
<td>fLockVertices</td>
<td>124</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Edit Points not allowed</td>
</tr>
<tr>
<td>fLockText</td>
<td>125</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Do not edit text</td>
</tr>
<tr>
<td>fLockAdjustHandles</td>
<td>126</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Do not adjust</td>
</tr>
<tr>
<td>fLockAgainstGrouping</td>
<td>127</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Do not group this shape</td>
</tr>
</tbody>
</table>

### Text

How text fits in a shape. Text is host-dependent, so some hosts may ignore some of these properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lTxid</td>
<td>128</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>id for the text, value determined by the host</td>
</tr>
<tr>
<td>dxTextLeft</td>
<td>129</td>
<td>LONG</td>
<td>1/10 inch</td>
<td>97</td>
<td>margins relative to shape's inscribed text</td>
</tr>
<tr>
<td>dyTextTop</td>
<td>130</td>
<td>LONG</td>
<td>1/20 inch</td>
<td>97</td>
<td>margins relative to shape's inscribed text</td>
</tr>
<tr>
<td>dxTextRight</td>
<td>131</td>
<td>LONG</td>
<td>1/10 inch</td>
<td>97</td>
<td>margins relative to shape's inscribed text</td>
</tr>
<tr>
<td>dyTextBottom</td>
<td>132</td>
<td>LONG</td>
<td>1/20 inch</td>
<td>97</td>
<td>margins relative to shape's inscribed text</td>
</tr>
<tr>
<td>WrapText</td>
<td>133</td>
<td>MSOWRAPMODE</td>
<td>FALSE</td>
<td>97</td>
<td>Wrap text at shape margins</td>
</tr>
<tr>
<td>scaleText</td>
<td>134</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Text zoom/scale (used if fFitTextToShape)</td>
</tr>
<tr>
<td>anchorText</td>
<td>135</td>
<td>MSOANCHOR</td>
<td>Top</td>
<td>97</td>
<td>How to anchor the text</td>
</tr>
<tr>
<td>txflTextFlow</td>
<td>136</td>
<td>MSOTXFL</td>
<td>HorzN</td>
<td>97</td>
<td>Text flow</td>
</tr>
<tr>
<td>cdirFont</td>
<td>137</td>
<td>MSOCDIR</td>
<td>msocdir0</td>
<td>97</td>
<td>Font rotation</td>
</tr>
<tr>
<td>hspNext</td>
<td>138</td>
<td>MSOHSP</td>
<td>NULL</td>
<td>97</td>
<td>ID of the next shape (used by Word for linked</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>textboxes)</td>
</tr>
<tr>
<td>txdir</td>
<td>139</td>
<td>MSOTXDIR</td>
<td>LTR</td>
<td>97</td>
<td>Bi-Di Text direction</td>
</tr>
<tr>
<td>Property</td>
<td>PID</td>
<td>Type</td>
<td>Default</td>
<td>Ver</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------</td>
<td>---------------</td>
<td>----------</td>
<td>-----</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>gtextUNICODE</td>
<td>192</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>97</td>
<td>UNICODE text string</td>
</tr>
<tr>
<td>gtextRTF</td>
<td>193</td>
<td>char*</td>
<td>NULL</td>
<td>97</td>
<td>RTF text string</td>
</tr>
<tr>
<td>gtextAlign</td>
<td>194</td>
<td>MSOGEOTEXTALIGN</td>
<td>Center</td>
<td>97</td>
<td>alignment on curve</td>
</tr>
<tr>
<td>gtextSize</td>
<td>195</td>
<td>LONG</td>
<td>36&lt;&lt;16</td>
<td>97</td>
<td>default point size</td>
</tr>
<tr>
<td>gtextSpacing</td>
<td>196</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>97</td>
<td>fixed point 16.16</td>
</tr>
<tr>
<td>gtextFont</td>
<td>197</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>97</td>
<td>font family name</td>
</tr>
<tr>
<td>gtextCSSFont</td>
<td>198</td>
<td>WCHAR*</td>
<td>&quot;&quot;</td>
<td>2000</td>
<td>To preserve CSS font selectors</td>
</tr>
<tr>
<td>gtextFReverseRows</td>
<td>240</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>By default multiple rows of text are laid out with the first at the top for horizontal text and with the first at the left for vertical text, this flag reverses that behavior (bottom to top or right to left)</td>
</tr>
<tr>
<td>fGtext</td>
<td>241</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Has text effect</td>
</tr>
<tr>
<td>gtextFVertical</td>
<td>242</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Rotate characters</td>
</tr>
<tr>
<td>gtextFKern</td>
<td>243</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Kern characters</td>
</tr>
</tbody>
</table>

GeoText

Effect text of the shape - this is what the WordArt tools use, and is separate from the attached text present in ordinary textboxes. Theoretically, a shape could have both (a WordArt with attached text), but this is not currently allowed by the UI. Note that font information is provided here. The default text size is in points, the text effect geometry interfaces require the device size of a point to interpret this. The default point size is a 16.16 fixed-point number. A text effect is present if the fGText boolean is set and either the gtextUNICODE (UNICODE) or gtextRTF (RTF) is present, the UNICODE string takes precedence, however it cannot include any additional font information (unlike the RTF).
<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gtextFTight</td>
<td>244</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Tightening or tracking</td>
</tr>
<tr>
<td>gtextFStretch</td>
<td>245</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Stretch to fit shape</td>
</tr>
<tr>
<td>gtextFShrinkFit</td>
<td>246</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Char bounding box</td>
</tr>
<tr>
<td>gtextFBestFit</td>
<td>247</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Scale text-on-path</td>
</tr>
<tr>
<td>gtextFNormalize</td>
<td>248</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Stretch char height</td>
</tr>
<tr>
<td>gtextFDxMeasure</td>
<td>249</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Do not measure along path</td>
</tr>
<tr>
<td>gtextFBold</td>
<td>250</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Bold font</td>
</tr>
<tr>
<td>gtextFItalic</td>
<td>251</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Italic font</td>
</tr>
<tr>
<td>gtextFUnderline</td>
<td>252</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Underline font</td>
</tr>
<tr>
<td>gtextFSHadow</td>
<td>253</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Shadow font</td>
</tr>
<tr>
<td>gtextFSmallCaps</td>
<td>254</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Small caps font</td>
</tr>
<tr>
<td>gtextFStrikethrough</td>
<td>255</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Strike through font</td>
</tr>
</tbody>
</table>

**Blip**

How a BLIP fits into a shape. This includes cropping information as well as picture display modifications such as brightness and contrast.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cropFromTop</td>
<td>256</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>16.16 fraction times total image width or height, as appropriate.</td>
</tr>
<tr>
<td>cropFromBottom</td>
<td>257</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>16.16 fraction times total image width or height, as appropriate.</td>
</tr>
<tr>
<td>cropFromLeft</td>
<td>258</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>16.16 fraction times total image width or height, as appropriate.</td>
</tr>
<tr>
<td>cropFromRight</td>
<td>259</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>16.16 fraction times total image width or height, as appropriate.</td>
</tr>
<tr>
<td>Pib</td>
<td>260</td>
<td>IMsoBlip*</td>
<td>NULL</td>
<td>97</td>
<td>Blip to display</td>
</tr>
<tr>
<td>pibName</td>
<td>261</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>97</td>
<td>Blip file name</td>
</tr>
<tr>
<td>pibFlags</td>
<td>262</td>
<td>MSOBLIPFLAGS</td>
<td>Comment</td>
<td>97</td>
<td>Blip flags</td>
</tr>
<tr>
<td>pictureTransparent</td>
<td>263</td>
<td>Extended Color</td>
<td>~0</td>
<td>97</td>
<td>transparent color (none if ~0UL)</td>
</tr>
<tr>
<td>pictureContrast</td>
<td>264</td>
<td>LONG</td>
<td>~&lt;16</td>
<td>97</td>
<td>contrast setting</td>
</tr>
<tr>
<td>pictureBrightness</td>
<td>265</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>brightness setting</td>
</tr>
<tr>
<td>pictureGamma</td>
<td>266</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>16.16 gamma</td>
</tr>
<tr>
<td>pictureId</td>
<td>267</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Host-defined ID for OLE</td>
</tr>
</tbody>
</table>
The geometry of the shape. Typically, these properties reside in a shape type definition, and so are not written to the file. However, freeform shapes drawing using the polygon tools set the pVertices and pSegmentInfo properties to define their geometries.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>geoLeft</td>
<td>320</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Defines the G (geometry) coordinate space.</td>
</tr>
<tr>
<td>geoTop</td>
<td>321</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Defines the G (geometry) coordinate space.</td>
</tr>
<tr>
<td>geoRight</td>
<td>322</td>
<td>LONG</td>
<td>21600</td>
<td>97</td>
<td>Defines the G (geometry) coordinate space.</td>
</tr>
<tr>
<td>geoBottom</td>
<td>323</td>
<td>LONG</td>
<td>21600</td>
<td>97</td>
<td>Defines the G (geometry) coordinate space.</td>
</tr>
<tr>
<td>shapePath</td>
<td>324</td>
<td>MSOSHA</td>
<td>msoSHAPE</td>
<td>97</td>
<td>Defines the G (geometry) coordinate space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PEPATH</td>
<td>peLines Closed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
pVertices 325 IMsoArray NULL 97 An array of points, in G units.

pSegmentInfo 326 IMsoArray NULL 97

adjustValue 327 LONG 0 97 Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.

adjust2Value 328 LONG 0 97 Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.

adjust3Value 329 LONG 0 97 Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.

adjust4Value 330 LONG 0 97 Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.

adjust5Value 331 LONG 0 97 Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.

adjust6Value 332 LONG 0 97 Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.

adjust7Value 333 LONG 0 97 Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.

adjust8Value 334 LONG 0 97 Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.

adjust9Value 335 LONG 0 97 Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.
Adjustment values corresponding to the positions of the adjust handles of the shape. The number of values used and their allowable ranges vary from shape type to shape type.

Array of connection sites (points where connectors can attach) in G units. Array entries are of type POINT.

Array of angles specifying angle at which connectors should connect to the corresponding connection sites. Angles are fixed point, 16.16 degrees. If this property is omitted, angles are determined automatically based on the geometric center of the shape. Array entries are of type LONG.

The point along the x dimension of the shape where it will limo stretch. Specified in G units.

The point along the y dimension of the shape where it will limo stretch. Specified in G units.

Array of adjust handles for the shape. Array entries are of type ADJH.

Array of guide formula for the shape which specify how the geometry of the shape changes as the adjust handles are dragged. Array entries are of type SG.

Array of inscribed rectangles. Array entries are of type RECT.

Type of connection sites

Array of fragment ids

Column style may be set

Shadow may be set

3D may be set

Line style may be set

Text effect (WordArt) supported

If true, a concentric gradient fill will be drawn based on the geometry of the shape. If false, a concentric gradient fill will be
fFillOK  383  BOOL  TRUE  97  OK to fill the shape through the UI or VBA?

For simple closed polygons, the pSegmentInfo property can be omitted. The data in the pVerticies array is interpreted as a list of vertices for the polygon. For a more complex path, the pSegmentInfo property specifies the types of segments and number of segments in the path and determines how you should interpret the list of values in the pVerticies array.

For example, let's take a look at the path data for the Heart shape. The pSegmentInfo array looks like:

```plaintext
msopathMoveTo, 0
msopathCurveTo, 5
msopathLineTo, 2
msopathCurveTo, 5
msopathClose, 1
msopathEnd, 0
```

and the pVerticies array looks like:

```plaintext
10860, 2187
10451, 1746
9529, 1018
9015, 730
7865, 152
6685, 0
5415, 0
4175, 152
2995, 575
1967, 1305
1150, 2187
575, 3222
242, 4220
0, 5410
242, 6560
575, 7597
10860, 21600
20995, 7597
21480, 6560
21600, 5410
21480, 4220
21115, 3222
20420, 2187
19632, 1305
18575, 575
17425, 152
16275, 0
15005, 0
13735, 152
12705, 730
12176, 1018
11254, 1746
```
10860, 2187

- The MoveTo,0 says to take the 1st vertex in pVertices, (10860,2187), and place the pen at that location.
- The CurveTo,5 says to draw 5 cubic bezier segments. The current pen location is used as the first vertex of the first cubic bezier segment, and the next 3 vertices from pVertices, (10451,1746), (9529,1018), and (9015,730), are used as the control points and endpoint of the first cubic bezier segments. This is repeated 4 more times using a total of 15 vertices from pVertices for the 5 cubic bezier segments.
- The LineTo,2 says to draw 2 straight line segments. One line segment is drawn from the current pen location to the next vertex in pVertices, (10860,21600), and another line drawn from that point to the next vertex in pVertices, (20995,7597).
- The CurveTo,5 says to draw 5 more cubic bezier segments using another 15 vertices in pVertices.
- The Close,1 says the draw 1 line segment from the current pen location back to first point in the path and make it a closed path. No vertices from pVertices are taken for this operation.
- The End,0 says to end the path. No vertices from pVertices are taken for this operation.

Each SegmentType,SegmentCount pair is stored in an unsigned short value. The segment type is stored in the upper 3 bits and segment count is stored in the lower 13 bits.

Segment type can be any of the follow enum values:

```c
typedef enum
{
    msopathLineTo,  // Draw a straight line (one point)
    msopathCurveTo, // Draw a cubic Bezier curve (three points)
    msopathMoveTo,  // Move to a new point (one point)
    msopathClose,   // Close a sub-path (no points)
    msopathEnd,     // End a path (no points)
    msopathEscape,  // Escape code
    msopathClientEscape, // Escape code interpreted by the client
    msopathInvalid  // Invalid - should never be found
} MSOPATHTYPE;
```

If the segment type is msopathEscape, the lower 13 bits are divided in a 5 bit escape code and 8 bit vertex count (not segment count!).

The freeform objects produced in Office will contain some msopathEscape. These store editing information (like whether or not to allow the control points to be adjusted on a bezier segment). But these are not needed to understand how the freeforms are rendered.

All CurveTo segments in a path are cubic beziers. The mathematical definition for a cubic bezier can be found in most computer graphics textbooks. In Office, some of the built-in AutoShapes have some cubic bezier curve segments in them. Shapes drawn with the "Curve" tool have only cubic bezier curve segments in them. Shapes drawn with the "Freeform" tool have cubic bezier curve segments in the smooth sections and straight-line segments in the straight sections.

**Fill Style**

Two main colors are defined - a foreground color and a background color. Different fillTypes use these values differently. In addition to the foreground and background any number of shade colors can be
defined. Each shade color is associated with a "position" which says how far into the shade the color appears – colors must be given in position order.

For a solid fill the foreground color is used and the background (and everything else except transparency) is ignored. For pattern and texture fills the fillBlip identifies a BLIP, which will be used for the fill.

fillWidth and fillHeight define the desired pattern/texture size in EMUs. The pattern/tile will be expanded to this size. If the pattern is a bitmap the actual size will be rounded to a close integer multiple of the original (pixel) size of the bitmap. If the size is 0 then the pattern/tile will not be expanded or contracted at all in pixel terms, so the fill will be device dependent - this may result in non-proportional scaling between devices (on devices with non-square pixels).

For a pattern the foreground and background colors define the colors to use when filling with a pattern, for a texture the colors are in the bitmap (this is the only difference).

For both pattern and texture fills the fill is registered with (0,0) on the view in which the effect appears unless fillShape is set to TRUE, in which case the pattern/texture is registered relative to the shape (so it moves with the shape).

For a picture fill the fillBlip is centered in the shape - not tiled.

For a shaded fill the colors define shade bands to use across the shade, shading between each pair of colors. The positions are the positions of the shade at which the given color appears - the shade ends with the first index that is >= 1 (in 16.16 notation), the indices must be in ascending order or the result is undefined. The interpretation of the shade start and shade end points varies according to the exact shade type.

The fillBackground fill indicates a fill inherited from a background object.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillType</td>
<td>384</td>
<td>MSOFILLTYPE</td>
<td>Solid</td>
<td>97</td>
<td>Type of fill</td>
</tr>
<tr>
<td>fillColor</td>
<td>385</td>
<td>Extended Color</td>
<td>white</td>
<td>97</td>
<td>Foreground color</td>
</tr>
<tr>
<td>fillOpacity</td>
<td>386</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>97</td>
<td>Fixed 16.16</td>
</tr>
<tr>
<td>fillBackColor</td>
<td>387</td>
<td>Extended Color</td>
<td>white</td>
<td>97</td>
<td>Background color</td>
</tr>
<tr>
<td>fillBackOpacity</td>
<td>388</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>97</td>
<td>Shades only</td>
</tr>
<tr>
<td>fillCrMod</td>
<td>389</td>
<td>MSOCLR</td>
<td>undefined</td>
<td>97</td>
<td>Modification for BW views</td>
</tr>
<tr>
<td>fillBlip</td>
<td>390</td>
<td>IMsoBlip*</td>
<td>NULL</td>
<td>97</td>
<td>Pattern/texture</td>
</tr>
<tr>
<td>fillBlipName</td>
<td>391</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>97</td>
<td>Blip file name</td>
</tr>
<tr>
<td>fillBlipFlags</td>
<td>392</td>
<td>MSOBLIPFLAGS</td>
<td>Comment</td>
<td>97</td>
<td>Blip flags</td>
</tr>
<tr>
<td>fillWidth</td>
<td>393</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>How big (A units) to make a metafile texture.</td>
</tr>
<tr>
<td>fillHeight</td>
<td>394</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>How big (A units) to make a metafile texture.</td>
</tr>
<tr>
<td>fillAngle</td>
<td>395</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Fade angle - degrees in 16.16</td>
</tr>
<tr>
<td>fillFocus</td>
<td>396</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Linear shaded fill focus percent</td>
</tr>
<tr>
<td>fillToLeft</td>
<td>397</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Fraction 16.16</td>
</tr>
<tr>
<td>fillToTop</td>
<td>398</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Fraction 16.16</td>
</tr>
</tbody>
</table>
fillToRight  399  LONG  0  97  Fraction 16.16
fillToBottom  400  LONG  0  97  Fraction 16.16
fillRectLeft  401  LONG  0  97  For shaded fills, use the specified rectangle instead of the shape’s bounding rect to define how large the fade is going to be.
fillRectTop  402  LONG  0  97  For shaded fills, use the specified rectangle instead of the shape’s bounding rect to define how large the fade is going to be.
fillRectRight  403  LONG  0  97  For shaded fills, use the specified rectangle instead of the shape’s bounding rect to define how large the fade is going to be.
fillRectBottom  404  LONG  0  97  For shaded fills, use the specified rectangle instead of the shape’s bounding rect to define how large the fade is going to be.
fillDztype  405  MSODZTYPE  Default  97
fillShadePreset  406  LONG  0  97  Special shades
fillShadeColors  407  IMsoArray  NULL  97  a preset array of colors
fillOriginX  408  LONG  0  97
fillOriginY  409  LONG  0  97
fillShapeOriginX  410  LONG  0  97
fillShapeOriginY  411  LONG  0  97
fillShadeType  412  MSOSHADETYPE  Default  97  Type of shading, if a shaded (gradient) fill.
RecolorFillAsPicture  441  BOOL  FALSE  XP  Recolor the picture fill according to the recoloring properties from the Picture property set
fUseShapeAnchor  442  BOOL  TRUE  XP  Fit the fill to the shape anchor, not the bounds
fFilled  443  BOOL  TRUE  97  Is shape filled?
fHitTestFill  444  BOOL  TRUE  97  Should we hit test fill?
fillShape  445  BOOL  TRUE  97  Register pattern on shape
fillUseRect  446  BOOL  FALSE  97  Use the large rect?
noFillHitTest  447  BOOL  FALSE  97  Hit test a shape as though filled

Line Style
Lines are centered about the infinitely thin proto-line along which they are drawn. Complex dash effects are supported only for simple lines (e.g. changing the end cap) - defaults should be used for
other line styles. The line width is in EMUs; a line width of zero should not be used - there is no logical
interpretation on a high-resolution printer.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lineColor</td>
<td>448</td>
<td>Extended Color</td>
<td>black</td>
<td>97</td>
<td>Color of line</td>
</tr>
<tr>
<td>lineOpacity</td>
<td>449</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>97</td>
<td>Not implemented</td>
</tr>
<tr>
<td>lineBackColor</td>
<td>450</td>
<td>Extended Color</td>
<td>white</td>
<td>97</td>
<td>Background color</td>
</tr>
<tr>
<td>lineCrMod</td>
<td>451</td>
<td>MSOCLR</td>
<td>undefined</td>
<td>97</td>
<td>Modification for BW views</td>
</tr>
<tr>
<td>lineType</td>
<td>452</td>
<td>MSOLINETYPE</td>
<td>Solid</td>
<td>97</td>
<td>Type of line</td>
</tr>
<tr>
<td>lineFillBlip</td>
<td>453</td>
<td>IMsoBlip*</td>
<td>NULL</td>
<td>97</td>
<td>Pattern/texture</td>
</tr>
<tr>
<td>lineFillBlipName</td>
<td>454</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>97</td>
<td>Blip file name</td>
</tr>
<tr>
<td>lineFillBlipFlags</td>
<td>455</td>
<td>MSOBLIPFLAGS</td>
<td>Comment</td>
<td>97</td>
<td>Blip flags</td>
</tr>
<tr>
<td>lineFillWidth</td>
<td>456</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>How big (A units) to make a metafile texture.</td>
</tr>
<tr>
<td>lineFillHeight</td>
<td>457</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>How big (A units) to make a metafile texture.</td>
</tr>
<tr>
<td>lineFillDztype</td>
<td>458</td>
<td>MSODZTYPE</td>
<td>Default</td>
<td>97</td>
<td>How to interpret fillWidth/Height numbers.</td>
</tr>
<tr>
<td>lineWidth</td>
<td>459</td>
<td>LONG</td>
<td>9525</td>
<td>97</td>
<td>A units; 1pt == 12700 EMUs</td>
</tr>
<tr>
<td>lineMiterLimit</td>
<td>460</td>
<td>LONG</td>
<td>8&lt;&lt;16</td>
<td>97</td>
<td>ratio (16.16) of width</td>
</tr>
<tr>
<td>lineStyle</td>
<td>461</td>
<td>MSOLINESTYLE</td>
<td>Simple</td>
<td>97</td>
<td>Draw parallel lines?</td>
</tr>
<tr>
<td>lineDashing</td>
<td>462</td>
<td>MSOLINEDASHING</td>
<td>Solid</td>
<td>97</td>
<td>Can be overridden by:</td>
</tr>
<tr>
<td>lineDashStyle</td>
<td>463</td>
<td>IMsoArray</td>
<td>NULL</td>
<td>97</td>
<td>As Win32 ExtCreatePen</td>
</tr>
<tr>
<td>lineStartArrowhead</td>
<td>464</td>
<td>MSOLINEEND</td>
<td>NoEnd</td>
<td>97</td>
<td>Arrow at start</td>
</tr>
<tr>
<td>lineEndArrowhead</td>
<td>465</td>
<td>MSOLINEEND</td>
<td>NoEnd</td>
<td>97</td>
<td>Arrow at end</td>
</tr>
<tr>
<td>lineStartArrowWidth</td>
<td>466</td>
<td>MSOLINEENDENDWI</td>
<td>MediumW</td>
<td>97</td>
<td>Arrow at start</td>
</tr>
<tr>
<td>lineStartArrowLength</td>
<td>467</td>
<td>MSOLINEENDENL</td>
<td>MediumLe</td>
<td>97</td>
<td>Arrow at end</td>
</tr>
<tr>
<td>lineEndArrowWidth</td>
<td>468</td>
<td>MSOLINEENDENDWI</td>
<td>MediumW</td>
<td>97</td>
<td>Arrow at start</td>
</tr>
<tr>
<td>lineEndArrowLength</td>
<td>469</td>
<td>MSOLINEENDENL</td>
<td>MediumLe</td>
<td>97</td>
<td>Arrow at end</td>
</tr>
<tr>
<td>lineJoinStyle</td>
<td>470</td>
<td>MSOLINEJOIN</td>
<td>JoinRound</td>
<td>97</td>
<td>How to join lines</td>
</tr>
<tr>
<td>lineEndCapStyle</td>
<td>471</td>
<td>MSOLINECAP</td>
<td>EndCapFla</td>
<td>97</td>
<td>How to end lines</td>
</tr>
<tr>
<td>fInsetPen</td>
<td>505</td>
<td>BOOL</td>
<td>FALSE</td>
<td>XP</td>
<td>Draw line inside the shape</td>
</tr>
</tbody>
</table>
fInsetPenOK | 506 | BOOL | TRUE | XP | Allow inset pen if prop. is set
fArrowheadsOK | 507 | BOOL | FALSE | 97 | Allow arrowheads if prop. is set
fLine | 508 | BOOL | TRUE | 97 | Any line?
fHitTestLine | 509 | BOOL | TRUE | 97 | Should we hit test lines?
lineFillShape | 510 | BOOL | TRUE | 97 | Register pattern on shape
fNoLineDrawDash | 511 | BOOL | FALSE | 97 | Draw a dashed line if no line

Shadow Style

The interpretation of the transform properties depends on the type of shadow:

**msoshadowOffset, msoshadowDouble:**

Only the offset is used. It is interpreted as an absolute offset expressed in EMUs. The default corresponds to 1/36” in both X and Y (2 or 3 pixels on screen depending on monitor resolution). The offset is relative to the drawing axes (as msoshadowDrawing below, not msoshadowRich) so a shadow offset to the bottom right of the drawing is still offset (by the same amount) to the bottom right if the shape is rotated. The “double” case causes two shadows to be drawn, the first (lower) at the second offset and in the shadowHighlightColor. If the second offset is 0,0, it defaults to being the inverse of the first.

**msoshadowRich:**

The offsets and transformation properties are in absolute units measured relative to the shape on the drawing - the shadow moves with the shape, but anisotropic scaling of the shape changes the proportions of the shadow, not its angles. Compare with the following where such scaling scales the shadow in proportion too, thus changes the angle between (e.g.) a vertical line in the shape and it’s shadow.

**msoshadowShape:**

The offsets and transformation properties are relative to the shape; 1.0 corresponds to the shape width/height as appropriate. The shadow is cast relative to the shape then scaled with the shape, so it moves with the shape. The units are simple numbers (ratios of the G unit space effectively). This transformation type is unnatural in real world terms, but behaves nicely in geometric terms. The offset elements of the property set are treated as fixed-point 16.16 values.

**msoshadowDrawing:**

A rich shadow cast onto a plane in drawing space. The transform is applied to the drawing coordinates of the shape and is thus expressed in EMUs. This shadow type enables creation of shadows from multiple objects; however the shadows may overlap higher (different) objects if the shadow plane and shape drawing planes overlap on the screen.

The shadowWeight parameter is used as in the perspective property set to apply additional scaling to the perspective parameters - these are divided by the weight.

Shadow transformations are independent of the perspective transformation applied to a shape - either the perspective transformation or the shadow transformation is used as appropriate.
<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>shadowType</td>
<td>512</td>
<td>MSOSHADOFTYPE</td>
<td>Offset</td>
<td>97</td>
<td>Type of effect</td>
</tr>
<tr>
<td>shadowColor</td>
<td>513</td>
<td>Extended Color</td>
<td>0x808080</td>
<td>97</td>
<td>Foreground color</td>
</tr>
<tr>
<td>shadowHighlight</td>
<td>514</td>
<td>Extended Color</td>
<td>0xCBCBCB</td>
<td>97</td>
<td>Embossed color</td>
</tr>
<tr>
<td>shadowCrMod</td>
<td>515</td>
<td>MSOCLR</td>
<td>undefined</td>
<td>97</td>
<td>Modification for BW views</td>
</tr>
<tr>
<td>shadowOpacity</td>
<td>516</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>97</td>
<td>Fixed 16.16</td>
</tr>
<tr>
<td>shadowOffsetX</td>
<td>517</td>
<td>LONG</td>
<td>25400</td>
<td>97</td>
<td>Offset shadow</td>
</tr>
<tr>
<td>shadowOffsetY</td>
<td>518</td>
<td>LONG</td>
<td>25400</td>
<td>97</td>
<td>Offset shadow</td>
</tr>
<tr>
<td>shadowSecondOffsetX</td>
<td>519</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Double offset shadow</td>
</tr>
<tr>
<td>shadowSecondOffsetY</td>
<td>520</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Double offset shadow</td>
</tr>
<tr>
<td>shadowScaleXToX</td>
<td>521</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>97</td>
<td>16.16</td>
</tr>
<tr>
<td>shadowScaleYToX</td>
<td>522</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>16.16</td>
</tr>
<tr>
<td>shadowScaleXToY</td>
<td>523</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>16.16</td>
</tr>
<tr>
<td>shadowScaleYToY</td>
<td>524</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>97</td>
<td>16.16</td>
</tr>
<tr>
<td>shadowPerspectiveX</td>
<td>525</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>16.16 / weight</td>
</tr>
<tr>
<td>shadowPerspectiveY</td>
<td>526</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>16.16 / weight</td>
</tr>
<tr>
<td>shadowWeight</td>
<td>527</td>
<td>LONG</td>
<td>1&lt;&lt;8</td>
<td>97</td>
<td>scaling factor</td>
</tr>
<tr>
<td>shadowOriginX</td>
<td>528</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>shadowOriginY</td>
<td>529</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>fShadow</td>
<td>574</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Any shadow?</td>
</tr>
<tr>
<td>fshadowObscured</td>
<td>575</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Excel5-style shadow</td>
</tr>
</tbody>
</table>

**Perspective Style**

This is just a 2D transformation matrix (3x3). Specifying peculiar values will cause the shape to render completely outside its geometry - normally clients will constrain the values to get reasonable results. The transformation may be applied at various times as the geometry is processed, this affects the behavior of the perspective which results in the same way as the corresponding shadow perspective types.

- `msoxformAbsolute`  Equivalent to msoshadowRich
- `msoxformShape`  Equivalent to msoshadowShape
- `msoxformDrawing` Equivalent to msoshadowDrawing

In the case of perspective the `msoxformShape` form is the default - the perspective will then scale proportionally with the overall shape scaling.

All parameters except the weight and offset elements are 16.16 fixed-point numbers. The offset is interpreted according to the perspective type, if the type is `msoxformShape` then the offset is assumed
to be a 16.16 fixed point value, otherwise it is assumed to be an integral value (effectively a number of EMUs). The weight acts as an additional divisor for the perspectivePerspectiveX/Y elements - the values in the transformation matrix are obtained by dividing the property values by the weight, the default value of 256 gives a useful range of values for the msoxformShape case, for the other cases the weight should normally be around 1 to 256 times the size of the shape in the coordinate space.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>perspectiveType</td>
<td>576</td>
<td>MSOXFORM</td>
<td>Shape</td>
<td>97</td>
<td>Where transform applies</td>
</tr>
<tr>
<td>perspectiveOffsetX</td>
<td>577</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>perspectiveOffsetY</td>
<td>578</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>perspectiveScaleXToX</td>
<td>579</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>perspectiveScaleYToX</td>
<td>580</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>perspectiveScaleXToY</td>
<td>581</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>perspectiveScaleYToY</td>
<td>582</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>perspectivePerspectiveX</td>
<td>583</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>perspectivePerspectiveY</td>
<td>584</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>perspectiveWeight</td>
<td>585</td>
<td>LONG</td>
<td>1&lt;&lt;8</td>
<td>97</td>
<td>Scaling factor</td>
</tr>
<tr>
<td>perspectiveOriginX</td>
<td>586</td>
<td>LONG</td>
<td>1&lt;&lt;15</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>perspectiveOriginY</td>
<td>587</td>
<td>LONG</td>
<td>1&lt;&lt;15</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>fPerspective</td>
<td>639</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>On/off</td>
</tr>
</tbody>
</table>
3D Object

Material properties of a 3D object. A 3D effect overrides the fill and line effects and corresponding colors. Extrusion depths are always specified in absolute units.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c3DSpecularAmt</td>
<td>640</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Fixed-point 16.16</td>
</tr>
<tr>
<td>c3DDiffuseAmt</td>
<td>641</td>
<td>LONG</td>
<td>65536</td>
<td>97</td>
<td>Fixed-point 16.16</td>
</tr>
<tr>
<td>c3DShininess</td>
<td>642</td>
<td>LONG</td>
<td>5</td>
<td>97</td>
<td>Default gives OK results</td>
</tr>
<tr>
<td>c3DEdgeThickness</td>
<td>643</td>
<td>LONG</td>
<td>12700</td>
<td>97</td>
<td>Specular edge thickness</td>
</tr>
<tr>
<td>C3DExtrudeForward</td>
<td>644</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Distance of extrusion in EMUs</td>
</tr>
<tr>
<td>c3DExtrudeBackward</td>
<td>645</td>
<td>LONG</td>
<td>457200</td>
<td>97</td>
<td>Distance of extrusion in EMUs</td>
</tr>
<tr>
<td>c3DExtrudePlane</td>
<td>646</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Extrusion direction</td>
</tr>
<tr>
<td>c3DExtrusionColor</td>
<td>647</td>
<td>Extended Color</td>
<td>FillThenLine</td>
<td>97</td>
<td>Basic color of extruded part of shape; the lighting model used will determine the exact shades used when rendering.</td>
</tr>
<tr>
<td>c3DCrMod</td>
<td>648</td>
<td>MSOCLR</td>
<td>undefined</td>
<td>97</td>
<td>Modification for BW views</td>
</tr>
<tr>
<td>f3D</td>
<td>700</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Does this shape have a 3D effect?</td>
</tr>
<tr>
<td>fc3DMetallic</td>
<td>701</td>
<td>BOOL</td>
<td>0</td>
<td>97</td>
<td>Use metallic specularity?</td>
</tr>
<tr>
<td>fc3DUseExtrusionColor</td>
<td>702</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>fc3DLightFace</td>
<td>703</td>
<td>BOOL</td>
<td>TRUE</td>
<td>97</td>
<td></td>
</tr>
</tbody>
</table>

3D Style

Properties of a 3D view; note that distances are in drawing units.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c3DYRotationAngle</td>
<td>704</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>degrees (16.16) about y axis</td>
</tr>
<tr>
<td>c3DXRotationAngle</td>
<td>705</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>degrees (16.16) about x axis</td>
</tr>
<tr>
<td>c3DRotationAxisX</td>
<td>706</td>
<td>LONG</td>
<td>100</td>
<td>97</td>
<td>These specify the rotation axis; only their relative magnitudes matter.</td>
</tr>
<tr>
<td>c3DRotationAxisY</td>
<td>707</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>These specify the rotation axis; only their relative magnitudes matter.</td>
</tr>
<tr>
<td>c3DRotationAxisZ</td>
<td>708</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>These specify the rotation axis; only their relative magnitudes matter.</td>
</tr>
</tbody>
</table>
c3DRotationAngle 709  LONG  0  97  degrees (16.16) about axis

**Shape**

Miscellaneous properties of a single shape which do not apply to group shapes.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c3DRotationAngle</td>
<td>709</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>degrees (16.16) about axis</td>
</tr>
<tr>
<td>c3DRotationCenterX</td>
<td>710</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>rotation center x (16.16 or g-units)</td>
</tr>
<tr>
<td>c3DRotationCenterY</td>
<td>711</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>rotation center y (16.16 or g-units)</td>
</tr>
<tr>
<td>c3DRotationCenterZ</td>
<td>712</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>rotation center z (absolute (emus))</td>
</tr>
<tr>
<td>c3DRotateMode</td>
<td>713</td>
<td><strong>MSO3D_RENDERMODE</strong></td>
<td>FullRender</td>
<td>97</td>
<td>Full, wireframe, or bcube</td>
</tr>
<tr>
<td>c3DTolerance</td>
<td>714</td>
<td>LONG</td>
<td>30000</td>
<td>97</td>
<td>pixels (16.16)</td>
</tr>
<tr>
<td>c3DXViewpoint</td>
<td>715</td>
<td>LONG</td>
<td>1250000</td>
<td>97</td>
<td>X viewpoint (emus)</td>
</tr>
<tr>
<td>c3DYViewpoint</td>
<td>716</td>
<td>LONG</td>
<td>-1250000</td>
<td>97</td>
<td>Y viewpoint (emus)</td>
</tr>
<tr>
<td>c3DZViewpoint</td>
<td>717</td>
<td>LONG</td>
<td>9000000</td>
<td>97</td>
<td>Z viewpoint distance (emus)</td>
</tr>
<tr>
<td>c3DOriginX</td>
<td>718</td>
<td>LONG</td>
<td>32768</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>c3DOriginY</td>
<td>719</td>
<td>LONG</td>
<td>-32768</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>c3DSkewAngle</td>
<td>720</td>
<td>LONG</td>
<td>-8847360</td>
<td>97</td>
<td>degree (16.16) skew angle</td>
</tr>
<tr>
<td>c3DSkewAmount</td>
<td>721</td>
<td>LONG</td>
<td>50</td>
<td>97</td>
<td>Percentage skew amount</td>
</tr>
<tr>
<td>c3DAmbientIntensity</td>
<td>722</td>
<td>LONG</td>
<td>20000</td>
<td>97</td>
<td>Fixed point intensity</td>
</tr>
<tr>
<td>c3DKeyX</td>
<td>723</td>
<td>LONG</td>
<td>50000</td>
<td>97</td>
<td>Key light source direction</td>
</tr>
<tr>
<td>c3DKeyY</td>
<td>724</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>only their relative magnitudes matter</td>
</tr>
<tr>
<td>c3DKeyZ</td>
<td>725</td>
<td>LONG</td>
<td>10000</td>
<td>97</td>
<td>Fixed point intensity</td>
</tr>
<tr>
<td>c3DKeyIntensity</td>
<td>726</td>
<td>LONG</td>
<td>38000</td>
<td>97</td>
<td>Fill light source direction</td>
</tr>
<tr>
<td>c3DFillX</td>
<td>727</td>
<td>LONG</td>
<td>-50000</td>
<td>97</td>
<td>only their relative magnitudes matter</td>
</tr>
<tr>
<td>c3DFillY</td>
<td>728</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Fill light source direction</td>
</tr>
<tr>
<td>c3DFillZ</td>
<td>729</td>
<td>LONG</td>
<td>10000</td>
<td>97</td>
<td>Magnitudes matter</td>
</tr>
<tr>
<td>c3DFillIntensity</td>
<td>730</td>
<td>LONG</td>
<td>38000</td>
<td>97</td>
<td>Fixed point intensity</td>
</tr>
<tr>
<td>fc3DConstrainRotation</td>
<td>763</td>
<td>BOOL</td>
<td>TRUE</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>fc3DRotationCenterAuto</td>
<td>764</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>fc3DParallel</td>
<td>765</td>
<td>BOOL</td>
<td>1</td>
<td>97</td>
<td>Parallel projection?</td>
</tr>
<tr>
<td>fc3DKeyHarsh</td>
<td>766</td>
<td>BOOL</td>
<td>1</td>
<td>97</td>
<td>Is key lighting harsh?</td>
</tr>
<tr>
<td>fc3DFillHarsh</td>
<td>767</td>
<td>BOOL</td>
<td>0</td>
<td>97</td>
<td>Is fill lighting harsh?</td>
</tr>
<tr>
<td>Name</td>
<td>Key</td>
<td>Type</td>
<td>Value</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-------</td>
<td>----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>hspMaster</td>
<td>769</td>
<td>MSOHSP</td>
<td>NULL</td>
<td>master shape</td>
<td></td>
</tr>
<tr>
<td>cxstyle</td>
<td>771</td>
<td>MSOCXSTYLE</td>
<td>None</td>
<td>Type of connector</td>
<td></td>
</tr>
<tr>
<td>bWMode</td>
<td>772</td>
<td>MSOBWMODE</td>
<td>Automatic</td>
<td>Settings for modifications to be made when in different forms of black-and-white mode.</td>
<td></td>
</tr>
<tr>
<td>bWModePureBW</td>
<td>773</td>
<td>MSOBWMODE</td>
<td>Automatic</td>
<td>Settings for modifications to be made when in different forms of black-and-white mode.</td>
<td></td>
</tr>
<tr>
<td>bWModeBW</td>
<td>774</td>
<td>MSOBWMODE</td>
<td>Automatic</td>
<td>Settings for modifications to be made when in different forms of black-and-white mode.</td>
<td></td>
</tr>
<tr>
<td>idDiscussAnchor</td>
<td>775</td>
<td>LONG</td>
<td>STD</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>dgmLayout</td>
<td>777</td>
<td>MSODGMLO</td>
<td>msodgmlo Nil</td>
<td>Node layout</td>
<td></td>
</tr>
<tr>
<td>dgmNodeKind</td>
<td>778</td>
<td>DGMNK</td>
<td>-1</td>
<td>XP</td>
<td></td>
</tr>
<tr>
<td>dgmLayoutMRU</td>
<td>779</td>
<td>MSODGMLO</td>
<td>msodgmlo Nil</td>
<td>Most recently used layout for its child</td>
<td></td>
</tr>
<tr>
<td>wzEquationXML</td>
<td>780</td>
<td>char*</td>
<td>NULL</td>
<td>2007</td>
<td></td>
</tr>
<tr>
<td>fPolicyLabel</td>
<td>822</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2007</td>
<td></td>
</tr>
<tr>
<td>fPolicyBarcode</td>
<td>823</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2007</td>
<td></td>
</tr>
<tr>
<td>fFlipHQFE5152</td>
<td>824</td>
<td>BOOL</td>
<td>FALSE</td>
<td>XP</td>
<td></td>
</tr>
</tbody>
</table>

This property is present if the shape represents an equation generated by Office 2007 or later. The property is a string of XML representing a Word 2003 XML document. The original equation is stored within the “oMathPara” tag within the document. Refer to the Office Open XML documentation for details on this XML representation of equations. If the document containing the shape is opened in Office 2007 or later, the shape is replaced with the equation in this document.

True if the shape is a policy label representing metadata about a document.

True if the shape represents a barcode as part of a barcode policy for record management.

The value of this property should match the value of
the fFlipH property (in the transform property set) if the pib property exists (in the blip property set.)

The value of this property should match the value of the fFlipV property (in the transform property set) if the pib property exists (in the blip property set.)

fPreferRelativeResize 827  BOOL  FALSE  97  For UI only. Prefer relative resizing.

fLockShapeType 828  BOOL  FALSE  97  Lock the shape type (don’t allow Change Shape)

fInitiator 829  BOOL  NULL  97  Set by the solver

fDeleteAttachedObject 830  BOOL  FALSE  97

fBackground 831  BOOL  FALSE  97  If TRUE, this is the background shape.

---

**Callout**

Properties of a callout shape.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>spcot</td>
<td>832</td>
<td>MSOSPCOT</td>
<td>TwoSegment</td>
<td>97</td>
<td>Callout type</td>
</tr>
<tr>
<td>dxyCalloutGap</td>
<td>833</td>
<td>LONG</td>
<td>1/12 inch</td>
<td>97</td>
<td>Distance from box to first point. (EMUs)</td>
</tr>
<tr>
<td>spcoa</td>
<td>834</td>
<td>MSOSPCOA</td>
<td>Any</td>
<td>97</td>
<td>Callout angle</td>
</tr>
<tr>
<td>spcod</td>
<td>835</td>
<td>MSOSPCOD</td>
<td>Specified</td>
<td>97</td>
<td>Callout drop type</td>
</tr>
<tr>
<td>dxyCalloutDropSpecified</td>
<td>836</td>
<td>LONG</td>
<td>9 points</td>
<td>97</td>
<td>if msospcodSpecified, the actual drop distance</td>
</tr>
<tr>
<td>dxyCalloutLengthSpecified</td>
<td>837</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>if fCalloutLengthSpecified, the actual distance</td>
</tr>
<tr>
<td>fCallout</td>
<td>889</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Is the shape a callout?</td>
</tr>
<tr>
<td>fCalloutAccentBar</td>
<td>890</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>does callout have accent bar</td>
</tr>
<tr>
<td>fCalloutTextBorder</td>
<td>891</td>
<td>BOOL</td>
<td>TRUE</td>
<td>97</td>
<td>does callout have a text border</td>
</tr>
<tr>
<td>fCalloutMinusX</td>
<td>892</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>If true, callout tail is to the right of the box</td>
</tr>
<tr>
<td>fCalloutMinusY</td>
<td>893</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>If true, callout tail is</td>
</tr>
</tbody>
</table>
### Group Shape

Miscellaneous shape properties that can apply to group shapes.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wzName</td>
<td>896</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>97</td>
<td>Shape Name (present only if explicitly set)</td>
</tr>
<tr>
<td>wzDescription</td>
<td>897</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>97</td>
<td>alternate text</td>
</tr>
<tr>
<td>pihlShape</td>
<td>898</td>
<td>IHlink*</td>
<td>NULL</td>
<td>97</td>
<td>The hyperlink in the shape.</td>
</tr>
<tr>
<td>pWrapPolygonVertices</td>
<td>899</td>
<td>IMsoArray</td>
<td>NULL</td>
<td>97</td>
<td>The polygon that text will be wrapped around (Word)</td>
</tr>
<tr>
<td>dxWrapDistLeft</td>
<td>900</td>
<td>LONG</td>
<td>1/8 inch</td>
<td>97</td>
<td>Left wrapping distance from text (Word)</td>
</tr>
<tr>
<td>dyWrapDistTop</td>
<td>901</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Top wrapping distance from text (Word)</td>
</tr>
<tr>
<td>dxWrapDistRight</td>
<td>902</td>
<td>LONG</td>
<td>1/8 inch</td>
<td>97</td>
<td>Right wrapping distance from text (Word)</td>
</tr>
<tr>
<td>dyWrapDistBottom</td>
<td>903</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Bottom wrapping distance from text (Word)</td>
</tr>
<tr>
<td>lidRegroup</td>
<td>904</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Regroup ID</td>
</tr>
<tr>
<td>groupLeft</td>
<td>905</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>The group's coordinate rectangle</td>
</tr>
<tr>
<td>groupTop</td>
<td>906</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>groupRight</td>
<td>907</td>
<td>LONG</td>
<td>20000</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>groupBottom</td>
<td>908</td>
<td>LONG</td>
<td>20000</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>wzTooltip</td>
<td>909</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>Tooltip for the hyperlink in the shape.</td>
</tr>
<tr>
<td>wzScript</td>
<td>910</td>
<td>WCHAR*</td>
<td>STD</td>
<td>2000</td>
<td>Script (JavaScript, VBScript etc) attached to shape</td>
</tr>
<tr>
<td>posh</td>
<td>911</td>
<td>MSOPH</td>
<td>msophAbs</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>posrelh</td>
<td>912</td>
<td>MSOPRH</td>
<td>msoprhText</td>
<td>2000</td>
<td></td>
</tr>
<tr>
<td>posv</td>
<td>913</td>
<td>MSOPV</td>
<td>msopvAbs</td>
<td>2000</td>
<td></td>
</tr>
</tbody>
</table>

- If true, then we occasionally invert the drop distance.
- If true, we look at dxyCalloutLengthSpecified.
<table>
<thead>
<tr>
<th>Field</th>
<th>Code</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>posrelv</td>
<td>914</td>
<td>MSOPRV</td>
<td>msoprvT ext</td>
</tr>
<tr>
<td>pctHR</td>
<td>915</td>
<td>LONG</td>
<td>Percentage width for a horizontal rule</td>
</tr>
<tr>
<td>alignHR</td>
<td>916</td>
<td>LONG</td>
<td>Alignment for an HR; left == 0, center == 1, right == 2</td>
</tr>
<tr>
<td>dxHeightHR</td>
<td>917</td>
<td>LONG</td>
<td>Height for an HR</td>
</tr>
<tr>
<td>dxWidthHR</td>
<td>918</td>
<td>LONG</td>
<td>Width for an HR</td>
</tr>
<tr>
<td>wzScriptExtAttr</td>
<td>919</td>
<td>WCHAR*</td>
<td>STD</td>
</tr>
<tr>
<td>scriptLang</td>
<td>920</td>
<td>LONG</td>
<td>1</td>
</tr>
<tr>
<td>wzScriptIdAttr</td>
<td>921</td>
<td>WCHAR*</td>
<td>STD</td>
</tr>
<tr>
<td>wzScriptLangAttr</td>
<td>922</td>
<td>WCHAR*</td>
<td>STD</td>
</tr>
<tr>
<td>borderTopColor</td>
<td>923</td>
<td>COLORREF</td>
<td>MSOCOL ORNONE</td>
</tr>
<tr>
<td>borderLeftColor</td>
<td>924</td>
<td>COLORREF</td>
<td>MSOCOL ORNONE</td>
</tr>
<tr>
<td>borderBottomColor</td>
<td>925</td>
<td>COLORREF</td>
<td>MSOCOL ORNONE</td>
</tr>
<tr>
<td>borderRightColor</td>
<td>926</td>
<td>COLORREF</td>
<td>MSOCOL ORNONE</td>
</tr>
<tr>
<td>tableProperties</td>
<td>927</td>
<td>LONG</td>
<td>0</td>
</tr>
<tr>
<td>tableRowProperties</td>
<td>928</td>
<td>IMsoArray y*</td>
<td>NULL</td>
</tr>
<tr>
<td>scriptHtmlLocation</td>
<td>929</td>
<td>LONG</td>
<td>2</td>
</tr>
<tr>
<td>wzApplet</td>
<td>930</td>
<td>WCHAR*</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Extended Script Attributes (other than Lang, Id) of script(VBScript etc) attached to shape

Script Language of script attached to shape (JavaScript, VBScript or other)

Id Script Attribute of script(VBScript etc) attached to shape

Lang Script Attribute of script(VBScript etc) attached to shape

Top border color (WORD)

Left border color (WORD)

Bottom border color (WORD)

Right border color (WORD)

Flags that indicate whether the group shape represents a PowerPoint table.

Row heights if the group shape is a PowerPoint table. Array entries are 32-bit integers, where each entry is a row height in PowerPoint master coordinates.

Script location

Applet Body - not really a shape - visual cue to indicate presence of
<table>
<thead>
<tr>
<th>Field</th>
<th>ID</th>
<th>Type</th>
<th>Value</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>wzFrameTrgtUnused</td>
<td>932</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wzWebBot</td>
<td>933</td>
<td>WCHAR*</td>
<td>STD</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wzAppletArg</td>
<td>934</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wzAccessBlob</td>
<td>936</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>XP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>metroBlob</td>
<td>937</td>
<td>IByteStr</td>
<td>NULL</td>
<td>2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eam*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dhgt</td>
<td>938</td>
<td>DHGT</td>
<td>0</td>
<td>2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fLayoutInCell</td>
<td>944</td>
<td>BOOL</td>
<td>TRUE</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fIsBullet</td>
<td>945</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fStandardHR</td>
<td>946</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fNoshadeHR</td>
<td>947</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fHorizRule</td>
<td>948</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fUserDrawn</td>
<td>949</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fAllowOverlap</td>
<td>950</td>
<td>BOOL</td>
<td>TRUE</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fReallyHidden</td>
<td>951</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Relative Transform

Defines the size and location of the shape in the parent group or drawing. The coordinates are relative to the position of the parent group or drawing. The units are relative to the rcg of the parent. The Relative Transform and Transform property sets express the same information in different ways. For top-level shape they are equivalent.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>relLeft</td>
<td>0</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Bounds of the unrotated shape expressed as top left and bottom right in drawing units.</td>
</tr>
<tr>
<td>relTop</td>
<td>1</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>relRight</td>
<td>2</td>
<td>LONG</td>
<td>1</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>relBottom</td>
<td>3</td>
<td>LONG</td>
<td>1</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>relRotation</td>
<td>4</td>
<td>LONG</td>
<td>0</td>
<td>97</td>
<td>Rotation is about the top left. Fixed point: 16.16 degrees</td>
</tr>
<tr>
<td>gvRelPage</td>
<td>5</td>
<td>MSOGV</td>
<td>0</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>fRelChangeP</td>
<td>61</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>fRelFlipV</td>
<td>62</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Flip vertically</td>
</tr>
<tr>
<td>fRelFlipH</td>
<td>63</td>
<td>BOOL</td>
<td>FALSE</td>
<td>97</td>
<td>Flip horizontally</td>
</tr>
</tbody>
</table>

Unknown HTML

ID properties from generic HTML.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wzLineId</td>
<td>1026</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML line element ID</td>
</tr>
<tr>
<td>wzFillId</td>
<td>1027</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML fill element ID</td>
</tr>
<tr>
<td>wzPictureId</td>
<td>1028</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML picture element ID</td>
</tr>
<tr>
<td>wzPathId</td>
<td>1029</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML path element ID</td>
</tr>
<tr>
<td>wzShadowId</td>
<td>1030</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML shadow element ID</td>
</tr>
<tr>
<td>wzPerspectiveId</td>
<td>1031</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML perspective element ID</td>
</tr>
<tr>
<td>wzGtextId</td>
<td>1032</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML geotext element ID</td>
</tr>
<tr>
<td>wzFormulaeId</td>
<td>1033</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML formula element ID</td>
</tr>
<tr>
<td>wzHandlesId</td>
<td>1034</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML handle element ID</td>
</tr>
<tr>
<td>wzCalloutId</td>
<td>1035</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML callout element ID</td>
</tr>
<tr>
<td>wzLockId</td>
<td>1036</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML lock element ID</td>
</tr>
<tr>
<td>wzTextId</td>
<td>1037</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2000</td>
<td>XML text element ID</td>
</tr>
</tbody>
</table>
### Diagram

Properties related to Canvases and Diagrams

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dgmt</td>
<td>1280</td>
<td>MSODGDM</td>
<td>msodgmNil</td>
<td>XP</td>
<td>Diagram/Canvas type</td>
</tr>
<tr>
<td>dgmStyle</td>
<td>1281</td>
<td>MSODGMS</td>
<td>msodgmsNil</td>
<td>XP</td>
<td>Diagram style</td>
</tr>
<tr>
<td>pRelationTbl</td>
<td>1284</td>
<td>IMsoArray</td>
<td>NULL</td>
<td>XP</td>
<td>Table of shape relationships. The array is a list of DGMRL structures. Each entry holds SPIDs for source, destination, and connector shapes</td>
</tr>
<tr>
<td>dgmScaleX</td>
<td>1285</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>XP</td>
<td>Scale factor for width of diagram</td>
</tr>
<tr>
<td>dgmScaleY</td>
<td>1286</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>XP</td>
<td>Scale factor for height of diagram</td>
</tr>
<tr>
<td>dgmDefaultFontSize</td>
<td>1287</td>
<td>LONG</td>
<td>-1</td>
<td>XP</td>
<td>Font size for new nodes</td>
</tr>
<tr>
<td>dgmConstrainBounds</td>
<td>1288</td>
<td>IMsoArray</td>
<td>NULL</td>
<td>XP</td>
<td>An array of four LONGs recording the left, top, right, bottom bounds that the nodes are constrained to.</td>
</tr>
<tr>
<td>dgmBaseTextScale</td>
<td>1289</td>
<td>LONG</td>
<td>1&lt;&lt;16</td>
<td>2003</td>
<td>Scale factor for base text size</td>
</tr>
<tr>
<td>fBorderlessCanvas</td>
<td>1338</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2007</td>
<td>Indicates that the canvas is intended to be used for inking. The canvas does not have a border around it unless the user is actively inking. As soon as the user stops inking, the border goes away. All ink that was drawn over the canvas will stick to the canvas.</td>
</tr>
<tr>
<td>fNonStickyInkCanvas</td>
<td>1339</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2007</td>
<td>A sub-property of fBorderlessCanvas; it won’t exist on its own. A non-sticky canvas is a borderless canvas that ink does not necessarily bind to if the user inks over it. The ink should not crop if the user draws outside the canvas, and the canvas won’t grow to accommodate the new stroke.</td>
</tr>
<tr>
<td>fDoFormat</td>
<td>1340</td>
<td>BOOL</td>
<td>FALSE</td>
<td>XP</td>
<td>True if auto format enabled</td>
</tr>
<tr>
<td>fReverse</td>
<td>1341</td>
<td>BOOL</td>
<td>FALSE</td>
<td>XP</td>
<td>True if diagram layout is reversed</td>
</tr>
<tr>
<td>fDoLayout</td>
<td>1342</td>
<td>BOOL</td>
<td>TRUE</td>
<td>XP</td>
<td>True if layout is needed</td>
</tr>
</tbody>
</table>
Line Left Style

On rectangular shapes, OfficeArt supports having different line styles of each of the four borders (left, top, right, bottom, and on the boundary between columns of text. When a rectangular shape has no line (fLine property is FALSE), the display engine will derive the line style for the four borders from the left, top, right and bottom property sets.

Thus there is a Line Left Style property set that controls the rendering of the left boundary of a rectangle, and is identical to the Line Style property set in every way, except that the properties count up from 1344 (lineLeftColor) instead of 448 (lineColor.)

Line Top Style

The Line Top Style property set controls the rendering of the top boundary of a rectangle, and is identical to the Line Style property set in every way, except that the properties count up from 1408 (lineTopColor) instead of 448 (lineColor.)

Line Right Style

The Line Top Style property set controls the rendering of the right boundary of a rectangle, and is identical to the Line Style property set in every way, except that the properties count up from 1472 (lineRightColor) instead of 448 (lineColor.)

Line Bottom Style

The Line Bottom Style property set controls the rendering of the bottom boundary of a rectangle, and is identical to the Line Style property set in every way, except that the properties count up from 1536 (lineBottomColor) instead of 448 (lineColor.)

Line Column Style

The Line Column Style property set controls the rendering of column borders between text, and is identical to the Line Style property set in every way, except that the properties count up from 1600 (lineLeftColor) instead of 448 (lineColor.)

Line Top Style

The Line Top Style property set controls the rendering of the top boundary of a rectangle, and is identical to the Line Style property set in every way, except that the properties count up from 1408 (lineLeftColor) instead of 448 (lineColor.)

Web Component

Properties related to Web Component functionality

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webComponentWzHtml</td>
<td>1664</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>XP</td>
<td>HTML content of web comp.</td>
</tr>
<tr>
<td>webComponentWzName</td>
<td>1665</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>XP</td>
<td>Friendly name of web comp.</td>
</tr>
<tr>
<td>webComponentWzUrl</td>
<td>1666</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>XP</td>
<td>URL of web comp.</td>
</tr>
</tbody>
</table>
webComponentWzProperties 1667 WCHAR* NULL XP Property string for web comp.

fIsWebComponent 1727 BOOL FALSE XP TRUE if this shape is a web comp.

**Clip**

This property set defines an additional clipping path for a shape or group

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pVerticesClip</td>
<td>1728</td>
<td>IMsoArray</td>
<td>NULL</td>
<td>XP</td>
<td>Vertices of clipping path. Format is same as of pVertices property in the Geometry property set</td>
</tr>
<tr>
<td>pSegmentInfo</td>
<td>1729</td>
<td>IMsoArray</td>
<td>NULL</td>
<td>XP</td>
<td>Segment info of clipping path. Format is same as of pSegmentInfo property in the Geometry property set</td>
</tr>
<tr>
<td>shapePathClip</td>
<td>1730</td>
<td>MSOSHAPEPATH</td>
<td>linesClosed</td>
<td>XP</td>
<td>Type of clipping path. Format is same as of shapePath property in the Geometry property set</td>
</tr>
<tr>
<td>fClipToWrap</td>
<td>1790</td>
<td>BOOL</td>
<td>FALSE</td>
<td>XP</td>
<td>If true, shape is clipped to its text tight wrap polygon</td>
</tr>
<tr>
<td>fClippedOK</td>
<td>1791</td>
<td>BOOL</td>
<td>FALSE</td>
<td>XP</td>
<td>If true, additional clipping is enabled</td>
</tr>
</tbody>
</table>

**Ink**

This property set defines properties for Tablet PC Ink data.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pInkData</td>
<td>1792</td>
<td>IMsoInkData</td>
<td>NULL</td>
<td>2003</td>
<td>Ink stroke data for the shape. This data consists of a. 16 byte GUID (CLSID_InkDisp) followed by b. 4 byte unsigned integer, giving the size of the ink blob, followed by c. an ink blob. The ink blob is in the Tablet PC Ink Serialized Format (ISF), more details on the Ink Serialized Format can be found at <a href="http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD86/InkSerializedFormat(ISF)Specification.pdf">http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD86/InkSerializedFormat(ISF)Specification.pdf</a>.</td>
</tr>
<tr>
<td>fInkAnnotation</td>
<td>1852</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2003</td>
<td>TRUE if current shape is ink annotation</td>
</tr>
<tr>
<td>fHitTestInk</td>
<td>1853</td>
<td>BOOL</td>
<td>TRUE</td>
<td>2003</td>
<td>TRUE to allow hit testing on ink strokes</td>
</tr>
</tbody>
</table>
| fRenderShape | 1854  | BOOL          | FALSE   | 2003| TRUE to render geometry/path on a shape with ink (normally these are
hidden for shapes with ink)

**fRenderInk 1855 BOOL FALSE 2003 TRUE to render ink**

## Signature

These properties specify that the shape is a signature line a document generated by Office 2007 or later. A signature line provides a visual representation of a signature in a document that is digitally signed. The properties in this set correspond to attributes of the “signatureline” element in the Office Open XML format. For more details on these properties, refer to the documentation of that element in the Office Open XML documentation.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wzSigSetupId</td>
<td>1921</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2007</td>
<td>GUID representing a place to sign in the doc</td>
</tr>
<tr>
<td>wzSigSetupProvId</td>
<td>192</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2007</td>
<td>GUID representing the signature provider (e.g. Office default is GUID_NULL, Office stamp is well-defined – can look up if you need it, and third-party is CLSID)</td>
</tr>
<tr>
<td>wzSigSetupSuggSigner</td>
<td>1923</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2007</td>
<td>The first line to show in the signature line representing who is suggested as appropriate signer for the signature line.</td>
</tr>
<tr>
<td>wzSigSetupSuggSigner2</td>
<td>1924</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2007</td>
<td>Like above, but the second line.</td>
</tr>
<tr>
<td>wzSigSetupSuggSignerEmail</td>
<td>1925</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2007</td>
<td>Like above, but email address.</td>
</tr>
<tr>
<td>wzSigSetupSignInst</td>
<td>1926</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2007</td>
<td>Instructions to show the user in the signing ceremony.</td>
</tr>
<tr>
<td>wzSigSetupAddlXml</td>
<td>1927</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2007</td>
<td>generic string to store additional information with a signature line.</td>
</tr>
<tr>
<td>wzSigSetupProvUrl</td>
<td>1928</td>
<td>WCHAR*</td>
<td>NULL</td>
<td>2007</td>
<td>url to redirect user if the signature line is from a signature provider not installed.</td>
</tr>
<tr>
<td>fSigSetupShowSignDate</td>
<td>1980</td>
<td>BOOL</td>
<td>TRUE</td>
<td>2007</td>
<td>whether the sign date should be shown in the signed signature line.</td>
</tr>
<tr>
<td>fSigSetupAllowComments</td>
<td>1981</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2007</td>
<td>whether the signing ceremony should allow comments to be entered.</td>
</tr>
<tr>
<td>fSigSetupSignInstSet</td>
<td>1982</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2007</td>
<td>whether a suggested signer is set</td>
</tr>
<tr>
<td>fIsSignatureLine</td>
<td>1983</td>
<td>BOOL</td>
<td>FALSE</td>
<td>2007</td>
<td>whether the shape is a signature line object.</td>
</tr>
</tbody>
</table>
Group Shape 2

Relative position and size properties for shapes and textboxes in Word.

<table>
<thead>
<tr>
<th>Property</th>
<th>PID</th>
<th>Type</th>
<th>Default</th>
<th>Ver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pctHoriz</td>
<td>1984</td>
<td>LONG</td>
<td>msopctSizeNone</td>
<td>2007</td>
<td>Percentage width for a shape (Word)</td>
</tr>
<tr>
<td>pctVert</td>
<td>1985</td>
<td>LONG</td>
<td>msopctSizeNone</td>
<td>2007</td>
<td>Percentage height for a shape (Word)</td>
</tr>
<tr>
<td>pctHorizPos</td>
<td>1986</td>
<td>LONG</td>
<td>msopctPosNone</td>
<td>2007</td>
<td>Percentage horizontal position for a shape (Word)</td>
</tr>
<tr>
<td>pctVertPos</td>
<td>1987</td>
<td>LONG</td>
<td>msopctPosNone</td>
<td>2007</td>
<td>Percentage vertical position for a shape (Word)</td>
</tr>
<tr>
<td>sizerelh</td>
<td>1988</td>
<td>MSOSRH</td>
<td>msosrhPage</td>
<td>2007</td>
<td>relative size horizontal relation (Word)</td>
</tr>
<tr>
<td>sizerelv</td>
<td>1989</td>
<td>MSOSRV</td>
<td>msoprvPage</td>
<td>2007</td>
<td>relative size vertical relation (Word)</td>
</tr>
<tr>
<td>colStart</td>
<td>1990</td>
<td>LONG</td>
<td>0</td>
<td>2007</td>
<td>Starting column (Word)</td>
</tr>
<tr>
<td>colSpan</td>
<td>1991</td>
<td>LONG</td>
<td>0</td>
<td>2007</td>
<td>Number of columns to span (Word)</td>
</tr>
</tbody>
</table>

Appendix A: Change History

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/11/96</td>
<td>Created the document. Left work to be done in the BStore and Complex shape properties.</td>
</tr>
<tr>
<td>1/23/97</td>
<td>Extensively updated and reorganized document. Added overview and shape hierarchy information. Added more complete shape property information.</td>
</tr>
<tr>
<td>4/15/97</td>
<td>Add details on compression of Metafile Blips. See section “Metafile/PICT Blips”</td>
</tr>
<tr>
<td>9/1/03</td>
<td>Update document for Office 2003.</td>
</tr>
</tbody>
</table>

Appendix B: Colors

For each color in a property table there is a “main” property ID. A well formed file that specifies a value for a given color must always specify a four-byte LONG as the value for the “main” property table entry. The colors designated “Extended Color” also have optional “Extended” property IDs in the property table. The values assigned to these extended IDs can define the color more precisely in a different color model (say, CMYK.)

Thus, a color defined in the Office Drawing file format consists of one required “main” property ID and zero or more optional “extended” property IDs. For example, a fill color is defined by the “main” property 385 (fillColor) and the “extended” properties 414, 415, 416, 417, and 422 (fillColorExt, fillColorExtCMY, fillColorExtMod, fillColorExtWzName, and fillColorExtK, respectively.) To reduce redundancy, the “extended” property table IDs are not listed in the tables in the main part of the
document. To see the complete listing of these IDs, refer to the section below discussing “extended" properties.

Interpreting the “main” property

Recall that this value is a four-byte LONG. If the high byte is zero, then the low three bytes contain a normal RGB COLORREF. Otherwise, the value of the high byte is used as a bitfield of flags (notice that the values below are the indices of the bits that are set).

typedef enum
{
    msocolorFlagPaletteIndex,  // PALETTEINDEX macro
    msocolorFlagPaletteRGB,    // PALETTERGB macro
    msocolorFlagSystemRGB,     // MSOSYSTEMRGB
    msocolorFlagSchemeIndex,   // MSOSCHEMECOLOR
    msocolorFlagSysIndex,      // MSOSYSCOLOR
} MSOCOLORINDEX;

Windows defines the first two. The third (SystemRGB) is a flag that it set on an otherwise normal RGB value to indicate to the rendering engine to bypass its normal halftone dithering process and just render the color directly using Windows.

The presence of either of the last two flags indicates that the low three bytes are an index into a predefined array of colors. For SchemeIndex colors, the host application provides the translation to RGB colors when necessary (PowerPoint and Excel both use this). SysIndex colors are indices into colors tracked by Escher itself:

typedef enum
{
    msosyscolorButtonFace,          // COLOR_BTNFACE
    msosyscolorWindowText,          // COLOR_WINDOWTEXT
    msosyscolorMenu,                // COLOR_MENU
    msosyscolorHighlight,           // COLOR_HIGHLIGHT
    msosyscolorHighlightText,       // COLOR_HIGHLIGHTTEXT
    msosyscolorCaptionText,         // COLOR_CAPTIONTEXT
    msosyscolorActiveCaption,       // COLOR_ACTIVECAPTION
    msosyscolorButtonHighlight,     // COLOR_BTNHIGHLIGHT
    msosyscolorButtonShadow,        // COLOR_BTNSHADOW
    msosyscolorButtonText,          // COLOR_BTNTEXT
    msosyscolorGrayText,            // COLOR_GRAYTEXT
    msosyscolorInactiveCaption,     // COLOR_INACTIVECAPTION
    msosyscolorInactiveCaptionText, // COLOR_INACTIVECAPTIONTEXT
    msosyscolorInfoBackground,      // COLOR_INFOBK
    msosyscolorInfoText,            // COLOR_INFOTEXT
    msosyscolorMenuText,            // COLOR_MENUTEXT
    msosyscolorScrollbar,           // COLOR_SCROLLBAR
    msosyscolorWindow,              // COLOR_WINDOW
    msosyscolorWindowFrame,         // COLOR_WINDOWFRAME
    msosyscolor3DLight,             // COLOR_3DLIGHT
    msosyscolorMax,                 // Count of system colors

    msocolorFillColor =0xF0,  // Use the fillColor property
    msocolorLineOrFillColor,  // Use the line color only if there is a line
    msocolorLineColor,        // Use the lineColor property
    msocolorShadowColor,      // Use the shadow color
msocolorThis,         // Use this color (only valid as described below)
msocolorFillBackColor,   // Use the fillBackColor property
msocolorLineBackColor,   // Use the lineBackColor property
msocolorFillThenLine,     // Use the fillColor unless no fill and line
msocolorIndexMask =0xFF,   // Extract the color index

msocolorProcessMask       =0xFFFF00, // All the processing bits
msocolorModificationMask  =0x0F00, // Just the function
msocolorModFlagMask       =0xF000, // Just the additional flags
msocolorDarken            =0x0100,  // Darken color by parameter/255
msocolorLighten           =0x0200,  // Lighten color by parameter/255
msocolorAdd               =0x0300,  // Add grey level RGB(param,param,param)
msocolorSubtract          =0x0400,  // Subtract grey level RGB(p,p,p)
msocolorReverseSubtract   =0x0500,  // Subtract from grey level RGB(p,p,p)
/* In the following "black" means maximum component value, white minimum.  
The operation is per component, to guarantee white combine 
with msocolorGray */
msocolorBlackWhite        =0x0600,  // Black if < uParam, else white (>=)
msocolorInvert            =0x2000,  // Invert color (at the *end*)
msocolorInvert128         =0x4000,  // Invert by toggling the top bit
msocolorGray              =0x8000,  // Make the color gray (before the above!)
msocolorBParamMask        =0xFF0000, // Parameter used as above
msocolorBParamShift       =16,      // To extract the parameter value
}
MSOSYS COLORINDEX;

Interpreting “extended” properties

For a given “Extended” color the following can be specified:

MSOCLR color – The “main” color property for this extended color.

- If none of the extended properties is set, it contains the full color definition which is either a straight RGB, or a simple scheme color, or a derived color. The format of this property is described in the section on “Interpreting the Main property”.

- If the other extended properties are set, this field must contain the ‘resolved’ RGB that is computed by applying colorExtMod to colorExt. When the main and extended values for a given color are inconsistent, (i.e. colorExt does not match the sRGB translation of colorExtCMY / colorExtK / colorExtWzName) an implementer should respect the main value and discard the extended values.

- In either case, implementers interested only in the sRGB color space (the usual case) can use the value of this property as the color and ignore all the extended color properties.

MSOCLR colorExt – this is base color. Resolving it (applying colorExtMod modification to it) we will get color. It’s used in the following situations:

- if colorExtWzName is set, this must contain the RGB approximation of the CMS color.

- if colorExtCMY and colorExtK are set, this must contain the RGB approximation of the CMYK or spot color they contain.

- Otherwise, contains either a ‘base’ RGB or ‘base’ scheme color. colorExtMod should be tint or shade color modification.
MSOCLR **colorExtMod** – if set, contains a tint or shade (color modification) to apply to **colorExt**. This should be set if **colorExt** is set, otherwise it should be **MSOCOLORMODUNDEFINED**.

LONG **colorExtCMY, colorExtK** – these two properties comprise the low-order (**colorExtCMY**) and high-order (**colorExtK**) bits of an **MSOINKCOLOR** (this type is defined below). They are actually mis-named as they can contain information other than CMYK. Used in the following cases:

- If **colorExtWzName** is set, these properties must contain the CMYK approximation of the CMS color.
- Otherwise, if set, they contain a CMYK color and/or spot color(s). (Again, see **MSOINKCOLOR** for full details.)

WCHAR * **colorExtWzName** - if set, contains a string that identifies a CMS (Color Management System) color. The format of the string is documented in the MSO headers, but only the UI code really cares.

Here are the sets of property IDs that comprise each extended color:

<table>
<thead>
<tr>
<th>Property</th>
<th>color</th>
<th>colorExt</th>
<th>colorExtMod</th>
<th>colorExtCMY</th>
<th>colorExtK</th>
<th>colorExtWzName</th>
</tr>
</thead>
<tbody>
<tr>
<td>pictureTransparent</td>
<td>263</td>
<td>277</td>
<td>279</td>
<td>278</td>
<td>281</td>
<td>280</td>
</tr>
<tr>
<td>pictureRecolor</td>
<td>282</td>
<td>283</td>
<td>285</td>
<td>284</td>
<td>287</td>
<td>286</td>
</tr>
<tr>
<td>fillColor</td>
<td>385</td>
<td>414</td>
<td>416</td>
<td>415</td>
<td>422</td>
<td>417</td>
</tr>
<tr>
<td>fillBackColor</td>
<td>387</td>
<td>418</td>
<td>420</td>
<td>419</td>
<td>423</td>
<td>421</td>
</tr>
<tr>
<td>lineColor</td>
<td>448</td>
<td>473</td>
<td>475</td>
<td>474</td>
<td>481</td>
<td>476</td>
</tr>
<tr>
<td>lineBackColor</td>
<td>450</td>
<td>477</td>
<td>479</td>
<td>478</td>
<td>482</td>
<td>480</td>
</tr>
<tr>
<td>shadowColor</td>
<td>513</td>
<td>530</td>
<td>532</td>
<td>531</td>
<td>538</td>
<td>533</td>
</tr>
<tr>
<td>shadowHighlight</td>
<td>514</td>
<td>534</td>
<td>536</td>
<td>535</td>
<td>539</td>
<td>536</td>
</tr>
<tr>
<td>c3DExtrusionColor</td>
<td>647</td>
<td>649</td>
<td>651</td>
<td>650</td>
<td>653</td>
<td>652</td>
</tr>
</tbody>
</table>

The main color properties were all added in Office 97 (hence they should be saved in **Block 1**.) The extended color properties first appeared in Office XP, and so should be written in **Block 3**.

**MSOINKCOLOR**

Finally, here is the definition of the 64-bit MSOINKCOLOR type.

- Bit 0 (lowest order) is an "overprinting" flag. If set it is a hint that when printing color separations, ink from objects underneath this object does not need to be removed.
- Bit 1 is reserved and should be zero.
- Bits 2-4 determine the format of the data stored in bits 6 and up. The possibilities are:

```
typedef enum
{
    msoprocessNone,   // No process – color just contains spot/named inks
    msoprocesssRGB,   // sRGB64 – actually three 16 bit values, R,G,B.
    msoprocessCMYK,   // CMYK as defined by our canonical ICM profile
}
MSOPROCESS;
```

- If the format is msoprocessNone, *spot ink values* (see below) follow starting at bit 5.
• If the format is msoprocesssRGB, a color in the sRGB64 color space follows, with the red channel in bits 5-20, green channel in bits 21-36, and blue channel in bits 37-52. No spot ink values will follow.

• If the format is msoprocessCMYK, a packed CMYK color follows. Bits 5-8 determine which of the Cyan (bit 8), Magenta (bit 7), Yellow (bit 6), and Black (bit 5) inks is present in the color. For each bit that is set, the amount of the corresponding ink present in this MSOINKCOLOR will appear as an 8 bit value in the range 0-255. No bits are used for inks that are not present. The order of the colors (from low-order bits to high-order bits) is C, M, Y, K (=black.) Zero or more spot ink values follow.

Spot ink values are stored as an ink index increment (low-order bits) followed by an ink amount (high-order bits.) The ink index increment’s three lowest order bits are a number taking the following values:

0: end of list; no more spot ink values

1..6: This value is the ink index increment

7: 6 more bits of ink index increment follow. The total ink index increment for this spot ink value is 7 + (the number in these 6 bits.) The maximum permissible value for an ink index increment is 63.

As an implementation scans an MSOINKCOLOR from low-order to high-order bits, it must maintain an ink index, initialized to zero. Each time it encounters an ink index increment, it should add the increment to the current ink index to obtain the next ink index. The ink index is an index into a table of inks stored by the host application (as of Office 2003, only Publisher supports spot ink values.) The format of this table of inks and its location in the binary file are external to OfficeArt and therefore are not described in this document.

The ink amount is an 8 bit number in the range 0-255, with 0 denoting no ink and 255 denoting maximum ink coverage.

Exempli Gratia: Here are three sample MSOINKCOLOR structures to illustrate the definitions above.

<table>
<thead>
<tr>
<th>Flags:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overprint = 0, reserved = 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11 bits unused</th>
<th>1111111111111111</th>
<th>0000000000000000</th>
<th>1111111111111111</th>
<th>001</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue color channel: Max. intensity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green color channel: Min. intensity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red color channel: Max. intensity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process: msoprocesssRGB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spot ink value: End of list (no spot inks in this color)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yellow channel: Intensity 8/255</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cyan channel: Intensity 255/255</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C and Y channels present; M and K channels absent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process: msoprocessCMYK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix C: Shape Types

Internally, a shape type is defined as a fixed set of property values, the most important being the geometry of the shape (the pVertices property, etc.). Each shape stores in itself only those properties that differ from its shape type. When a shape is asked for a property that isn’t in its local table, it looks in the shape type’s table. If the shape type doesn’t define a value for the property, then the property’s default value is used.

In the file format, shapes store their shape types in the instance field of the msofbtSp record.

typedef enum
{
    msosptMin = 0,
    msosptNotPrimitive = msosptMin,
    msosptRectangle = 1,
    msosptRoundRectangle = 2,
    msosptEllipse = 3,
    msosptDiamond = 4,
    msosptIsocelesTriangle = 5,
    msosptRightTriangle = 6,
    msosptParallelogram = 7,
    msosptTrapezoid = 8,
    msosptHexagon = 9,
    msosptOctagon = 10,
    msosptPlus = 11,
    msosptStar = 12,
    msosptArrow = 13,
    msosptThickArrow = 14,
    msosptHomePlate = 15,
    msosptCube = 16,
    msosptBalloon = 17,
    msosptSeal = 18,
    msosptArc = 19,
    msosptLine = 20,
    msosptPlaque = 21,
    msosptCan = 22,
    msosptDonut = 23,
    msosptTextSimple = 24,
    msosptTextOctagon = 25,
    msosptTextHexagon = 26,
    msosptTextCurve = 27,
    msosptTextWave = 28,
    msosptTextRing = 29,
    msosptTextOnCurve = 30,
    msosptTextOnRing = 31,
    msosptStraightConnector1 = 32,
    msosptBentConnector2 = 33,
    msosptBentConnector3 = 34,
    msosptBentConnector4 = 35,
    msosptBentConnector5 = 36,
    msosptCurvedConnector2 = 37,
    msosptCurvedConnector3 = 38,
    msosptCurvedConnector4 = 39,
    msosptCurvedConnector5 = 40,
    msosptCallout1 = 41,
    msosptCallout2 = 42,
    msosptCallout3 = 43,
    msosptCallout4 = 44,
    msosptCallout5 = 45,
    msosptAccentCallout1 = 46,
    msosptAccentCallout2 = 47,
    msosptAccentCallout3 = 48,
    msosptBorderCallout1 = 49,
    msosptBorderCallout2 = 50,
    msosptBorderCallout3 = 51,
    msosptBorderCallout4 = 52,
    msosptRibbon = 53,
    msosptRibbon2 = 54,
    msosptChevron = 55,
    msosptPentagon = 56,
    msosptChevron2 = 57,
};
msosptNoSmoking = 57,
msosptSeal18 = 58,
msosptSeal16 = 59,
msosptSeal32 = 60,
msosptWedgeRectCallout = 61,
msosptWedgeRectCallout = 62,
msosptWedgeEllipseCallout = 63,
msosptWave = 64,
msosptFoldedCorner = 65,
msosptLeftArrow = 66,
msosptDownArrow = 67,
msosptUpArrow = 68,
msosptLeftRightArrow = 69,
msosptUpDownArrow = 70,
msosptIrregularSeal1 = 71,
msosptIrregularSeal2 = 72,
msoptLightningBolt = 73,
msoptHeart = 74,
msoptPictureFrame = 75,
msoptQuadArrow = 76,
msoptLeftArrowCallout = 77,
msoptRightArrowCallout = 78,
msoptUpArrowCallout = 79,
msoptDownArrowCallout = 80,
msoptLeftRightArrowCallout = 81,
msoptUpDownArrowCallout = 82,
msoptQuadArrowCallout = 83,
msoptBevel = 84,
msoptLeftBracket = 85,
msoptRightBracket = 86,
msoptLeftBrace = 87,
msoptRightBrace = 88,
msoptLeftUpArrow = 89,
msoptBentUpArrow = 90,
msoptBentArrow = 91,
msoptSeal24 = 92,
msoptStripedRightArrow = 93,
msoptNotchedRightArrow = 94,
msoptBlockArc = 95,
msoptSmileyFace = 96,
msoptVerticalScroll = 97,
msoptHorizontalScroll = 98,
msoptCircularArrow = 99,
msoptNotchedCircularArrow = 100,
msoptUturnArrow = 101,
msoptCurvedRightArrow = 102,
msoptCurvedLeftArrow = 103,
msoptCurvedUpArrow = 104,
msoptCurvedDownArrow = 105,
msoptCloudCallout = 106,
msoptEllipseRibbon = 107,
msoptEllipseRibbon2 = 108,
msoptFlowChartProcess = 109,
msoptFlowChartDecision = 110,
msosptTextDeflateTop = 165,
msosptTextDeflateInflate = 166,
msosptTextDeflateInflateDeflate = 167,
msosptTextFadeRight = 168,
msosptTextFadeLeft = 169,
msosptTextFadeUp = 170,
msosptTextFadeDown = 171,
msosptTextSlantUp = 172,
msosptTextSlantDown = 173,
msosptTextCanUp = 174,
msosptTextCanDown = 175,
msosptFlowChartAlternateProcess = 176,
msosptFlowChartOffpageConnector = 177,
msosptCallout90 = 178,
msosptAccentCallout90 = 179,
msosptBorderCallout90 = 180,
msosptAccentBorderCallout90 = 181,
msosptLeftRightUpArrow = 182,
msosptSun = 183,
msosptMoon = 184,
} MSOSPT;

msosptBracketPair = 185,
msosptBracePair = 186,
msosptSeal4 = 187,
msosptDoubleWave = 188,
msosptActionButtonBlank = 189,
msosptActionButtonHome = 190,
msosptActionButtonHelp = 191,
msosptActionButtonInformation = 192,
msosptActionButtonForwardNext = 193,
msosptActionButtonBackPrevious = 194,
msosptActionButtonEnd = 195,
msosptActionButtonBeginning = 196,
msosptActionButtonReturn = 197,
msosptActionButtonDocument = 198,
msosptActionButtonSound = 199,
msosptActionButtonMovie = 200,
msosptHostControl = 201,
msosptTextBox = 202,
msosptMax,
msosptNil = 0xFFFF,
Appendix D: AutoShapes

Introduction

This document describes each of the AutoShapes available in Microsoft Office. It provides a detailed description of each shape, including its geometric properties. This document is intended for use in conjunction with the Office Drawing File Format specification, which documents the binary file format used to represent shapes.

The first section of this document describes the meaning of each shape property and provides a mapping to its representation in the binary file format. The section of this document lists each AutoShape and its individual properties.

AutoShapes

AutoShapes are drawing objects with a particular shape that may be customized through smart resizing and adjustments. Multiple disjoint paths and subpaths, and quadratic and cubic curves give shapes rich geometry. Multiple adjust handles which can adjust in two dimensions anywhere inside or outside of a shape, and a robust set of formulas allow very smart adjustment behaviors to be defined. In addition, AutoShapes may contain text sized to fit within the shape.

AutoShapes have been designed with the following consistency guidelines:

- Left or top perimeter. Perimeter adjust handles are used when only one degree of freedom needs to be adjusted, and the vertices that are adjusted are near the perimeter of the shape. Perimeter adjust handles are placed so that they never overlap resize handles. If the shape is symmetrical, the perimeter adjust handle is placed on the left or top perimeter instead of the right or bottom.

- Inner adjustments. For most shapes, the adjust handles do not change the bounding box of the shape; rather, they make an internal adjustment within the bounding box of the shape. Exceptions to this rule include the callout shapes, for which it is useful to place the point independent of the box, and the arc shape.

- Arrow adjust handle. Most of the block arrow shapes use a single adjust handle that controls both the length of the arrowhead and the width of the trunk. Some block arrow shapes have other adjust handles in addition to this one.

- Limo-stretch. Some shapes that have portions that should be constrained to a fixed aspect ratio are designed with limo-stretch to keep those portions at the fixed aspect ratio.

Description of Shape Properties

A table describing the properties of each AutoShape follows this section. In the table, AutoShapes are organized into six categories:
This section describes each property that may be specified for each shape. All coordinates are given in shape coordinates: the origin is at the top-left, positive-x is to the right, positive-y is down. The region from (0,0) to (S,S) maps to the geometry box of the shape (S=21600 is a constant). Adjust values are “inputs” which are usually determined by the location of the adjust handles.
Shape Diagrams

For each shape, a diagram of the shape is shown, and a list of information about the shape is given. The diagram shows the following information:

- **Internal Name.** The friendly name of the shape. This is name is used for the name of the command bar button (and the tooltip for it) and for the default value of the Name property in VBA.

- **Shaped Concentric Fill.** Specifies whether or not this shape will use a shaped concentric fill (i.e. the shape is rendered at various different sizes to create a concentric gradient fill). If not, the shape will use a rectangular concentric fill.

- **Text rectangle.** The dark gray area shows the rectangular region in which the attached textbox is placed (this region isn’t really filled differently; it is just shown in dark grey for this diagram).

- **Connection sites.** The blue Xs show the locations at which a connector can attach to this shape. The connection sites adjust with the shape as the adjust handles are used. Connection sites are numbered counterclockwise starting from the top or left side.

- **Adjust handles.** The yellow diamonds show the handles which you can drag to adjust the shape. Each adjust handle can control either one adjust value (horizontal or vertical) or two adjust values (both horizontal and vertical). The adjust handles are shown in their default location for new shapes.

- **Resize handles.** The white squares show the handles which you can drag to resize the shape. The resize handles are shown around the “geometry box” of the shape (when you create a shape by dragging out, you are specifying the position and size of the geometry box). Most shapes are contained entirely within their geometry box, but some extend past it. Most shapes extend to each edge of their geometry box, but some do not.
This values corresponds to the fFillShadeShapeOK variable in the shape’s geometry properties.

**Joins**
Specifies what join style the shape has. Since there is no UI for changing the join style, all shapes of this type will always have the specified join style.

This values corresponds to the lineJoinStyle variable in the shape’s line style properties.

**Endcaps**
Specifies what endcap style the shape has. Since there is no UI for changing the endcap style (except for the implicit change in endcap style caused by changing the dash style), all shapes of this type will always have the specified endcap style (except when certain dash styles are used).

This values corresponds to the lineEndCapStyle variable in the shape’s line style properties.

**Adjustments**
Describes the behavior of the adjustments including a brief description of what the adjust handles do, the range of the adjust handles, and whether or not the shape has limo-stretch behavior.

**Path**
This string describes a sequence of commands that define the shape’s path. This string describes both the pSegmentInfo array and pVertices array in the shape’s geometry properties.

The following rules apply to path strings:
- Commas or spaces delimit parameters for each command. Both "m 0,0" and "m0 0" are acceptable.
- A parameter that is omitted using commas is treated as having a value of zero. Thus, "c 10,10,0,0,25,13" and "c 10,10,,25,13" are equivalent.
- Parameterized paths are also allowed. In this case, the shape must also have a list of guide formulas that are substituted into the path using the @ symbol followed by the number of the formula. The adj property of the shape contains the input parameters for these formulas. For example, "moveto @1@4". The evaluations of the formulas are substituted into the appropriate positions. Note that @ also serves as a delimiter.

The allowed commands are given below. An asterisk (*) indicates that the command is allowed to be repeated. For the qb command, the controlpoint parameter is also allowed to be repeated.

<table>
<thead>
<tr>
<th>Command</th>
<th>Name</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>moveto</td>
<td>2</td>
<td>Start a new sub-path at the given (x,y) coordinate.</td>
</tr>
<tr>
<td>Command</td>
<td>Name</td>
<td>Parameters</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>lineto</td>
<td>2*</td>
<td>Draw a line from the current point to the given (x,y) coordinate which becomes the new current point. Specifying a number of coordinate pairs forms a polyline.</td>
</tr>
<tr>
<td>c</td>
<td>curveto</td>
<td>6*</td>
<td>Draw a cubic bézier curve from the current point to the coordinate given by the final two parameters. The control points are given by the first four parameters.</td>
</tr>
<tr>
<td>x</td>
<td>close</td>
<td>0</td>
<td>Close the current sub-path by drawing a straight line from the current point to the original moveto point.</td>
</tr>
<tr>
<td>e</td>
<td>end</td>
<td>0</td>
<td>End the current set of sub-paths. A given set of sub-paths (as delimited by end) is filled. Subsequent sets of sub-paths are filled independently and superimposed on existing ones.</td>
</tr>
<tr>
<td>t</td>
<td>rmoveto</td>
<td>2*</td>
<td>Start a new sub-path at a coordinate relative to the current point, cp (cpx+x, cpy+y).</td>
</tr>
<tr>
<td>r</td>
<td>rlineto</td>
<td>2*</td>
<td>Draw a line from the current point to the given relative coordinate (cpx+x, cpy+y).</td>
</tr>
<tr>
<td>v</td>
<td>rcurveto</td>
<td>6*</td>
<td>Cubic bézier curve using the given coordinate relative to the current point.</td>
</tr>
<tr>
<td>nf</td>
<td>nofill</td>
<td>0</td>
<td>The current set of sub-paths (delimited by e) will not be filled.</td>
</tr>
<tr>
<td>ns</td>
<td>nostroke</td>
<td>0</td>
<td>The current set of sub-paths (delimited by e) will not be stroked.</td>
</tr>
<tr>
<td>ae</td>
<td>angleellipseto</td>
<td>6*</td>
<td>Draws a segment of an ellipse as described using these parameters. A straight line is drawn from the current point to the start point of the segment. The parameters are: center (x,y), size(w,h), start angle, end angle.</td>
</tr>
<tr>
<td>al</td>
<td>angleellipse</td>
<td>6*</td>
<td>Same as angleellipseto except that there is an implied moveto the starting point of the segment.</td>
</tr>
<tr>
<td>Command</td>
<td>Name</td>
<td>Parameters</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>at</td>
<td>arcto</td>
<td>8*</td>
<td>A segment of the ellipse is drawn which starts at the angle defined by the start radius vector and ends at the angle defined by the end vector. A straight line is drawn from the current point to the start of the arc. The arc is always drawn in a counterclockwise direction. The parameters are: left, top, right, bottom, start(x,y), end(x,y). The first four values define the bounding box of an ellipse. The last four define two radial vectors.</td>
</tr>
<tr>
<td>ar</td>
<td>arc</td>
<td>8*</td>
<td>Same as arcto except there is an implied moveto the start point of the arc.</td>
</tr>
<tr>
<td>wa</td>
<td>clockwisearcto</td>
<td>8*</td>
<td>Same as arcto but the arc is drawn in a clockwise direction.</td>
</tr>
<tr>
<td>wr</td>
<td>clockwisearc</td>
<td>8*</td>
<td>Same as arc but the arc is drawn in a clockwise direction.</td>
</tr>
<tr>
<td>qx</td>
<td>ellipticalquadrantx</td>
<td>2*</td>
<td>A quarter ellipse is drawn from the current point to the given end point. The elliptical segment is initially tangential to a line parallel to the x-axis. (i.e. the segment starts out horizontal). The parameters are: end(x,y).</td>
</tr>
<tr>
<td>qy</td>
<td>ellipticalquadrant</td>
<td>2*</td>
<td>Same as ellipticalquadrantx except that the elliptical segment is initially tangential to a line parallel to the y-axis (i.e. the segment starts out vertical).</td>
</tr>
<tr>
<td>qb</td>
<td>quadraticbezier</td>
<td>2+2*</td>
<td>Defines one or more quadratic bézier curves by means of control points and an end point. Intermediate (on-curve) points are obtained by interpolation between successive control points as in the OpenType font specification. The sub-path need not be started in which case the sub-path will be closed. In this case the last point of the sub-path defines the start point of the quadratic bézier. The parameters are: controlpoint(x,y)*, end(x,y).</td>
</tr>
</tbody>
</table>
**Guide Formulas**

This specifies a list of formulas whose calculated values are referenced by other properties. Each formula is listed on a separate line. Formulas are ordered, with the first formula having index 0. This section can be omitted if the shape doesn’t need any guides.

The result of each calculation is referenced using @ followed by a number corresponding to the zero-based index for that formula in the list of formulas.

The list of formulas corresponds to the pGuides array in the shape’s geometry properties. Individual formulas correspond to an instance of the SG (Shape Guide) structure.

The following list describes the set of possible formulas:

- # followed by an integer specifies the adjustment value with the given zero-based index in the list of adjustment values.
- @ followed by an integer specifies another guide formula in the list by its zero-based index in the list.
- var is the name of a variable. Variables can be of the form xn, yn, or tn where n is an integer between 0 and 127.
- const is a constant. Constants can be integers between 0 and 65535 (unsigned 16-bit integers) or of the form SDn or SDnTm where n and m are integers. SDn = S/n and SDnTm = (S/n) * m.
- val is a value. Values can be constants or variables:
  - Constants are described above.
  - Variables can be any of the variable names that are defined in the guides section. For values in the guides section, you can only reference variables that have been defined earlier in the section. If there is no guides section, you cannot use variables as values.
  - In the guides section only, val can also be one of the following:
    - Adjust values (of the form adjustn where n is an integer between 1 and 8). Must be ones that are defined in the adjust handles section.
    - width or height. Variables that represent the width and height of the shape in shape coordinates. For non-limo-stretch shapes, the constant S can be used instead, but for limo-stretch these variables may be needed.
    - func is a function that can be used in defining the value of a variable in the guides section. The full list of functions that can be used is given in the following table:

<table>
<thead>
<tr>
<th>Function</th>
<th>Name</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add and Subtract</td>
<td>sum</td>
<td>a + b - c</td>
</tr>
<tr>
<td>Multiply and Divide</td>
<td>product</td>
<td>a * b / c</td>
</tr>
<tr>
<td>Midpoint</td>
<td>mid</td>
<td>(a + b) / 2</td>
</tr>
<tr>
<td>Absolute Value</td>
<td>absolute</td>
<td>abs(a)</td>
</tr>
<tr>
<td>Minimum</td>
<td>min</td>
<td>min(a,b)</td>
</tr>
<tr>
<td>Maximum</td>
<td>max</td>
<td>max(a,b)</td>
</tr>
<tr>
<td>If</td>
<td>if</td>
<td>a &gt; 0 ? b : c</td>
</tr>
<tr>
<td>Square Root</td>
<td>sqrt</td>
<td>sqrt(a)</td>
</tr>
<tr>
<td>Modulus</td>
<td>mod</td>
<td>sqrt(a^2 + b^2 + c^2)</td>
</tr>
<tr>
<td>Sine</td>
<td>sin</td>
<td>a * sin(b)</td>
</tr>
<tr>
<td>Cosine</td>
<td>cos</td>
<td>a * cos(b)</td>
</tr>
<tr>
<td>Tangent</td>
<td>tan</td>
<td>a * tan(b)</td>
</tr>
<tr>
<td>ArcTangent</td>
<td>atan2</td>
<td>atan2(b,a)</td>
</tr>
<tr>
<td>Sine of ArcTangent</td>
<td>sinatan2</td>
<td>a * sin(atan2(c,b))</td>
</tr>
<tr>
<td>Cosine of ArcTangent</td>
<td>cosatan2</td>
<td>a * cos(atan2(c,b))</td>
</tr>
</tbody>
</table>
### Angle Add and Subtract

<table>
<thead>
<tr>
<th>sumangle</th>
<th>( a + b\degree - c\degree )</th>
</tr>
</thead>
</table>

### Ellipse Intersection

| ellipse | \( c \times \sqrt{1-(a/b)^2} \) |

### Adjustment Values

Specifies a comma-delimited list of parameters, or adjustment values, used to define values for a parameterized formula. These values represent the location of an adjust handle and may be referenced by the geometry of an adjust handle or as a parameter guide function.

Values may be omitted. Each value is referenced using # followed by a number corresponding to the zero-based index for that value in the list of adjustment values.

These values correspond to adjustValue, adjust2Value, adjust3Value, adjust4Value, adjust5Value, adjust6Value, adjust7Value, adjust8Value, adjust9Value, and adjust10Value in the shape’s geometry properties.

### Connector Locations

These values specify the location of connection points on the shape’s path. The connection points are defined by a string consisting of pairs of x and y values, delimited by commas.

The value “Rectangle” specifies that the shape has four default connection sites; one at each midpoint of the shape’s top, bottom, left, and right sides.

These values correspond to the pConnectionSites array in the shape’s geometry properties.

### Connector Angles

Associated with each connection site, there is a direction which specifies at what angle elbow and curved connectors should attach to it:

These are listed in the same order as the connection sites that they correspond to. Values are specified in degrees and separated by commas. If this property is omitted, the direction...
for each site is computed automatically as the angle from the geometric center of the shape to the connection site.

These values correspond to the pConnectionSitesDir array in the shape’s geometry properties.

**Text Box Rectangle**

Specifies one or more text boxes inscribed inside the shape. A textbox is defined by one or more sets of numbers specifying (in order) the left, top, right, and bottom points of the rectangle. Multiple sets are delimited by a semicolon. If omitted, the text box is the same as the geometry’s bounding box.

For normal shapes, the same text rectangle is used for both horizontal and vertical text; for shapes with two text rectangles, the first text rectangle is used for horizontal text and the second is used for vertical text.

These values correspond to entries in the pInscribe array in the shape’s geometry properties.

**Handles**

This section specifies the properties of each adjust handle on the shape. One adjust handle is specified per line. The properties for each handle correspond to values of the ADJH structure contained in the pAdjustHandles array in the shape’s geometry properties. The following properties are specified if different than their default values:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
</table>
| position  | Specifies the x and y position of the handle. If the polar attribute is present, defines the handle position using radius and angle values. Default is “0,0”.

Each values is one of the following:
- constant
- formula (e.g., @2)
- adjustment value (e.g., #2)
- center
- topleft
- bottomright

Each of the above except for an adj value reference fixes the handle position for that dimension. Specifying an adjustment value allows the handle to move in that dimension and the handle position for that dimension is stored in the adjustment value.

These values correspond to the apX and apY variables in the ADJH structure.

polar | Specifies the center position of a handle that uses polar coordinates. If specified, the position attribute is assumed to contain radius and angle values. If omitted, the position attribute is assumed to contain x and y positions. Default is 0,0.

These values correspond to the xRange and yRange variables in the ADJH structure.
| **xrange** | Specifies a range of minimum and maximum values that constrain the x position of a handle. Default is "0,0". Each value is either a constant or a formula reference. Omitting a value leaves that bound unconstrained.

These values correspond to the xMin and xMax variables in the ADJH structure. |
| --- | --- |
| **yrange** | Specifies a range of minimum and maximum values that constrain the y position of a handle. Default is "0,0". Each value is either a constant or a formula reference. Omitting a value leaves that bound unconstrained.

These values correspond to the yMin and yMax variables in the ADJH structure. |
| **radiusrange** | Specifies a range of minimum and maximum values that constrain the radius of a handle using polar coordinates. Default is "0,0". Each value is either a constant or a formula reference. Omitting a value leaves that bound unconstrained.

These values correspond to the xMin and xMax variables in the ADJH structure, when the adjust handle type is PolarPin. |
| **switch** | Specifies whether the x and y dimensions of the handle are switched when the shape is taller than it is wide. Default is false. This is useful for shapes with limo stretch behavior.

If true, the SwitchPosition property is added to the adjust handle type, specified by the variable f in the ADJH structure. For example, Pin becomes Pin|SwitchPosition. |

The above values also determine the adjust handle type, specified by the f variable in the ADJH structure. By default, the type is Pin. If the polar attribute but the radiusrange attribute is not present, the type is Polar. If the polar attribute and radiusrange attributes are present, the type is PolarPin.

**Limo**

Specifies the (x,y) coordinates of the limo stretch point. Some shapes that have portions that should be constrained to a fixed aspect ratio, are designed with limo-stretch to keep those portions at the fixed aspect ratio. If omitted, the shape has no limo stretch point.

This value corresponds to the xLimo and yLimo variables in the shape’s geometry properties.
Description of Shape Properties

### Basic Shapes

**Rectangle**

*Internal Name:* Rect  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m.1,21600r21600,21600,xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>Rectangle</td>
</tr>
</tbody>
</table>

**Parallelogram**

*Internal Name:* Parallelogram  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* A top perimeter adjust handle controls the amount of skew.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,1,21600@1,21600,21600,xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @0</td>
</tr>
<tr>
<td></td>
<td>prod #0 1 2</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @2</td>
</tr>
<tr>
<td></td>
<td>mid #0 @0</td>
</tr>
<tr>
<td></td>
<td>mid @1 0</td>
</tr>
<tr>
<td></td>
<td>prod height width #0</td>
</tr>
<tr>
<td></td>
<td>prod @6 1 2</td>
</tr>
<tr>
<td></td>
<td>sum height 0 @7</td>
</tr>
<tr>
<td></td>
<td>prod width 1 2</td>
</tr>
<tr>
<td></td>
<td>sum #0 0 @9</td>
</tr>
<tr>
<td></td>
<td>if @10 @8 0</td>
</tr>
<tr>
<td></td>
<td>if @10 @7 height</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>5400</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>@4,0;10800;0;11;@3,10800;0;5,21600;10800;@12;@2,10800</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>1800,1800,19800,19800;8100,8100,13500,13500;10800,10800,10800,10800</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;:0,topLeft&quot; xrange=&quot;:0,21600&quot;</td>
</tr>
</tbody>
</table>
Trapezoid

*Internal Name:* Trapezoid
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* A bottom perimeter adjust handle controls the length of the bottom edge. The adjust handle can extend halfway across.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m.0@0,21600@1,21600,21600.xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td></td>
</tr>
<tr>
<td></td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @0</td>
</tr>
<tr>
<td></td>
<td>prod #0 1 2</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @2</td>
</tr>
<tr>
<td></td>
<td>mid #0 width</td>
</tr>
<tr>
<td></td>
<td>mid @ 1 0</td>
</tr>
<tr>
<td></td>
<td>prod height width #0</td>
</tr>
<tr>
<td></td>
<td>prod @6 1 2</td>
</tr>
<tr>
<td></td>
<td>sum height 0 @7</td>
</tr>
<tr>
<td></td>
<td>prod width 1 2</td>
</tr>
<tr>
<td></td>
<td>sum #0 0 @9</td>
</tr>
<tr>
<td></td>
<td>if @10 @8 0</td>
</tr>
<tr>
<td></td>
<td>if @10 @7 height</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>5400</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>@3,10800;10800,21600;@2,10800;10800,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>1800,1800,19800,19800;4500,4500,17100,17100;7200,7200,14400,14400</td>
</tr>
<tr>
<td>Handles</td>
<td>position=”@0, bottomRight”, x-range=”0,10800”</td>
</tr>
</tbody>
</table>

Diamond

*Internal Name:* Diamond
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m10800.1,10800,10800,21600,21600,10800xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>Rectangle</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>5400,5400,16200,16200</td>
</tr>
</tbody>
</table>

Rounded Rectangle

*Internal Name:* RRect
*Shaped Concentric Fill:* Yes.
*Joins:* Rounded.
*Endcaps:* Rounded.
*Adjustments:* A top or left perimeter adjust handle controls the roundedness of the corners. It limo-stretches both vertically and horizontally at the midpoint, so that the rounded corners are always circular. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend halfway across.

**Geometric properties:**

The rounded rectangle shape is a geometric primitive. It uses default rectangle connection points and its one adjust handle adjusts the radius of the corners.
Octagon

*Internal Name:* Octagon
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.

*Adjustments:* A top or left perimeter adjust handle controls the amount cropped off of the corners. It stretches both vertically and horizontally at the midpoint, so that the corners are always at 45°. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension.

*Geometric properties:*

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,l0@0,0#0@0,21600@1,21600,21600@0@1,xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>sum width 0 #0</td>
</tr>
<tr>
<td></td>
<td>sum height 0 #0</td>
</tr>
<tr>
<td></td>
<td>prod @0 2929 10000</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @3</td>
</tr>
<tr>
<td></td>
<td>sum height 0 @3</td>
</tr>
<tr>
<td></td>
<td>val width</td>
</tr>
<tr>
<td></td>
<td>val height</td>
</tr>
<tr>
<td></td>
<td>prod width 1 2</td>
</tr>
<tr>
<td></td>
<td>prod height 1 2</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>6326</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>@8.0,0,0;@9,8,0;@6,6,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>0,21600,21600,2700,2700,18900,18900,5400,5400,16200,16200</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,topLeft&quot; switch=&quot;true&quot; xrange=&quot;0,10800&quot;</td>
</tr>
<tr>
<td>Limo</td>
<td>10800,10800</td>
</tr>
</tbody>
</table>

Isosceles Triangle

*Internal Name:* IsoscelesTriangle
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.

*Adjustments:* A top perimeter adjust handle controls the location of the top vertex.

*Geometric properties:*

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,l21600r21600,xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>prod #0 1 2</td>
</tr>
<tr>
<td></td>
<td>sum @1 10800 0</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>10800</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>@0,0;@1,10800;0,21600;10800,21600;21600,21600;@2,10800</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>0,10800,10800,10800,18000,18000,5400,10800,16200,18000,10800,10800,21600,18000,0,7200,7200,21600;7200,7200,14400,21600;14400,7200,21600,21600</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,topLeft&quot; xrange=&quot;0,21600&quot;</td>
</tr>
</tbody>
</table>

Right Triangle

*Internal Name:* RightTriangle
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.

*Adjustments:* None.

*Geometric properties:*

<table>
<thead>
<tr>
<th>Path</th>
<th>m,l21600r21600,xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>0,0,0,10800;0,21600,10800,21600;10800,10800,21600;21600,10800,10800,10800,10800,21600,18000</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>1800,12600,12600,19800</td>
</tr>
</tbody>
</table>

Oval

*Internal Name:* Oval
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.

*Adjustments:* None.

*Geometric properties:*

The oval is a geometric primitive with default values.
### Hexagon

*Internal Name:* Hexagon  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* A top perimeter adjust handle controls the length of the pointed ends on both sides. The adjust handle can extend halfway across.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,1,10800@0,0,21600@1,21600,21600,10800@1,xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
sum width 0 #0  
sum height 0 #0  
prod @0 2929 10000  
sum width 0 @3  
sum height 0 @3 |
| Adjustment Values | 5400 |
| Connector Locations | Rectangle |
| Text Box Rectangle | 1800,1800,19800,19800,3600,3600,18000,18000,6300,6300,15300,15300 |
| Handles | position=’#0,topLeft’ xrange=’0,10800’ |

### Cross

*Internal Name:* Plus  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* A top or left perimeter adjust handle controls the amount cut out of the corners. It limo-stretches both vertically and horizontally at the midpoint, so that the amount cut out of the corners is always square. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,1,0,0@0,0@0,0@0,2@0@2@0,21600@1,21600@1,21600@2,21600,10800@1,0@1@0,1,xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
sum width 0 #0  
sum height 0 #0  
prod @0 2929 10000  
sum width 0 @3  
sum height 0 @3  
val width  
val height  
prod width 1 2  
prod height 1 2 |
| Adjustment Values | 5400 |
| Connector Locations | @8,0,0@9,9@8,7@6,6,6,9 |
| Text Box Rectangle | 0,0,21600,21600,5400,5400,16200,16200,10800,10800,10800,10800 |
| Handles | position=’#0,topLeft’ switch=’true’ xrange=’0,10800’ |
| Limo | 10800,10800 |

### Regular Pentagon

*Internal Name:* Pentagon  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m10800.1,8259,4200,21600,13200,121600,8259xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>10800,0,0,8259,4200,21600,10800,21600,17400,21600,8259</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270,180,90,90,90,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>4200,5077,17400,21600</td>
</tr>
</tbody>
</table>
### Can

**Internal Name:** Can  
**Shaped Concentric Fill:** Yes  
**Joins:** Rounded  
**Endcaps:** Rounded

**Adjustments:** One vertical adjust handle along the center adjusts the tilt of the can. The adjust handle can extend halfway down.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m10800,qx0@1l0@2qy10800,21600,21600@2l21600@1qy10800,xem0@1qy10800@0,21600@1l2fe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
prod #0 1 2  
sum height 0 @1 |
| Adjustment Values | 5400 |
| Connector Locations | 10800,0,10800,0,0,10800,10800,21600,21600,10800 |
| Connector Angles | 270,270,180,90,0 |
| Text Box Rectangle | 0@0,21600,0@2 |
| Handles | position="center,#0",yrange="0,10800" |

### Cube

**Internal Name:** Cube  
**Shaped Concentric Fill:** Yes  
**Joins:** Mitered  
**Endcaps:** Rounded

**Adjustments:** A top or left perimeter adjust handle controls the amount of extrusion. It limo-stretches both vertically and horizontally at the midpoint, so that the extrusion is always at 45°. The adjust handle switches between the top and left sides depending on which dimension is smaller.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,l0@0,,21600@1,21600,21600@2,21600,xem0@0nfl@1@0,21600,em@1@0nfl@1,21600e</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
sum width 0 #0  
sum height 0 #0  
mid height #0  
prod @1 1 2  
prod @2 1 2  
mid width #0 |
| Adjustment Values | 5400 |
| Connector Locations | @6,0;@4,0;@3;4,21600;@1,0;3,21600,5 |
| Connector Angles | 270,270,180,90,0,0 |
| Text Box Rectangle | 0@0,21600 |
| Handles | position="topLeft,#0",switch="true",yrange="0,21600" |
| Limo | 10800,10800 |
### Bevel

*Internal Name:* Bevel  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension.

#### Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m,L21600r21600,l21600,xem@0@0nfl@0@2@2@1@0xem,nfl@0@0em ,21600nfl@0@2em21600,21600nfl@1@2em21600,nfl@1@0e</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
sum width 0 #0  
sum height 0 #0  
prod width 1 2  
prod height 1 2  
prod #0 1 2  
prod #0 3 2  
sum @1 @5 0  
sum @2 @5 0 |
| Adjustment Values | 2700 |
| Connector Locations | 0,@4;@0,@4;@3,21600;@3,@2;21600,0;@1,@4;@3,0;@3,00 |
| Text Box Rectangle | @0,@0,@1,@2 |
| Handles | position=”#0,topLeft” switch=”true” xrange="0,10800" |
| Limo | 10800,10800 |

### Folded Corner

*Internal Name:* FoldedCorner  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  

**Adjustments:** A bottom perimeter adjust handle controls the amount that the corner is folded.

#### Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m,L21600@0,21600,21600@0,21600,xem@0,21600nfl@3@5@7@9@11@13,21600@0e</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
sum 21600 0 #0  
prod @1 8481 32768  
sum @2 @0 0  
prod @1 1117 32768  
sum @4 @0 0  
prod @1 11764 32768  
sum @6 @0 0  
prod @1 6144 32768  
sum @8 @0 0  
prod @1 20480 32768  
sum @10 @0 0  
prod @1 6144 32768  
sum @12 @0 0 |
| Adjustment Values | 18900 |
| Connector Locations | Rectangle |
| Text Box Rectangle | 0,21600@13 |
| Handles | position=”#0,bottomRight” xrange="10800,21600" |
### Smiley Face

**Internal Name:** SmileyFace  
**Shaped Concentric Fill:** Yes.  
**Joins:** Rounded.  
**Endcaps:** Rounded.  
**Adjustments:** A vertical adjust handle along the center adjust the curvature of the mouth changing it from a smile to a frown.

**Geometric properties:**

| Path | m10800.qx,10800,10800,21600,10800,10800,21600,10800,10800,10800,21600,10800,10800,xem7340,6445qx6215,7570,7340,8695,8465,7570,7340,6445xnfem14260,6445qx13135,7570,14260,8695,15385,7570,14260,6445xnfem9650@0c8853@3,12747@3,16640@0nfem |
| Guide Formulas | sum \(33030\) 0 \(\#0\)  
prod \(\#0\) 4 3  
prod \(\#0\) 1 3  
sum \(\@1\) 0 \(\@2\)  
| Adjustment Values | 17520 |
| Connector Locations | 10800,0;3163,3163;0,10800;3163,18437;10800,21600;18437,18437;21600,10800;18437,3163 |
| Text Box Rectangle | 3163,3163,18437,18437 |
| Handles | position="center,\#0" xrange="15510,17520" |

### Donut

**Internal Name:** Donut  
**Shaped Concentric Fill:** No.  
**Joins:** Rounded.  
**Endcaps:** Rounded.  
**Adjustments:** One horizontal adjust handle along the center adjusts the inner radius of the ring. The adjust handle can extend halfway across.

**Geometric properties:**

| Path | m.10800.qy,10800,21600,10800,10800,21600,10800,10800,xm@0,10800.qy,10800@2 @1,10800,10800@0@0,10800xe |
| Guide Formulas | val \#0  
sum width 0 \#0  
sum height 0 \#0  
prod \(\#0\) 2929 10000  
sum width 0 \@3  
sum height 0 \@3  |
| Adjustment Values | 5400 |
| Connector Locations | 10800,0;3163,3163;0,10800;3163,18437;10800,21600;18437,18437;21600,10800;18437,3163 |
| Text Box Rectangle | 3163,3163,18437,18437 |
| Handles | position="#0,center" xrange="0,10800" |
**“No” Symbol**

*Internal Name:* NoSmoking  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* One horizontal adjust handle along the center adjusts the inner radius of the ring and thickness of the slash. The adjust handle can extend 1/3 of the way across.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m,10800qy10800,,21600,10800,10800,21600,,10800xar@0@0@16@16@12 @14@15@13xar@0@0@16@16@13@15@14@12xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
| | prod @0 2 1  
| | sum 21600 0 @1  
| | prod @2 2 1  
| | prod @0 0 1  
| | sum @3 0 @4  
| | prod @5 1 8  
| | sqrt @6  
| | prod @4 1 8  
| | sqrt @8  
| | sum @7 9 0  
| | sum @7 0 9  
| | sum @10 10800 0  
| | sum 10800 0 @10  
| | sum @11 10800 0  
| | sum 10800 0 @11  
| | sum 21600 0 @0  |

| Adjustment Values | 2700 |
| Connector Locations | 10800,0;3163,3163;0,10800;3163,18437;10800,21600;18437,18437;21600,10800;18437,3163 |
| Text Box Rectangle | 3163,3163,18437,18437 |
| Handles | position=“#0,center” xrange=“0,7200” |
**Block Arc**

*Internal Name:* BlockArc  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  

**Adjustments:** A polar adjust handle controls the starting angle and inner radius. The shape goes clockwise starting from the starting angle and is symmetrical about the vertical center.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>a10800,10800@0@0@2@14,10800,10800,10800,10800@3@15xe</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Guide Formulas</strong></td>
<td></td>
</tr>
<tr>
<td>val #1</td>
<td></td>
</tr>
<tr>
<td>val #0</td>
<td></td>
</tr>
<tr>
<td>sum 0 0 #0</td>
<td></td>
</tr>
<tr>
<td>sumangle #0 0 180</td>
<td></td>
</tr>
<tr>
<td>sumangle #0 0 90</td>
<td></td>
</tr>
<tr>
<td>prod @4 2 1</td>
<td></td>
</tr>
<tr>
<td>sumangle #0 90 0</td>
<td></td>
</tr>
<tr>
<td>prod @6 2 1</td>
<td></td>
</tr>
<tr>
<td>abs #0</td>
<td></td>
</tr>
<tr>
<td>sumangle @8 0 90</td>
<td></td>
</tr>
<tr>
<td>if @9 @7 @5</td>
<td></td>
</tr>
<tr>
<td>sumangle @10 0 360</td>
<td></td>
</tr>
<tr>
<td>if @10 @11 @10</td>
<td></td>
</tr>
<tr>
<td>sumangle @12 0 360</td>
<td></td>
</tr>
<tr>
<td>if @12 @13 @12</td>
<td></td>
</tr>
<tr>
<td>sum 0 0 @14</td>
<td></td>
</tr>
<tr>
<td>val 10800</td>
<td></td>
</tr>
<tr>
<td>sum 10800 0 #1</td>
<td></td>
</tr>
<tr>
<td>prod #1 1 2</td>
<td></td>
</tr>
<tr>
<td>sum @18 5400 0</td>
<td></td>
</tr>
<tr>
<td>cos @19 #0</td>
<td></td>
</tr>
<tr>
<td>sin @19 #0</td>
<td></td>
</tr>
<tr>
<td>sum @20 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @21 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum 10800 0 @20</td>
<td></td>
</tr>
<tr>
<td>sum #1 10800 0</td>
<td></td>
</tr>
<tr>
<td>if @9 @17 @25</td>
<td></td>
</tr>
<tr>
<td>if @9 0 21600</td>
<td></td>
</tr>
<tr>
<td>cos 10800 #0</td>
<td></td>
</tr>
<tr>
<td>sin 10800 #0</td>
<td></td>
</tr>
<tr>
<td>sin @28 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @29 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @30 10800 0</td>
<td></td>
</tr>
<tr>
<td>if @4 0 @31</td>
<td></td>
</tr>
<tr>
<td>if #0 @34 @0</td>
<td></td>
</tr>
<tr>
<td>if @6 @35 @31</td>
<td></td>
</tr>
<tr>
<td>sum 21600 0 @36</td>
<td></td>
</tr>
<tr>
<td>if @4 0 @33</td>
<td></td>
</tr>
<tr>
<td>if @6 @38 @32</td>
<td></td>
</tr>
<tr>
<td>if @6 @39 @0</td>
<td></td>
</tr>
<tr>
<td>if @4 @32 21600</td>
<td></td>
</tr>
<tr>
<td>if @6 @41 @33</td>
<td></td>
</tr>
</tbody>
</table>

**Adjustment Values:** 11796480,5400  
**Connector Locations:** 10800;@27;@22;@23;10800;@26;@24;@23  
**Text Box Rectangle:** @36;@40;@37;@42  
**Handles:** position="#1,#0" polar="10800,10800" radiusrange="0,10800"
Heart
*Internal Name:* Heart
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* None.

**Geometric properties:**

| Path | m10860,2187c10451,1746,9529,1018,9015,730,7865,152,6685,4175,5415,2187,1967,1305,1150,2187,575,3222,242,4220,5410,242,6560,575,759 710860,21600,20995,7597,485,-1037,605,-2187,485,-3377c21115,3222,2042 0,2187,19632,1305,18575,575,17425,152,16275,15005,13735,152,12705,73 0v-529,298,-1451,1016,-1845,1457xe |
| Connector Locations | 10860,2187;2928,10800;10860,21600;18672,10800 |
| Connector Angles | 270,180,90,0 |
| Text Box Rectangle | 5037,2277,16557,13677 |

Lightning Bolt
*Internal Name:* LightningBolt
*Shaped Concentric Fill:* No.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* None. Note that the text rectangle extends past the interior of the shape.

**Geometric properties:**

<p>| Path | m8472,13890,7602,8382,5022,9705r7200,4192110012,14915r11588,66851147 67,12877r1810,-87011050,6797r1810,-717xe |
| Connector Locations | 8472,0,3890,5022,9705,10012,14915,21600,16577,12007;12860,6080 |
| Connector Angles | 270,270,180,180,90,0,0 |
| Text Box Rectangle | 8757,7437,13917,14277 |</p>
<table>
<thead>
<tr>
<th><strong>Sun</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal Name:</strong> Sun</td>
<td></td>
</tr>
<tr>
<td><strong>Shaped Concentric Fill:</strong> No.</td>
<td></td>
</tr>
<tr>
<td><strong>Joins:</strong> Mitered.</td>
<td></td>
</tr>
<tr>
<td><strong>Endcaps:</strong> Rounded.</td>
<td></td>
</tr>
<tr>
<td><strong>Adjustments:</strong> A horizontal adjust handle along the center controls the radius of the core part of the sun.</td>
<td></td>
</tr>
</tbody>
</table>

**Geometric properties:**

| Path | m21600,10800l@15@14@15@18xem18436,3163l@17@12@16@13xem10800l@14@10@18xem3163,3163l@12@13@12xem10800l@11@10@14xem3163,18436l@13@16@12@17xem10800,21600l@18@15@14@15xem18436,18436l@16@17@17@16xem10800l@19qx@19,10800,10800l@20@20,10800,10800l@19xe |
| Guide Formulas | sum 10800 0 #0 prod @0 30274 32768 prod @0 12540 32768 sum @1 10800 0 sum @2 10800 0 sum 10800 0 @1 sum 10800 0 @2 prod @0 23170 32768 sum @7 10800 0 sum 10800 0 @7 prod @5 3 4 prod @6 3 4 sum @10 791 0 sum @11 791 0 sum @11 2700 0 sum 21600 0 @10 sum 21600 0 @12 sum 21600 0 @13 sum 21600 0 @14 val #0 sum 21600 0 #0 |
| Adjustment Values | $5400 |
| Text Box Rectangle | @9.9.8.8 |
| Handles | position="#0,center" xrange="2700,10125" |
Moon

*Internal Name:* Moon
*Shaped Concentric Fill:* No.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* A horizontal adjust handle along the center adjusts the thickness of the crescent.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m21600,qx,10800,21600,21600wa@0@10@6@11,21600,21600,21600,xе</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
sum 21600 0 #0  
prod #0 #0 @1  
prod 21600 21600 @1  
prod @3 2 1  
sum @4 0 @2  
sum @5 0 #0  
prod @5 1 2  
sum @7 0 #0  
prod @8 1 2  
sum 10800 0 @9  
sum @9 10800 0  
prod #0 9598 32768  
sum 21600 0 @12  
ellipse @13 21600 10800  
sum 10800 0 @14  
sum @14 10800 0 |
| Adjustment Values | 10800 |
| Connector Locations | 21600,0;0,10800;21600,21600;@0,10800 |
| Connector Angles | 270,180,90,0 |
| Text Box Rectangle | @12,@15,@0,@16 |
| Handles | position="#0,center" xrange="0,18900" |

Arc

*Internal Name:* Arc
*Shaped Concentric Fill:* Yes (elliptical fill).
*Joins:* Rounded.
*Endcaps:* Rounded.
*Adjustments:* The first adjust handle controls the start angle and the second adjust handle controls the end angle.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>wr-21600,21600,43200,21600,21600nfewr-21600,21600,43200,21600,21600,xе</th>
</tr>
</thead>
</table>
| Guide Formulas | val #2  
val #3  
val #4 |
| Adjustment Values | -5898240,21600,21600 |
| Connector Locations | 0,0;21600,21600;0,21600 |
| Handles | position="@2,#0" polar="@0,@1"  
position="@2,#1" polar="@0,@1" |
Double Bracket

*Internal Name:* BracketPair

*Shaped Concentric Fill:* Yes.

*Joins:* Rounded.

*Endcaps:* Flat.

*Adjustments:* A top or left perimeter adjust handle controls the roundedness of the corners. It limo-stretches both vertically and horizontally at the midpoint, so that the rounded corners are always circular. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend halfway across.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,nfqx0@0l0@<a href="mailto:2qy0@0.21600em">2qy0@0.21600em</a>@1,nfqx21600@0l21600@2qy21600@1,xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
| | sum width 0 #0  
| | sum height 0 #0  
| | prod @0 2929 10000  
| | sum width 0 @3  
| | sum height 0 @3  
| | val width  
| | val height  
| | prod width 1 2  
| | prod height 1 2 |
| Adjustment Values | 3600 |
| Connector Locations | @8,0,0;@9,0,0;@7,0,0;@6,0,0 |
| Text Box Rectangle | @3,0,0;@4,0,0 |
| Handles | position="topLeft" switch="true" xrange="0,10800" |
| Limo | 10800,10800 |

Double Brace

*Internal Name:* BracePair

*Shaped Concentric Fill:* No.

*Joins:* Mitered.

*Endcaps:* Flat.

*Adjustments:* A top or left perimeter adjust handle controls the roundedness of the corners. It limo-stretches both vertically and horizontally at the midpoint, so that the rounded corners are always circular. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend halfway across.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@9,nfqx@0l@0@7qy@0@4@0@8l@0@6qy@9,21600em@10,nfqx5 @0@5@7qy21600@4@5@8l@5@6qy@10,21600em@9,nfqx0@0@0@7qy@4@0@8l@0@6qy@9,21600@10,21600qx@5@6l@5@8qy21600@4 @5@7@5@Oqy@10,xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
| | val width  
| | val height  
| | prod width 1 2  
| | prod height 1 2  
| | sum width 0 #0  
| | sum height 0 #0  
| | sum @4 0 #0  
| | sum @4 #0 0  
| | prod #0 2 1  
| | sum width 0 @9  
| | prod #0 9598 32768  
| | sum height 0 @11  
| | sum @11 #0 0  
| | sum width 0 @13 |
| Adjustment Values | 1800 |
| Connector Locations | @3,0,0;@4,0,0;@3,0,0;@1,0,0 |
| Text Box Rectangle | @13,0,0;@14,0,0 |
| Handles | position="topLeft," switch="true" yrange="0,5400" |
| Limo | 10800,10800 |
Plaque

*Internal Name:* Plaque  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* A top or left perimeter adjust handle controls the roundedness of the corners. It limo-stretches both vertically and horizontally at the midpoint, so that the rounded corners are always circular. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,qy0@0l0@<a href="mailto:2qx@0.21600l">2qx@0.21600l</a>@1,21600qy21600@2l21600@20x@1,xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val 0  
| | sum width 0 @0  
| | sum height 0 @0  
| | prod @0 7071 10000  
| | sum width 0 @3  
| | sum height 0 @3  
| | val width  
| | val height  
| | prod width 1 2  
| | prod height 1 2 |
| Adjustment Values | 3600  
| Connector Locations | @8,0;0,9;@8,8;6,9  
| Text Box Rectangle | 6326,0;21600,10800  
| Handles | position="#0,topLeft" switch="true" xrange="0,10800" |
| Limo | 10800,10800 |

Left Bracket

*Internal Name:* LeftBracket  
*Shaped Concentric Fill:* No.  
*Joins:* Rounded.  
*Endcaps:* Flat.  
*Adjustments:* A left perimeter adjust handle controls the curvature of the corners. The adjust handle can extend halfway down.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m21600,qx0@0l0@1qy21600,21600e</th>
</tr>
</thead>
</table>
| Guide Formulas | val 0  
| | sum 21600 0 @0  
| | prod #0 9598 32768  
| | sum 21600 @2 |
| Adjustment Values | 1800  
| Connector Locations | 21600,0;0,10800;21600,21600  
| Text Box Rectangle | 0,0;2,21600,10800  
| Handles | position="topLeft,#0" yrange="0,10800" |

Right Bracket

*Internal Name:* RightBracket  
*Shaped Concentric Fill:* No.  
*Joins:* Rounded.  
*Endcaps:* Flat.  
*Adjustments:* A right perimeter adjust handle controls the curvature of the corners. The adjust handle can extend halfway down.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m,qx21600@0l21600@1qy,21600e</th>
</tr>
</thead>
</table>
| Guide Formulas | val 0  
| | sum 21600 0 @0  
| | prod #0 9598 32768  
| | sum 21600 @2 |
| Adjustment Values | 1800  
| Connector Locations | 0.0,0.21600;21600,10800  
| Text Box Rectangle | 0,0;2,15274,10800  
| Handles | position="bottomRight,#0" yrange="0,10800" |
Left Brace

*Internal Name:* LeftBrace  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Flat.  

*Adjustments:* A vertical adjust handle along the center controls the curvature of the corners and cusp. The adjust handle can extend ¼ of the way down. A left perimeter adjust handle controls the position of the cusp.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m21600.qx10800@010800@2qy0@11,10800@3110800@1qy21600,21600e</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
| | sum 21600 0 #0  
| | sum #1 0 #0  
| | sum #1 #0 0  
| | prod #0 9598 32768  
| | sum 21600 0 @4  
| | sum 21600 0 #1  
| | min #1 @6  
| | prod @7 1 2  
| | prod #0 2 1  
| | sum 21600 0 @9  
| | val #1 |
| Adjustment Values | 1800,10800 |
| Connector Locations | 21600,0,10800;21600,21600 |
| Text Box Rectangle | 13963,4,21600,5 |
| Handles | position="center,#0" yrange="0,@8  
| | position="topLeft,#1" yrange="@9,10" |

Right Brace

*Internal Name:* RightBrace  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  

*Adjustments:* A vertical adjust handle along the center controls the curvature of the corners and cusp. The adjust handle can extend ¼ of the way down. A right perimeter adjust handle controls the position of the cusp.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m.qx10800@010800@2qy21600@11,10800@3110800@1qy,21600e</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
| | sum 21600 0 #0  
| | sum #1 0 #0  
| | sum #1 #0 0  
| | prod #0 9598 32768  
| | sum 21600 0 @4  
| | sum 21600 0 #1  
| | min #1 @6  
| | prod @7 1 2  
| | prod #0 2 1  
| | sum 21600 0 @9  
| | val #1 |
| Adjustment Values | 1800,10800 |
| Connector Locations | 0,0;21600,11,0,21600 |
| Text Box Rectangle | 0,4,7637,5 |
| Handles | position="center,#0" yrange="0,@8  
| | position="bottomRight,#1" yrange="@9,10" |
Block Arrows

Right Arrow

*Internal Name:* RightArrow
*Shaped Concentric Fill:* No.
*Joins:* Mitered.
*Endcaps:* Flat.
*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. The adjust handle can extend all of the way across and halfway down.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,l@0@1.0@1.0@2@0@2@0.21600,21600,10800xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>val #1</td>
</tr>
<tr>
<td></td>
<td>sum height 0 #1</td>
</tr>
<tr>
<td></td>
<td>sum 10800 0 #1</td>
</tr>
<tr>
<td></td>
<td>sum width 0 #0</td>
</tr>
<tr>
<td></td>
<td>prod @4 @3 10800</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @5</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>16200,5400</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>@0,0,10800;@0,21600;21600,10800</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270,180,90,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>0,1,6,2</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,#1&quot; xrange=&quot;0,21600&quot; yrange=&quot;0,10800&quot;</td>
</tr>
</tbody>
</table>

Left Arrow

*Internal Name:* LeftArrow
*Shaped Concentric Fill:* No.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. The adjust handle can extend all of the way across and halfway down.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,l@0@1.21600@1,21600@2@0@2@0.21600,10800xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>val #1</td>
</tr>
<tr>
<td></td>
<td>sum 21600 0 #1</td>
</tr>
<tr>
<td></td>
<td>prod @4 @3 10800</td>
</tr>
<tr>
<td></td>
<td>sum #0 0 @3</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>5400,5400</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>@0,0,10800;@0,21600;21600,10800</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270,180,90,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@4,1,21600,2</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,#1&quot; xrange=&quot;0,21600&quot; yrange=&quot;0,10800&quot;</td>
</tr>
</tbody>
</table>

Up Arrow

*Internal Name:* UpArrow
*Shaped Concentric Fill:* No.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. The adjust handle can extend halfway across and all of the way down.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,l@1@0@1.21600@2.21600@2@0.21600@0,10800xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>val #1</td>
</tr>
<tr>
<td></td>
<td>sum 21600 0 #1</td>
</tr>
<tr>
<td></td>
<td>prod @4 @3 10800</td>
</tr>
<tr>
<td></td>
<td>sum #0 0 @3</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>5400,5400</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>10800,0,0;@0,10800,21600,21600,0</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270,180,90,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@1,4,2.21600</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#1,#0&quot; xrange=&quot;0,10800&quot; yrange=&quot;0,21600&quot;</td>
</tr>
</tbody>
</table>
**Down Arrow**
*Internal Name:* DownArrow  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. The adjust handle can extend halfway across and all of the way down.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m0@0l@1@0@1,0@2,0@2@0,21600@0,10800,21600xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>val #1</td>
</tr>
<tr>
<td></td>
<td>sum height 0 #1</td>
</tr>
<tr>
<td></td>
<td>sum 10800 0 #1</td>
</tr>
<tr>
<td></td>
<td>sum width 0 #0</td>
</tr>
<tr>
<td></td>
<td>prod @4 @3 10800</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @5</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>16200,5400</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>10800,0,0;0,10800;21600;21600;@0</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270,180,90,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@1,0,@2,@6</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#1,#0&quot;,xrange=&quot;0,10800&quot;,yrange=&quot;0,21600&quot;</td>
</tr>
</tbody>
</table>

**Left-Right Arrow**
*Internal Name:* LeftRightArrow  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. It stays symmetric so that the arrowhead length is the same at both ends. The adjust handle can extend halfway across and halfway down.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m,10800@0@2,1600@0@3@2@3@2,1600@2,1600,10800@2,0@2@1@0@1@0,xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>val #1</td>
</tr>
<tr>
<td></td>
<td>sum 21600 0 #0</td>
</tr>
<tr>
<td></td>
<td>sum 21600 0 #1</td>
</tr>
<tr>
<td></td>
<td>prod #0 #1 10800</td>
</tr>
<tr>
<td></td>
<td>sum #0 0 @4</td>
</tr>
<tr>
<td></td>
<td>sum 21600 0 @5</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>4320,5400</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>@2,0,10800;@1;@0,0,10800;@0,2,1600;10800,3;@2,2,1600;21600,10800</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270,270,180,90,90,90,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@5,1,@6,0,3</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,#1&quot;,xrange=&quot;0,10800&quot;,yrange=&quot;0,10800&quot;</td>
</tr>
</tbody>
</table>
Up-Down Arrow

*Internal Name:* UpDownArrow

*Shaped Concentric Fill:* No.

*Joins:* Mitered.

*Endcaps:* Rounded.

*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. It stays symmetric so that the arrowhead length is the same at both ends. The adjust handle can extend halfway across and halfway down.

*Geometric properties:*

<table>
<thead>
<tr>
<th>Path</th>
<th>m10800,121600@0@3@0@3@2,21600@2,10800,21600,0@2@1@2@1@0,0@0xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #1  
val #0  
sum 21600 0 #1  
sum 21600 0 #0  
prod #1 #0 10800  
sum #1 0 @4  
sum 21600 0 @5 |
| Adjustment Values | 5400,4320 |
| Connector Locations | 10800,0;@0;1,10800;@2,10800,21600,21600,2;0,3,10800,21600,0@0 |
| Connector Angles | 270,180,180,90,0,0 |
| Text Box Rectangle | @1,@5,@3,@6 |
| Handles | position="#0,#1" xrange="0,10800" yrange="0,10800" |

Quad Arrow

*Internal Name:* QuadArrow

*Shaped Concentric Fill:* No.

*Joins:* Mitered.

*Endcaps:* Rounded.

*Adjustments:* The first adjust handle is a top perimeter adjust handle which controls the width of the arrowhead. The second adjust handle controls both the length of the arrowhead, and the width of the trunk. It stays symmetric so that the arrowhead length is the same for all four arrowheads when the shape is scaled to a 1:1 aspect ratio. The first adjust handle can extend as far left as possible before the tips of the arrowheads cross, and as far right as the current position of the second adjust handle. The second adjust handle can extend as far left as the current position of the first adjust handle, as far right as halfway across, and as far down as possible before the arrowheads intersect the trunks. A second text rect is defined for vertical text.

*Geometric properties:*

<table>
<thead>
<tr>
<th>Path</th>
<th>m10800,1@0@0@2@1@2@1@1@2@1@2@1@0,...</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
val #1  
val #2  
sum 21600 0 #0  
sum 21600 0 #1  
sum 21600 0 #2  
sum 0 0 10800  
sum 0 1 10800  
prod @7 #2 @6  
sum 21600 0 @8 |
| Adjustment Values | 6480,8640,4320 |
| Connector Locations | Rectangle |
| Text Box Rectangle | @8,@1,@9,@4,@1,@8,@4,@9 |
| Handles | position="#0,topLeft" xrange="@2,@1"  
position="#1,#2" xrange="@0,10800" yrange="0,0" |
**Left-Right-Up Arrow**

*Internal Name:* LeftRightUpArrow  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.

*Adjustments:* The first adjust handle is a top perimeter adjust handle which controls the width of the arrowhead. The second adjust handle controls both the length of the arrowhead, and the width of the trunk. It stays symmetric so that the arrowhead length is the same for all three arrowheads when the shape is scaled such that the horizontal part of the trunk is the same width as the vertical part of the trunk. The first adjust handle can extend as far left as possible before the tips of the arrowheads cross, and as far right as the current position of the second adjust handle. The second adjust handle can extend as far left as the current position of the first adjust handle, as far right as halfway across, and as far down as possible before the arrowheads intersect the trunks. A second text rect is defined for vertical text.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m10800.0@0@2@1@2@1@6@7@6@7@5.0@8@7.21600@7.9@10@9@10.21600,21600@8@10.5@10.6@6@4@6@2@3@2xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
val #1  
val #2  
sum 21600 0 #0  
sum 21600 0 #1  
prod @0 21600 @3  
prod @1 21600 @3  
prod @2 3 21600  
prod 10800 21600 @3  
prod 4 21600 @3  
sum 21600 0 @7  
sum @5 0 @8  
sum @6 0 @8  
prod @12 @7 @11  
sum 21600 0 @13  
sum @0 0 10800  
sum @1 0 10800  
prod 2 @16 @15 |
| Adjustment Values | 6480.8640,6171 |
| Connector Locations | 10800,0,0,8,10800,9,21600,8 |
| Connector Angles | 270,180,90,0 |
| Text Box Rectangle | @13,6,14,9,1,17,4,9 |
| Handles | position="#0,topLeft" xrange="@2,1"  
positional="#1,#2" xrange="@0,10800" yrange="0,5" |
**Bent Arrow**

*Internal Name:* BentArrow  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. The adjust handle can extend as far left as where the trunk begins to bend and halfway down the arrowhead. A second text rect is defined for vertical text.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m21600,6079l@0,0@0@1,12427@1qx,12158l,21600@4,21600@4,12158qy12427@2l@0@2@0,12158xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
| | val #1  
| | sum 12158 0 #1  
| | prod @2 0 #1  
| | sum @3 32768 32059  
| | prod @4 1 2  
| | sum 21600 0 #0  
| | prod @6 #1 6079  
| | sum @7 #0 0  |
| Adjustment Values | Connector Angles  
| Connector Locations | @0,0;@0,12158;@5,21600;21600,6079  
| Connector Angles | 270,90,90,0  
| Text Box Rectangle | 12427,1,8,2,0,12158,4,21600  
| Handles | position="#0,#1" xrange="12427,21600" yrange="0,6079" |

**U-Turn Arrow**

*Internal Name:* UturnArrow  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. The adjust handle can extend halfway across the arrowhead, as far up as where the trunk begins the bend, and as far down as the tip of the arrowhead.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m15662,14285l21600,8310r-2970,qy9250,,8485l,21600r6110,l6110,8310qy8907,5842r9725,5842q12520,831099725,8310xe</th>
</tr>
</thead>
</table>
| Connector Locations | 9250,0,055,21600,9725,8310,15662,14285,21600,8310  
| Connector Angles | 270,90,90,0  
| Text Box Rectangle | 0.8310,6110,21600  

Left-Up Arrow

Internal Name: LeftUpArrow

Shaped Concentric Fill: No.

Joins: Mitered.
Endcaps: Rounded.

Adjustments: The first adjust handle is a top perimeter adjust handle which controls the width of the arrowhead. The second adjust handle controls both the length of the arrowhead, and the width of the trunk. It stays symmetric so that the arrowhead length is the same for both arrowheads when the shape is scaled to a 1:1 aspect ratio. The first adjust handle can extend as far left as possible before the tips of the arrowheads cross, and as far right as twice the distance from the right perimeter as the current position of the second adjust handle. The second adjust handle can extend as far left as half the distance from the right perimeter as the current position of the first adjust handle, and as far down as possible before the arrowheads intersect the trunks. A second text rect is defined for vertical text.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m@4,1@0@2@5@2@5@2@5@2@0,0@4@2,21600@2@1@1@1@1@1@1@2,21600@2xc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0</td>
</tr>
<tr>
<td></td>
<td>val #1</td>
</tr>
<tr>
<td></td>
<td>val #2</td>
</tr>
<tr>
<td></td>
<td>prod #0 1 2</td>
</tr>
<tr>
<td></td>
<td>sum @3 10800 0</td>
</tr>
<tr>
<td></td>
<td>sum 21600 #0 #1</td>
</tr>
<tr>
<td></td>
<td>sum #1 #2 0</td>
</tr>
<tr>
<td></td>
<td>prod @6 1 2</td>
</tr>
<tr>
<td></td>
<td>prod #1 2 1</td>
</tr>
<tr>
<td></td>
<td>sum @8 0 21600</td>
</tr>
<tr>
<td></td>
<td>sum @5 0 @4</td>
</tr>
<tr>
<td></td>
<td>sum #0 0 @4</td>
</tr>
<tr>
<td></td>
<td>prod @2 @10 @11</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>9257,18514,6171</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>@4,0;@0,0;@2,2;@0,0;@4;@2,21600;@7,1;@1,7;21600,0</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270,180,270,180,90,90,0,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@12,5,0,1,1;@5,0,1,1</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,topLeft&quot; xrange=&quot;@2,0&quot;</td>
</tr>
<tr>
<td></td>
<td>position=&quot;#1,#2&quot; xrange=&quot;@4,21600&quot; yrange=&quot;0,0&quot;</td>
</tr>
</tbody>
</table>
Bent-Up Arrow

*Internal Name:* BentUpArrow

*Shaped Concentric Fill:* No.

*Joins:* Mitered.

*Endcaps:* Rounded.

*Adjustments:* The first adjust handle is a top perimeter adjust handle which controls the width of the arrowhead. The second adjust handle controls both the length of the arrowhead, and the width of the trunk. It stays symmetric so that the lengths of the horizontal and vertical trunks are the same when scaled so that widths of the horizontal and vertical trunks are the same. The first adjust handle can extend as far left as possible before the arrowhead intersects the trunk, and as far right as twice the distance from the right perimeter as the current position of the second adjust handle. The second adjust handle can extend as far left as half the distance from the right perimeter as the current position of the first adjust handle, and as far down as possible before the arrowhead intersect the trunk. A second text rect is defined for vertical text.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@4,1@0@0@2@5@5@12,0@12,.21600@1,21600@1@0@2,21600@2xe</th>
</tr>
</thead>
</table>

**Guide Formulas**

<table>
<thead>
<tr>
<th>val #0</th>
<th>val #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>prod #0 1 2</td>
<td></td>
</tr>
<tr>
<td>sum @3 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum 21600 #0 #1</td>
<td></td>
</tr>
<tr>
<td>sum #1 #2 0</td>
<td></td>
</tr>
<tr>
<td>prod @6 1 2</td>
<td></td>
</tr>
<tr>
<td>prod #1 1 2</td>
<td></td>
</tr>
<tr>
<td>sum @8 0 21600</td>
<td></td>
</tr>
<tr>
<td>prod 21600 @0 @1</td>
<td></td>
</tr>
<tr>
<td>prod 21600 @5 @1</td>
<td></td>
</tr>
<tr>
<td>prod 21600 @7 @1</td>
<td></td>
</tr>
<tr>
<td>prod #1 1 2</td>
<td></td>
</tr>
<tr>
<td>sum @4 0 @4</td>
<td></td>
</tr>
<tr>
<td>sum @15 @16</td>
<td></td>
</tr>
</tbody>
</table>

**Adjustment Values** 9257,18514,7200

**Connector Locations** @4,0;@0;@1;@11;@14,21600;@1;@13,21600,2

**Connector Angles** 270,180,180,90,0,0

**Text Box Rectangle** 0,0,1,21600;0,5,1,21600

**Handles**

<table>
<thead>
<tr>
<th>position=&quot;#0,topLeft&quot; xrange=&quot;@2,9&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>position=&quot;#1,#2&quot; xrange=&quot;@4,21600&quot; yrange=&quot;0,0&quot;</td>
</tr>
</tbody>
</table>
Curved Right Arrow

Internal Name: CurvedRightArrow
Shaped Concentric Fill: No.
Joins: Mitered.
Endcaps: Rounded.

Adjustments: The first adjust handle is a perimeter adjust handle that controls the curvature of the arrow. The second adjust handle controls both the length of the arrowhead, and the width of the trunk.

Geometric properties:

Path:
```
gr,0@23@3@22,0@4,0@15@25@1,0@7@2@13l@2@14@22@8@2@12w,a,0@23@3@2@11@26@17,0@15@23@1@26@17@22@15xear,0@23@3,0@4@26@17nfe
```

Guide Formulas:
```
val #0
val #1
val #2
sum #0 width #1
prod @3 1 2
sum #1 #1 width
t@3 1 2
sum @5 1 #0
prod @6 1 2
mid width #0
t#0 0 #2
ellipse @9 height @4	sum @4 1 0	sum @7 1 0
tsum @12 width #0
tsum @5 0 #0
prod @15 1 2
mid @4 @7
t#0 0 #1 width
prod @18 1 2
tsum @17 0 10
val width
tval height
prod height 2 1
tsum @17 0 @4
ellipse @24 @4 height
tsum height 0 @25
sum @8 128 0
prod @5 1 2
sum @5 0 128
sum @0 0 @17 @12
ellipse @20 @4 height
sum width 0 #0
prod @32 1 2
prod height height 1
prod @9 @9 1
sum @34 0 @35
sqrt @36
tsum @37 height 0
prod width height @38
tsum @39 64 0
prod @0 1 2
ellipse @33 @4 height
sum height 0 @42
sum @43 64 0
prod @4 1 2
tsum @1 0 @45
prod height 4390 32768
prod height 28378 32768
```

Adjustment Values:
12960, 19440, 14400

Connector Locations:
0, @17, @2, @14, @22, @8, @2, @12, @22, @16

Connector Angles:
180, 90, 0, 0, 0

Text Box Rectangle:
47, 45, 48, 46

Handles:
```
position="bottomRight,#0" yrange="@40, @29
position="bottomRight,#1" yrange="@27, @21
position="#2,bottomRight" xrange="@44, @22
```
**Curved Left Arrow**

*Internal Name:* CurvedLeftArrow  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.

**Adjustments:** The first adjust handle is a perimeter adjust handle that controls the curvature of the arrow. The second adjust handle controls both the length of the arrowhead, and the width of the trunk.

**Geometric properties:**

| Path | wr@22,0@21@3,0@21@4@22@14@21@1@21@7@2@121@2@13,0@8@2@11@22,0@21@3@2@10@24@16@22@14@21@1@24@16,0@14xear@22@14@21@1@21@7@24@16

<table>
<thead>
<tr>
<th>Guide Formulas</th>
</tr>
</thead>
</table>
| val #0  
| val #1  
| val #2  
| sum #0 width #1  
| prod @3 1 2  
| sum #1 #1 width  
| sum @5 #1 #0  
| prod @6 1 2  
| mid width #0  
| ellipse #2 height @4  
| sum @4 @9 0  
| sum @10 #1 width  
| sum @7 @9 0  
| sum @11 width #0  
| sum @5 0 #0  
| prod @14 1 2  
| mid @4 @7  
| sum #0 #1 width  
| prod @17 1 2  
| sum @16 0 @18  
| val width  
| val height  
| sum 0 0 height  
| sum @16 0 @4  
| ellipse @23 @4 height  
| sum @8 128 0  
| prod @5 1 2  
| sum @5 0 128  
| sum @0 @16 @11  
| sum width 0 #0  
| prod @29 1 2  
| prod height height 1  
| prod #2 #2 1  
| sum @31 0 @32  
| sqrt @33  
| sum @34 height 0  
| prod width height @35  
| sum @36 64 0  
| prod #0 1 2  
| ellipse @30 @38 height  
| sum @39 0 64  
| prod @4 1 2  
| sum #1 0 @41  
| prod height 4390 32768  
| prod height 28378 32768 |

| Adjustment Values | 12960,19440,7200 |
| Connector Locations | 0, @15; @2, @11; @0, @8; @2, @13; @21, @16 |
| Connector Angles | 180, 180, 180, 90, 0 |
| Text Box Rectangle | @43, @41, @44, @42 |
| Handles | position="topLeft,0" yrange="@37, @27"  
| position="topLeft,1" yrange="@25, @20"  
| position="#2, bottomRight" xrange="0, @40" |
Curved Up Arrow

Internal Name: CurvedUpArrow

Shaped Concentric Fill: No.

Joins: Mitered.

Endcaps: Rounded.

Adjustments: The first adjust handle is a perimeter adjust handle that controls the curvature of the arrow. The second adjust handle controls both the length of the arrowhead, and the width of the trunk.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>ar0@22@3@21.0@4@21@14@22@1@21@21@7@21@12@21@13@2@8.0@11@2wa0@22@3@21@10@2@16@24@14@22@1@21@16@<a href="mailto:24@14.xewr">24@14.xewr</a> @14@22@1@21@7@21@16@2-Infe</th>
</tr>
</thead>
</table>

Guide Formulas

<table>
<thead>
<tr>
<th>val #0</th>
<th>val #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>prod @3 1 2</td>
<td></td>
</tr>
<tr>
<td>sum #0 width #1</td>
<td></td>
</tr>
<tr>
<td>mid width #0</td>
<td></td>
</tr>
<tr>
<td>sum @5 #1 #0</td>
<td></td>
</tr>
<tr>
<td>prod @6 1 2</td>
<td></td>
</tr>
<tr>
<td>ellipse #2 height @4</td>
<td></td>
</tr>
<tr>
<td>sum @4 @9 0</td>
<td></td>
</tr>
<tr>
<td>sum @10 #1 width</td>
<td></td>
</tr>
<tr>
<td>sum @7 @9 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 width #0</td>
<td></td>
</tr>
<tr>
<td>sum @5 0 #0</td>
<td></td>
</tr>
<tr>
<td>prod @14 1 2</td>
<td></td>
</tr>
<tr>
<td>mid @4 @7</td>
<td></td>
</tr>
<tr>
<td>sum #0 #1 width</td>
<td></td>
</tr>
<tr>
<td>prod @17 1 2</td>
<td></td>
</tr>
<tr>
<td>sum @16 0 @18</td>
<td></td>
</tr>
<tr>
<td>val width</td>
<td></td>
</tr>
<tr>
<td>val height</td>
<td></td>
</tr>
<tr>
<td>sum @16 0 @4</td>
<td></td>
</tr>
<tr>
<td>ellipse @23 @4 height</td>
<td></td>
</tr>
<tr>
<td>sum @8 128 0</td>
<td></td>
</tr>
<tr>
<td>prod @5 1 2</td>
<td></td>
</tr>
<tr>
<td>sum @5 0 128</td>
<td></td>
</tr>
<tr>
<td>sum #0 @16 @11</td>
<td></td>
</tr>
<tr>
<td>sum width 0 #0</td>
<td></td>
</tr>
<tr>
<td>prod @29 1 2</td>
<td></td>
</tr>
<tr>
<td>prod height height 1</td>
<td></td>
</tr>
<tr>
<td>prod #2 #2 1</td>
<td></td>
</tr>
<tr>
<td>sum @31 0 @32</td>
<td></td>
</tr>
<tr>
<td>sqrt @33</td>
<td></td>
</tr>
<tr>
<td>sum @34 height 0</td>
<td></td>
</tr>
<tr>
<td>prod width height @35</td>
<td></td>
</tr>
<tr>
<td>sum @36 64 0</td>
<td></td>
</tr>
<tr>
<td>prod @16 0 #0</td>
<td></td>
</tr>
<tr>
<td>ellipse @30 @38 height</td>
<td></td>
</tr>
<tr>
<td>sum @39 0 64</td>
<td></td>
</tr>
<tr>
<td>prod @4 1 2</td>
<td></td>
</tr>
<tr>
<td>sum #1 0 @44</td>
<td></td>
</tr>
<tr>
<td>prod height 4390 32768</td>
<td></td>
</tr>
<tr>
<td>prod height 28378 32768</td>
<td></td>
</tr>
</tbody>
</table>

Adjustment Values

12960, 19440, 7200

Connector Locations

@8.0@11, @2; @15,0@16, @21; @13, @2

Connector Angles

270, 270, 270, 90, 0

Text Box Rectangle

"@41, @43, @42, @44

Handles

position="#0,topLeft" xrange="@37, @27" position="#1, topLeft" xrange="@25, @20" position="bottomRight, #2" yrange="0, @40"
**Curved Down Arrow**

*Internal Name:* CurvedDownArrow  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* The first adjust handle is a perimeter adjust handle that controls the curvature of the arrow. The second adjust handle controls both the length of the arrowhead, and the width of the trunk.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>wr,0@3@23,0@22@4,0@15,0@1@23@7,0@13@21@14@2@8@22@12@2at,0@3@23@11@2@17@26@15,0@123@17@26@15@222xewr,0@3@23@4,0@17@26fe</th>
</tr>
</thead>
</table>

| Guide Formulas | val #0  
|--------------|------  
| val #1 |  
| val #2 |  
| sum #0 width #1 |  
| prod @3 1 2 |  
| sum #1 1 width |  
| sum @5 1 #0 |  
| prod @6 1 2 |  
| mid width #0 |  
| sum height 0 #2 |  
| ellipse @9 height @4 |  
| sum @4 @10 0 |  
| sum @7 @10 0 |  
| sum @12 width #0 |  
| sum @5 0 #0 |  
| prod @15 1 2 |  
| mid @4 @7 |  
| sum #0 1 width |  
| prod @18 1 2 |  
| sum @17 0 @19 |  
| val width |  
| val height |  
| prod height 2 1 |  
| sum @17 0 @4 |  
| ellipse @24 @4 height |  
| sum height 0 @25 |  
| sum @8 128 0 |  
| prod @5 1 2 |  
| sum @5 0 128 |  
| sum @0 @17 @12 |  
| ellipse @20 @4 height |  
| sum width 0 #0 |  
| prod @32 1 2 |  
| prod height height 1 |  
| prod @9 @1 |  
| sum @34 0 35 |  
| sqrt @36 |  
| sum @37 height 0 |  
| prod width height @38 |  
| sum @39 64 0 |  
| prod #0 1 2 |  
| ellipse @33 @41 height |  
| sum height 0 @42 |  
| sum @43 64 0 |  
| prod @4 1 2 |  
| sum #1 0 @45 |  
| prod height 4390 32768 |  
| prod height 28378 32768 |  

<table>
<thead>
<tr>
<th>Adjustment Values</th>
<th>12960,19440,14400</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>@17.0;@16.0;@22,;@12,;@2,;@8,;@22,;@14,;@2</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270.0,90.0,90.0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@45,;@47,;@46,;@48</td>
</tr>
</tbody>
</table>

| Handles | position="#0,bottomRight" xrange="@40,;@29"  
|----------|---------------------------------------------|
|         | position="#1,bottomRight" xrange="@27,;@21"  
|         | position="bottomRight,#2" yrange="@44,;@22"  
|         | position="bottomRight,;#2" yrange="@44,;@22" |
**Striped Right Arrow**

*Internal Name:* StripedRightArrow

*Shaped Concentric Fill:* No.

*Joins:* Mitered.

*Endcaps:* Rounded.

*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. The adjust handle can extend as far left as where the stripes are and halfway down.

**Geometric properties:**

| Path | m@0,l@0@1,3,375@1,3,375@2@0@2@0,2,1600,2,1600,10,800xem1350@111
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>350@2,2,700@2,2,700@1xem0@110@2,6,75@2,6,75@1xe</td>
</tr>
</tbody>
</table>
| Guide Formulas | val #0
| | val #1
| | sum height 0 #1
| | sum 10,800 0 #1
| | sum width 0 0 #0
| | prod @4 @3 10,800
| | sum width 0 0 @5 |
| Adjustment Values | 16,200,5,400 |
| Connector Locations | @0,0,10,800;@0,2,1600,2,1600,10,800 |
| Connector Angles | 270,180,90,0 |
| Text Box Rectangle | 3,375,1,1,6,2 |
| Handles | position="#0,#1" xrange="3,375,2,1600" yrange="0,10,800" |

**Notched Right Arrow**

*Internal Name:* NotchedRightArrow

*Shaped Concentric Fill:* No.

*Joins:* Mitered.

*Endcaps:* Rounded.

*Adjustments:* One adjust handle controls both the length of the arrowhead, and the width of the trunk. The notch at the end stays adjusted so that it matches the shape of the arrowhead. The adjust handle can extend all of the way across and halfway down.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,l@0@1,0,1@5,10,800,0@2@0@2@0,2,1600,2,1600,10,800xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0
| | val #1
| | sum height 0 #1
| | sum 10,800 0 #1
| | sum width 0 #0
| | prod @4 @3 10,800
| | sum width 0 #5 |
| Adjustment Values | 16,200,5,400 |
| Connector Locations | @0,0,10,800;@0,2,1600,2,1600,10,800 |
| Connector Angles | 270,180,90,0 |
| Text Box Rectangle | @5,1,1,6,2 |
| Handles | position="#0,#1" xrange="0,2,1600" yrange="0,10,800" |

**Pentagon**

*Internal Name:* HomePlate

*Shaped Concentric Fill:* Yes.

*Joins:* Mitered.

*Endcaps:* Rounded.

*Adjustments:* A top perimeter adjust handle controls the length of the pointed end.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,l,2,1600@0,2,1600,2,1600,10,800xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0
| | prod #0 1 2 |
| Adjustment Values | 16,200 |
| Connector Locations | @1,0,0,10,800;@1,2,1600,2,1600,10,800 |
| Connector Angles | 270,180,90,0 |
| Text Box Rectangle | 0,0,10,800,2,1600,0,0,16,200,2,1600,0,0,2,1600,2,1600 |
| Handles | position="#0, topLeft" xrange="0,2,1600" |
**Chevron**

*Internal Name:* Chevron  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* A top perimeter adjust handle controls the length of the pointed end.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,0@1,10800,21600@0,21600,21600,10800xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
|              | sum 21600 0 @0  
|              | prod #0 1 2 | |
| Adjustment Values | 16200 |
| Connector Locations | @2,0;@1,10800;@2,21600;21600,10800 |
| Connector Angles | 270,180,90,0 |
| Text Box Rectangle | 0,0,10800,21600;0,0,16200,21600;0,0,21600,21600 |
| Handles | position="#0,topLeft" xrange="0,21600" |

**Right Arrow Callout**

*Internal Name:* RightArrowCallout  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* The first adjust handle is a top perimeter adjust handle which controls the width of the text rectangle area. The second adjust handle is a right perimeter adjust handle which controls the width the arrowhead. The third adjust handle controls both the length of the arrowhead, and the width of the trunk.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m,121600@0,21600@0@5@5@2@5@2@4,21600,10800@2@1@2@3@0@3@0,0,x</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
|              | val #1  
|              | val #2  
|              | val #3  
|              | sum 21600 0 @1  
|              | sum 21600 0 @3  
|              | prod #0 1 2 | |
| Adjustment Values | 14400,5400,18000,8100 |
| Connector Locations | @6,0;0,10800;@5,21600;21600,10800 |
| Connector Angles | 270,180,90,0 |
| Text Box Rectangle | 0,0,10800,21600 |
| Handles | position="#0,topLeft" xrange="0,2"  
|          | position="bottomRight,#1" yrange="0,3"  
|          | position="#2,#3" xrange="@0,21600" yrange="@1,10800" |
### Left Arrow Callout

**Internal Name:** LeftArrowCallout  
**Shaped Concentric Fill:** No.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** The first adjust handle is a top perimeter adjust handle which controls the width of the text rectangle area. The second adjust handle is a left perimeter adjust handle which controls the width the arrowhead. The third adjust handle controls both the length of the arrowhead, and the width of the trunk.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0,0@3@2@3@2@1@1,10800@2@4@2@5@5@0@5@0,0,21600,21600,21600,21600,xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
| | val #1  
| | val #2  
| | val #3  
| | sum 21600 0 #1  
| | sum 21600 0 #3  
| | sum #0 21600 0 |
| Adjustment Values | 7200,5400,3600,8100 |
| Connector Locations | @7,0,10800;10800,21600;21600,10800 |
| Connector Angles | 270,180,90,0 |
| Handles | position="#0,topLeft" xrange="@2,21600"  
| | position="topLeft,#1" yrange="0,3"  
| | position="#2,#3" xrange="0,0" yrange="@1,10800" |

### Up Arrow Callout

**Internal Name:** UpArrowCallout  
**Shaped Concentric Fill:** No.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** The first adjust handle is a left perimeter adjust handle which controls the height of the text rectangle area. The second adjust handle is a top perimeter adjust handle which controls the width the arrowhead. The third adjust handle controls both the length of the arrowhead, and the width of the trunk.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m0@0@3@0@3@2@1@1,10800,0@5@5@5@0,21600@0,21600,21600,xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
| | val #1  
| | val #2  
| | val #3  
| | sum 21600 0 #1  
| | sum 21600 0 #3  
| | sum #0 21600 0  
| | prod @6 1 2 |
| Adjustment Values | 7200,5400,3600,8100 |
| Connector Locations | 0,0,10800,0,0,10800,21600,21600,10800 |
| Connector Angles | 270,180,90,0 |
| Handles | position="topLeft,#0" yrange="@2,21600"  
| | position="#1,topLeft" xrange="0,3"  
| | position="#3,#2" xrange="@1,10800" yrange="0,0" |
### Down Arrow Callout

**Internal Name:** DownArrowCallout  
**Shaped Concentric Fill:** No.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** The first adjust handle is a left perimeter adjust handle which controls the width of the text rectangle area. The second adjust handle is a bottom perimeter adjust handle which controls the width the arrowhead. The third adjust handle controls both the length of the arrowhead, and the width of the trunk.

**Geometric properties:**

| Path | m.121600,,21600@0@5@0@5@2@4@2,10800,21600@1@2@3@2@3@0,0 @0xe |
| Guide Formulas | val #0  
| | val #1  
| | val #2  
| | val #3  
| | sum 21600 0 #1  
| | sum 21600 0 #3  
| | prod #0 1 2  |
| Adjustment Values | 14400,5400,18000,8100  
| Connector Locations | 10800,0;0,6,10800,21600;21600,10800  
| Connector Angles | 270,180,90,0  
| Text Box Rectangle | 0,0,21600,21600  
| Handles | position="topLeft,#0" yrange="0,0"  
| | position="#1,bottomRight" xrange="0,0"  
| | position="#3,#2" xrange="@1,10800" yrange="@0,21600" |

### Left-Right Arrow Callout

**Internal Name:** LeftRightArrowCallout  
**Shaped Concentric Fill:** No.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** The first adjust handle is a top perimeter adjust handle which controls the width of the text rectangle area. The second adjust handle is a right perimeter adjust handle which controls the width the arrowhead. The third adjust handle controls both the length of the arrowhead, and the width of the trunk. It stays symmetric so that the arrows at each end are the same size.

**Geometric properties:**

| Path | m@0,0@0@2@3@2@2@1,10800@2@4@2@5@0@5@0,21600@8,21600@8@5@9@5@9@4,21600,10800@9@1@9@3@8@3@8@3@8,xe |
| Guide Formulas | val #0  
| | val #1  
| | val #2  
| | val #3  
| | sum 21600 0 #1  
| | sum 21600 0 #3  
| | sum #0 21600 0  
| | prod 6 1 2  
| | sum 21600 0 #0  
| | sum 21600 0 #2  |
| Adjustment Values | 5400,5400,2700,8100  
| Connector Locations | 10800,0;0,10800,10800,21600;21600,10800  
| Connector Angles | 270,180,90,0  
| Text Box Rectangle | @0,0,8,21600  
| Handles | position="#0,topLeft" xrange="@2,10800"  
| | position="#0,topLeft" xrange="@2,10800"  
| | position="#2,#3" xrange="0,0" yrange="@1,10800" |

### Diagram

The diagrams illustrate the geometric properties and adjustments for the Down Arrow Callout and Left-Right Arrow Callout, including path definitions, guide formulas, adjustment values, connector locations, connector angles, text box rectangles, and handle positions.
### Up-Down Arrow Callout

**Internal Name:** UpDownArrowCallout  
**Shaped Concentric Fill:** No.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** The first adjust handle is a left perimeter adjust handle which controls the width of the text rectangle area. The second adjust handle is a top perimeter adjust handle which controls the width of the arrowhead. The third adjust handle controls both the length of the arrowhead, and the width of the trunk. It stays symmetric so that the arrows at each end are the same size.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>\begin{align*} &amp; m0@0l@3@0l@3@2l@1l@2,10800,0l@4l@2l@5l@2l@5l@0l@8l@0l@8l@0l@8l@0l@8l@3l@9l@3l@9l@1l@2l@6l@0l@8l@9l@4l@9l@5l@8l@5l@8l@5l@8l@5l@8l@5l@9l@4l@9l@10800,21600l@1l@9l@3l@9l@3l@8l@0l@8l@0l@8l@5l@2l@5l@2l@4l,10800l@2l@1l@2l@3l@0l@3xe \end{align*}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>\begin{align*} &amp; \text{val #0} \ &amp; \text{val #1} \ &amp; \text{val #2} \ &amp; \text{val #3} \ &amp; \text{sum 21600 0 #1} \ &amp; \text{sum 21600 0 #3} \ &amp; \text{sum #0 21600 0} \ &amp; \text{prod 6 1 2} \ &amp; \text{sum 21600 0 #0} \ &amp; \text{sum 21600 0 #2} \end{align*}</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>5400,5400,2700,8100</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>10800,0,0,10800,10800,21600,21600,10800</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270,180,90,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>0,0,21600,8</td>
</tr>
</tbody>
</table>

| Handles | position="topLeft," range="2,10800"  
|---|---|
| | position="#1," range="0,8"  
| | position="#3," range="1,10800" range="0,0" |

### Quad Arrow Callout

**Internal Name:** CloudCallout  
**Shaped Concentric Fill:** No.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** The first adjust handle is a left perimeter adjust handle which controls the width of the text rectangle area. The second adjust handle is a top perimeter adjust handle which controls the width of the arrowhead. The third adjust handle controls both the length of the arrowhead, and the width of the trunk. It stays symmetric so that the arrows on each side stay the same size when the shape is scaled to a 1:1 aspect ratio.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>\begin{align*} &amp; m@0@0l@3@0l@3@2l@1l@2,10800,0l@4l@2l@5l@2l@5l@0l@8l@0l@8l@0l@8l@3l@9l@3l@9l@1l@2l@6l@0l@8l@9l@4l@9l@5l@8l@5l@8l@5l@8l@5l@8l@5l@9l@4l@9l@10800,21600l@1l@9l@3l@9l@3l@8l@0l@8l@0l@8l@5l@2l@5l@2l@4l,10800l@2l@1l@2l@3l@0l@3xe \end{align*}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>\begin{align*} &amp; \text{val #0} \ &amp; \text{val #1} \ &amp; \text{val #2} \ &amp; \text{val #3} \ &amp; \text{sum 21600 0 #1} \ &amp; \text{sum 21600 0 #3} \ &amp; \text{sum #0 21600 0} \ &amp; \text{prod 6 1 2} \ &amp; \text{sum 21600 0 #0} \ &amp; \text{sum 21600 0 #2} \end{align*}</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>5400,8100,2700,9450</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>Rectangle</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@0,0,8,8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Handles</th>
<th>position=&quot;topLeft,&quot; range=&quot;2,0&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>position=&quot;#1,&quot; range=&quot;0,3&quot;</td>
</tr>
<tr>
<td></td>
<td>position=&quot;#3,&quot; range=&quot;1,10800&quot; range=&quot;0,0&quot;</td>
</tr>
</tbody>
</table>
Circular Arrow

*Internal Name:* CircularArrow

*Shaped Concentric Fill:* No.

*Joins:* Mitered.

*Endcaps:* Rounded.

*Adjustments:* A polar adjust handle along the outer circle controls the starting angle. A second polar adjust handle controls the ending angle, and the inner radius. The shape goes clockwise from the starting angle to the ending angle.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>Guide Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>val #1</td>
<td>val #0</td>
</tr>
<tr>
<td>sum #1 0 #0</td>
<td></td>
</tr>
<tr>
<td>sum 0 0 #1</td>
<td></td>
</tr>
<tr>
<td>if @2 @2 @5</td>
<td></td>
</tr>
<tr>
<td>val #2</td>
<td></td>
</tr>
<tr>
<td>sum #2 0 2700</td>
<td></td>
</tr>
<tr>
<td>cos @10 #1</td>
<td></td>
</tr>
<tr>
<td>sin @10 #1</td>
<td></td>
</tr>
<tr>
<td>cos 13500 #1</td>
<td></td>
</tr>
<tr>
<td>sin 13500 #1</td>
<td></td>
</tr>
<tr>
<td>sum @11 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @12 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @13 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @14 10800 0</td>
<td></td>
</tr>
<tr>
<td>prod #2 1 2</td>
<td></td>
</tr>
<tr>
<td>sum @19 5400 0</td>
<td></td>
</tr>
<tr>
<td>cos @20 #1</td>
<td></td>
</tr>
<tr>
<td>sin @20 #1</td>
<td></td>
</tr>
<tr>
<td>sum @21 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @12 @23 @22</td>
<td></td>
</tr>
<tr>
<td>sum @22 @23 @11</td>
<td></td>
</tr>
<tr>
<td>cos 10800 #1</td>
<td></td>
</tr>
<tr>
<td>sin 10800 #1</td>
<td></td>
</tr>
<tr>
<td>cos #2 #1</td>
<td></td>
</tr>
<tr>
<td>sin #2 #1</td>
<td></td>
</tr>
<tr>
<td>sum @26 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @27 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @28 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @29 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @19 5400 0</td>
<td></td>
</tr>
<tr>
<td>cos @34 #0</td>
<td></td>
</tr>
<tr>
<td>sin @34 #0</td>
<td></td>
</tr>
<tr>
<td>mid #0 @1</td>
<td></td>
</tr>
<tr>
<td>sumangle @37 180 0</td>
<td></td>
</tr>
<tr>
<td>if @2 @37 @38</td>
<td></td>
</tr>
<tr>
<td>cos 10800 @39</td>
<td></td>
</tr>
<tr>
<td>sin 10800 @39</td>
<td></td>
</tr>
<tr>
<td>cos #2 @39</td>
<td></td>
</tr>
<tr>
<td>sin #2 @39</td>
<td></td>
</tr>
<tr>
<td>sum @40 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @41 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @42 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @43 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @35 10800 0</td>
<td></td>
</tr>
<tr>
<td>sum @36 10800 0</td>
<td></td>
</tr>
</tbody>
</table>

**Adjustment Values:**

-11796480,5400

**Connector Locations:**

@44, @45, @48, @49, @46, @47, @17, @18, @24, @25, @15, @16

**Text Box Rectangle:**

3163, 3163, 18437, 18437

**Handles:**

position="@3,#0" polar="10800,10800"

position="@2,#1" polar="10800,10800" radiusrange="0,10800"
### Flowchart: Process
*Internal Name:* FlowChartProcess
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* None.

#### Geometric properties:
- **Path:** m121600c21600,21600.xe
- **Connector Locations:** Rectangle

### Flowchart: Alternate Process
*Internal Name:* FlowChartAlternateProcess
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* None.

#### Geometric properties:
- **Path:** m@0,qx0@0l0@2qy@0,21600l@1,21600qx21600@2l21600@0qy@1,xe
- **Guide Formulas:**
  - val #0
  - sum width 0 #0
  - sum height 0 #0
  - prod @0 2929 10000
  - sum width 0 @3
  - sum height 0 @3
  - val width
  - val height
  - prod width 1 2
  - prod height 1 2
- **Adjustment Values:** 2700
- **Connector Locations:** @8,0;0,@9;@8,@7;@6,@9
- **TextBox Rectangle:** @3,@3,@4,@5
- **Limo:** 10800,10800

### Flowchart: Decision
*Internal Name:* FlowChartDecision
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* None.

#### Geometric properties:
- **Path:** m10800,10800,10800,21600,21600,10800xe
- **Connector Locations:** Rectangle
- **TextBox Rectangle:** 3-800,5400,16200,16200

### Flowchart: Data
*Internal Name:* FlowChartInputOutput
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* None.

#### Geometric properties:
- **Path:** m4321,121600,17204,21600,21600xe
- **Connector Locations:** 12961,0;10800,0;2161,10800;8602,21600;10800,21600;19402,10800
- **TextBox Rectangle:** 4321,0,17204,21600
Flowchart: Terminator
Internal Name: FlowChartTerminator
Shaped Concentric Fill: Yes.
Joins: Rounded.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m3475.qx,10800,3475,21600</td>
<td>Rectangle</td>
<td>1018,3163,20582,18437</td>
</tr>
</tbody>
</table>

Flowchart: Preparation
Internal Name: FlowChartPreparation
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m4353.l17214,r4386,10800</td>
<td>Rectangle</td>
<td>4353,0,17214,21600</td>
</tr>
</tbody>
</table>

Flowchart: Manual Input
Internal Name: FlowChartManualInput
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m4292121600.r21600</td>
<td>10800,2146;0,10800;21600,10800</td>
<td></td>
</tr>
</tbody>
</table>

Flowchart: Manual Operation
Internal Name: FlowChartManualOperation
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m121600,17240,21600r</td>
<td>10800,0;2180,10800;10800,21600;19420,10800</td>
<td></td>
</tr>
</tbody>
</table>

FlowChart: Connector
Internal Name: FlowChartConnector
Shaped Concentric Fill: Yes.
Joins: Rounded.
Endcaps: Rounded.
Adjustments: None. Issue: the name “connector” is confusing since Escher has another type of shape called a “connector”, but this is the official name of this Flowchart shape.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m10800.qx,10800,10800,21600</td>
<td>10800,0;3163,3163,0,18437;10800,21600;18437,3163,3163,18437</td>
<td>3163,3163,18437,18437</td>
</tr>
</tbody>
</table>
**FlowChart: Off-page Connector**

*Internal Name:* FlowChartOffpageConnector  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m,l21600,r,17255,10800,21600,,17255xe</td>
<td>Rectangle</td>
<td>0,0,21600,17255</td>
</tr>
</tbody>
</table>

**FlowChart: Card**

*Internal Name:* FlowChartPunchedCard  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m4321,l21600,r,21600,12600,4338xe</td>
<td>Rectangle</td>
<td>0,4321,21600,21600</td>
</tr>
</tbody>
</table>

**FlowChart: Punched Tape**

*Internal Name:* FlowChartPunchedTape  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m21597,19450v-225,-558,-750,-1073,-1650,-1545c18897,17605,17585,17347,16197,17260v-1500,87,-2700,345,-3787,645c11472,18377,10910,10992,10800,0,199450v-188,515,-750,1075,-1613,1460c8100,21210,6825,21425,5400,21597,-3937,21425,2700,21210,1612,20910,675,20525,150,19965,19450l2417v150,-558,675,1073,1612,1460c2700,3950,3937,4165,5400,4337,6825,4165,8100,3950,4165,5400,4337,6825,4165,8100,3950,4165,5400,4337,6825,1460c10910,1632,11472,1072,12410,0,600,13497,300,14697,85,16197,v1388,85,2700,300,3750,600c20847,1072,21372,1632,21597,2147xe</td>
<td>10800,2147,0,10800;10800,19450,21600,10800</td>
<td>0,4337,21600,17260</td>
</tr>
</tbody>
</table>

**FlowChart: Summing Junction**

*Internal Name:* FlowChartSummingJunction  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m10800,qx,10800,10800,21600,21600,10800,10800,3163,3163in18437,18437em3163,18437in18437,3163e</td>
<td>10800,0,3163,3163;0,10800,3163,18437;10800,21600,18437,18437,21600,10800,18437,3163</td>
<td>3163,3163,18437,18437</td>
</tr>
</tbody>
</table>
FlowChart: Or
Internal Name: FlowChartOr
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m10800.qx,10800,10800,21600,21600,10800,10800,xem,10800.xf,10800,21600,0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>10800.0;3163,3163,10800,18437,3163,21600;18437,18437,18437,21600.10800;18437,3163</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>3163,3163,3163,18437,18437</td>
</tr>
</tbody>
</table>

FlowChart: Collate
Internal Name: FlowChartCollate
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m21600.21600l,21600,21600,,xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>10800.0;10800,10800,10800,21600</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>5400,5400,16200,16200</td>
</tr>
</tbody>
</table>

FlowChart: Sort
Internal Name: FlowChartSort
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m10800.110800,10800,10800,10800,10800,xem,10800.xf,10800,21600,0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>10800.0;5400,10800,10800,16200,10800</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>5400,5400,16200,16200</td>
</tr>
</tbody>
</table>

FlowChart: Extract
Internal Name: FlowChartExtract
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m10800.l21600,21600,21600,,xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>10800.0;5400,10800,10800,16200,10800</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>5400,10800,16200,21600</td>
</tr>
</tbody>
</table>

FlowChart: Merge
Internal Name: FlowChartMerge
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m.l21600,.10800,21600xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>10800.0;5400,10800,10800,16200,10800</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>5400,0,16200,10800</td>
</tr>
</tbody>
</table>
FlowChart: Stored Data
Internal Name: FlowChartOnlineStorage
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:
Path: m3600,21597c2662,21202,1837,20075,1087,18440,487,16240,75,13590,,1077 0,75,8007,487,5412,1087,3045,1837,1465,2662,337,3600,21597v937,337,-1 687,1465,-2512,3045c18485,5412,18072,8007,17997,10770v75,2820,488,547 0,1088,7670c19910,20075,20660,21202,21597,21597xe
Connector Locations: 10800,0,10800,10800,21600,17997,10800
Text Box Rectangle: 3600,0,17997,21600

FlowChart: Delay
Internal Name: FlowChartDelay
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:
Path: m10800,qx21600,10800,10800,21600l,21600,xe
Connector Locations: Rectangle
Text Box Rectangle: 0.3163,18437,18437

FlowChart: Sequential Access Storage
Internal Name: FlowChartMagneticTape
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:
Path: ar,21600,21600,18685,18165,10677,21597120990,21597xe
Connector Locations: Rectangle
Text Box Rectangle: 3163,3163,18437,18437

FlowChart: Magnetic Disk
Internal Name: FlowChartMagneticDisk
Shaped Concentric Fill: Yes.
Joins: Rounded.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:
Path: m10800,qx,3391l,18209qy10800,21600,21600,18209,l,18019,qx21600,10 800xe,3391infy10800,6782,21600,3391e
Connector Locations: 10800,0,10800,10800,10800,21600,10800
Connector Angles: 270,270,180,90,0,0
Text Box Rectangle: 0.6782,21600,18209

FlowChart: Direct Access Storage
Internal Name: FlowChartMagneticDrum
Shaped Concentric Fill: Yes.
Joins: Rounded.
Endcaps: Rounded.
Adjustments: None.

Geometric properties:
Path: m21600,10800qy18019,216003581,21600qy,10800,3581,118019,qy21600,10 800qem18019,21600nfqg14438,10800,18019,xe
Connector Locations: 10800,0,10800,10800,21600,14438,10800,10800
Connector Angles: 270,180,90,0,10
Text Box Rectangle: 3581,0,14438,21600
**FlowChart: Display**

*Internal Name: FlowChartDisplay*

- Shaped Concentric Fill: Yes.
- Joins: Mitered.
- Endcaps: Rounded.
- Adjustments: None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m17955,v862,282,1877,1410,2477,3045c21035,5357,21372,7895,21597,1082 7v-225,2763,-562,5300,-1165,7613c19832,20132,18817,21260,17955,21597r- 14388,1,10827,3567,xe</td>
<td>Rectangle</td>
<td>3567,0,17955,21600</td>
</tr>
</tbody>
</table>

**Stars & Banners**

**Explosion 1**

*Internal Name: IrregularSeal1*

- Shaped Concentric Fill: No.
- Joins: Mitered.
- Endcaps: Rounded.
- Adjustments: None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m10800,580008352,2295,7312,6320,370,2295,4627,7617,.8615r3722,3160113 5,14587r5532,-6504762,17617,7715,15627r770,597310532,14935r2715,480 2114020,14457r4125,363816837,12942r4763,348117607,10475,21097,8137,1 6702,7315,18380,4457r-4225,868114522,xe</td>
<td>14522,0,0,8615,8485,21600,21600,13290</td>
<td></td>
</tr>
</tbody>
</table>

**Explosion 2**

*Internal Name: IrregularSeal2*

- Shaped Concentric Fill: No.
- Joins: Mitered.
- Endcaps: Rounded.
- Adjustments: None.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>Connector Locations</th>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>m11462,43428722,1887,8550,6382,4502,3625r870,41921172,8270r2763,33 221,12877r3330,24931285,17825r3520,41514917,21600,7527,18125r1173,15 8799872,17370r1740,1472112180,15935r2762,1433514640,14350r4237,12821 6380,12310r1890,-10201r6985,9402,21600,6645,16380,6532,18007,3172,145 25,5777,14790,xe</td>
<td>9722,1887,0,12877,11612,18842,21600,6645</td>
<td></td>
</tr>
</tbody>
</table>

| Connector Angles | 270,180,90,0 |
| Connector Locations | 270,180,90,0 |
| Text Box Rectangle | 5372,6382,14640,15935 |
### 4-point Star

**Internal Name:** Seal4  
**Shaped Concentric Fill:** Yes.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** A horizontal adjust handle along the center adjusts the inner radius of the seal. The adjust handle can extend halfway across.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m21600,10800l@2@3,10800,0@3@3,.10800@3@2,10800,21600@2@2xe</th>
</tr>
</thead>
</table>
| Guide Formulas | sum 10800 0 #0  
|              | prod @0 23170 32768  
|              | sum @1 10800 0  
|              | sum 10800 0 @1  |
| Adjustment Values | 8100 |
| Connector Locations | Rectangle |
| Text Box Rectangle | @3, @3, @2, @2 |
| Handles | position="#0,center" xrange="0,10800" |

### 5-point Star

**Internal Name:** Star  
**Shaped Concentric Fill:** Yes.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** None.

**Geometric properties:**

| Path       | m10800,18280,8259,.8259r6720,51464200,21600r6600,-501917400,21600,1  
|            | 4880,13405,21600,8259-8280xe |
| Connector Locations | 10800,0;0,8259;4200,21600;17400;21600,8259 |
| Text Box Rectangle | 6720,8259,14880,15628 |

### 8-point Star

**Internal Name:** Seal8  
**Shaped Concentric Fill:** Yes.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** A horizontal adjust handle along the center adjusts the inner radius of the seal. The adjust handle can extend halfway across.

**Geometric properties:**

| Path       | m21600,10800l@3@6,18436,3163@4@5,10800@6@5,3163,3163@5@6,.1  
|            | 0800@5@4,3163,18436@6@3,10800,21600@4@3,18436,18436@3@4xe |
| Guide Formulas | sum 10800 0 #0  
|               | prod @0 30274 32768  
|               | prod @0 12540 32768  
|               | sum @1 10800 0  
|               | sum @2 10800 0  
|               | sum 10800 0 @1  
|               | sum 10800 0 @2  
|               | prod @0 23170 32768  
|               | sum @7 10800 0  
|               | sum 10800 0 @7  |
| Adjustment Values | 2538 |
| Connector Locations | Rectangle |
| Text Box Rectangle | @9, @9, @8, @8 |
| Handles | position="#0,center" xrange="0,10800" |
16-point Star

*Internal Name:* Seal16
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.

*Adjustments:* A horizontal adjust handle along the center adjusts the inner radius of the seal. The adjust handle can extend halfway across.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m21600,108000@5@10,20777,6667@7@12,18436,3163@8@11,14932,822@6@9,108000@10@9,6667,822@12@11,3163,3163@11@12,822,6667@9@10,108000@9@6,822,14932@11@8,3163,18436@12@7,6667,20777@10@5,108000,21600@6@5,14932,20777@8@7,18436,18436@7@8,20777,14932@5@6xe</th>
</tr>
</thead>
</table>

**Guide Formulas**

| sum 10800 0 @0 prod @0 32138 32768 prod @0 6393 32768 prod @0 27246 32768 prod @0 18205 32768 sum @1 10800 0 sum @2 10800 0 sum @3 10800 0 sum @4 10800 0 sum 10800 0 @1 sum 10800 0 @2 sum 10800 0 @3 sum 10800 0 @4 prod @0 23170 32768 sum @13 10800 0 sum 10800 0 @13 |

**Adjustment Values**

| 2700 |

**Connector Locations**

*Rectangle*

| @15, @15, @14, @14 |

**Text Box Rectangle**

| Position: "#0,center" Xrange="0,10800" |

**Handles**
### 24-point Star

**Internal Name:** Seal24  
**Shaped Concentric Fill:** Yes.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** A horizontal adjust handle along the center adjusts the inner radius of the seal. The adjust handle can extend halfway across.

**Geometric properties:**

| Path | m21600,108000|7@14,21232,8005@9@16,20153,5400@11@18,18437,3163@12@17,16200,1447@10@15,13595,368@8@13,10800,0@14@13,8005,368@16@15,5400,1447@18@17,3163,3163@17@18,18437,5400@15@16,368005@13@14,108000@15@8,368,13595@15@10,1447,162000@17@12,3163,18437@18@11,5400,20153@16@9,8005,21232@14@7,10800,21600@8@7,13595,21232@10@9,16200,20153@12@11,18437,18437@11@12,20153,16200@9@10,21232,13595@7@8xe |
| Guide Formulas | sum 10800 0 #0  
prod @0 32488 32768  
prod @0 4277 32768  
prod @0 30274 32768  
prod @0 12540 32768  
prod @0 25997 32768  
prod @0 19948 32768  
sum @1 10800 0  
sum @2 10800 0  
sum @3 10800 0  
sum @4 10800 0  
sum @5 10800 0  
sum @6 10800 0  
sum 10800 0 @1  
sum 10800 0 @2  
sum 10800 0 @3  
sum 10800 0 @4  
sum 10800 0 @5  
sum 10800 0 @6  
prod @0 23170 32768  
sum @19 10800 0  
sum 10800 0 @19 |
| Adjustment Values | 2700 |
| Connector Locations | Rectangle |
| Text Box Rectangle | @21.@21.@20.@20 |
| Handles | position="#0,center" xrange="0,10800" |
32-point Star

Internal Name: Seal32

Shaped Concentric Fill: Yes.

Joins: Mitered.

Endcaps: Rounded.

Adjustments: A horizontal adjust handle along the center adjusts the inner radius of the seal. The adjust handle can extend halfway across.

Geometric properties:

Path: m21600,10800l@9@18,21392,8693@11@20,20777,6667@13@22,19780,48 00@0@24,18436,3163@16@23,16800,1820@14@21,14932,822@12@19,1 2907,208@10@17,10800,0@18@17,8693,208@0@19,6667,822@22@21,4 800,1820@24@23,3163,3163@23@24,18436,4800@21@22,28899,6667@19@2 0,208,8693@17@18,10800@17@10,10808@19@12,12907,14932@21@1 4,1820,16800@23@16,3163,18436@24@15,4800,19780@22@13,6667,2077 7@20@11,8693,21392@18@9,10800,21600@10@9,12907,21392@12@11,1 4932,20777@14@13,16800,19780@16@15,18436,18436@15@16,19780,16 800@13@14,20777,14932@11@12,21392,12907@9@10se

Guide Formulas:

sum 10800 0 #0
prod 0 32610 32768
prod 0 31357 32768
prod 0 9512 32768
prod 0 28899 32768
prod 0 15447 32768
prod 0 25330 32768
prod 0 20788 32768
sum 1 10800 0
sum 2 10800 0
sum 3 10800 0
sum 4 10800 0
sum 5 10800 0
sum 6 10800 0
sum 7 10800 0
sum 8 10800 0
sum 10800 0 @1
sum 10800 0 @2
sum 10800 0 @3
sum 10800 0 @4
sum 10800 0 @5
sum 10800 0 @6
sum 10800 0 @7
sum 10800 0 @8
prod 0 23170 32768
sum 25 10800 0
sum 10800 0 @25

Adjustment Values: 2700

Connector Locations: Rectangle

Text Box Rectangle: @27, @27, @26, @26

Handles: position="#0,center" xrange="0,10800"
**Up Ribbon**

*Internal Name:* Ribbon2  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  

*Adjustments:* The first adjust handle is a top perimeter adjust handle which controls the position of the fold. The second adjust handle is a vertical adjust handle along the center which controls the tilt of the ribbon. The first adjust handle can extend as far left as the notch in the side of the ribbon and as far right as where the folds meet in the center. The second adjust handle can extend 1/3 of the way up.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>val #0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sum @0 675 0</td>
</tr>
<tr>
<td></td>
<td>sum @1 675 0</td>
</tr>
<tr>
<td></td>
<td>sum @2 675 0</td>
</tr>
<tr>
<td></td>
<td>sum @3 675 0</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @4</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @3</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @2</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @1</td>
</tr>
<tr>
<td></td>
<td>sum width 0 @0</td>
</tr>
<tr>
<td></td>
<td>val #1</td>
</tr>
<tr>
<td></td>
<td>prod @10 1 4</td>
</tr>
<tr>
<td></td>
<td>prod @10 1 2</td>
</tr>
<tr>
<td></td>
<td>prod @10 3 4</td>
</tr>
<tr>
<td></td>
<td>prod height 3 4</td>
</tr>
<tr>
<td></td>
<td>prod height 1 2</td>
</tr>
<tr>
<td></td>
<td>prod height 1 4</td>
</tr>
<tr>
<td></td>
<td>prod height 3 2</td>
</tr>
<tr>
<td></td>
<td>prod height 2 3</td>
</tr>
<tr>
<td></td>
<td>sum @11 @14 0</td>
</tr>
<tr>
<td></td>
<td>sum @12 @15 0</td>
</tr>
<tr>
<td></td>
<td>sum @13 @16 0</td>
</tr>
<tr>
<td></td>
<td>sum @17 0 @20</td>
</tr>
<tr>
<td></td>
<td>sum height 0 @10</td>
</tr>
<tr>
<td></td>
<td>sum height 0 @19</td>
</tr>
<tr>
<td></td>
<td>prod width 1 @2</td>
</tr>
<tr>
<td></td>
<td>sum width 0 2700</td>
</tr>
<tr>
<td></td>
<td>sum @25 0 2700</td>
</tr>
<tr>
<td></td>
<td>val width</td>
</tr>
<tr>
<td></td>
<td>val height</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Adjustment Values</th>
<th>5400,18900</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Locations</td>
<td>@25,0:2700,0;22;@25,0:10;@26,0:22</td>
</tr>
<tr>
<td>Connector Angles</td>
<td>270,180,90,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@0,0,0;9,0;10</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,topLeft&quot; xrange=&quot;2700,8100&quot;</td>
</tr>
<tr>
<td></td>
<td>position=&quot;center,#1&quot; yrange=&quot;14400,21600&quot;</td>
</tr>
</tbody>
</table>
**Down Ribbon**

*Internal Name:* Ribbon  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.

*Adjustments:* The first adjust handle is a bottom perimeter adjust handle which controls the position of the fold. The second adjust handle is a vertical adjust handle along the center which controls the tilt of the ribbon. The first adjust handle can extend as far left as the notch in the side of the ribbon and as far right as where the folds meet in the center. The second adjust handle can extend 1/3 of the way down.

*Geometric properties:*

<table>
<thead>
<tr>
<th>Path</th>
<th>m1@3 qx@4 11 l@4 10 l@5 10 l@5 11 qy@6 l@6 21 l@9 15 l@15 l@16 l@9 16 l@9 17 qy@8 22 l@1 1 1 17 l@0 16 16 2700 15 xm 1 1 11 nf qy@3 1 12 l@1 12 qx@0 13 1 10 l@4 10 10 em 5 5 11 nf qy@6 1 12 l@8 1 12 qx@9 1 13 1 10 l@5 10 10 em 0 0 13 nf l@0 16 em 9 13 nf l@9 16 e</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
sum @0 675 0  
sum @1 675 0  
sum @2 675 0  
sum @3 675 0  
sum width 0 @4  
sum width 0 @3  
sum width 0 @2  
sum width 0 @1  
sum width 0 @0  
val #1  
prod @10 1 4  
prod @11 2 1  
prod @11 3 1  
prod height 1 2  
sum @14 0 @12  
sum height 0 @10  
sum height 0 @11  
prod width 1 2  
sum width 0 2700  
sum @18 0 2700  
val width  
val height |
| Adjustment Values | 5400,2700 |
| Connector Locations | @18, @10, 2700, @15, @18, 21600; @19, @15 |
| Connector Angles | 270, 180, 90, 0 |
| Text Box Rectangle | @0, @10, @9, 21600 |
| Handles | position="#0,bottomRight" xrange=2700,8100  
position="center,#1" yrange=0,7200 |
**Curved Up Ribbon**

*Internal Name:* EllipseRibbon2  
*Shaped Concentric Fill:* No.  
*Joins:* Mitered.  
*Endcaps:* Rounded.

**Adjustments:** The first adjust handle is a top perimeter adjust handle which controls the position of the fold. The second adjust handle is a vertical adjust handle along the center which controls the tilt of the ribbon. The third adjust handle is a left perimeter adjust handle which controls the curvature of the ribbon. The first adjust handle can extend as far left as the notch in the side of the ribbon and as far right as where the folds meet in the center.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>wr@9@0@34@8@35,0@24@0@23@0@11@0@22@19@22@9@34@8 @35@19@23@3@24@7@36@3@44@9@31@8@32@3@24@18@30@9 @1@8@33@18@28@17@28@9@31@8@32@17@30,0@41@5@36xear@ 9@1@0@22@0@23@17@28@8@33@18@28@9@31@8@32@17@28@19 @29@18@8@33@18@30em@0@23nfl@0@29em@19@23nfl@19@29e</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
val #1  
val #2  
val width  
val height  
prod width 1 8  
prod width 1 2  
prod width 7 8  
prod width 3 2  
sum 0 0 @6  
prod #2 3073 4096  
prod @10 2 1  
sum @10 height #2  
sum @10 #1 10  
prod #1 1 2  
sum @10 @14 0  
sum @12 0 1  
sum #0 5 0  
sum width 0 1 7  
sum width 0 0  
sum @6 0 #0  
elipse @20 width @10  
sum @10 0 @21  
sum @22 0 10  
sum #2 3 10  
prod @10 2391 32768  
sum @6 0 @17  
elipse @22 width @10  
sum @10 #1 27  
sum @22 1 0  
sum @12 0 27  
sum height 0 2  
sum @10 @12 0  
sum @32 @10 @16  
sum @31 @10 @13  
sum @32 @10 @13  
sum @5 25 @12 @15  
sum @16 0 @15  
prod @37 2 3  
sum @1 38 0  
sum #2 38 0  
max @40 675  
prod width 3 8  
sum @42 4 |
| Adjustment Values | $400,16200,2700 |
| Connector Locations | @6,0@5,36@6,0@1; @7,36 |
| Connector Angles | 270,180,90,0 |
| Text Box Rectangle | @0,0@22,0@19@1 |
| Handles | position="#0,topLeft" xrange="#5,0@43  
position="center,#1" yrange="#39,0@31  
position="topLeft,#2" yrange="#41,0@24" |
Curved Down Ribbon

Internal Name: EllipseRibbon
Shaped Concentric Fill: No.
Joins: Mitered.
Endcaps: Rounded.

Adjustments: The first adjust handle is a bottom perimeter adjust handle which controls the position of the fold. The second adjust handle is a vertical adjust handle along the center which controls the tilt of the ribbon. The third adjust handle is a left perimeter adjust handle which controls the curvature of the ribbon. The first adjust handle can extend as far left as the notch in the side of the ribbon and as far right as where the folds meet in the center.

Geometric properties:

| Path | ar@9@38@8@37,0@27@0@26@9@13@8@4@0@25@22@25@9@38@8@37@22@26@3@27@7@40@3,wa@9@35@8@10@3,0@21@33@9@36@8@1@21@31@20@31@9@35@8@10@20@33,1@5@40xewr@9@36@8@1@20@31@0@32nfl@20@33ear@9@36@8@1@21@31@22@32nfl@2 1@33em@0@26nfl@0@32em@22@26nfl@22@32e |

Guide Formulas

val #0
val #1
val #2
val width
val height
prod width 1 8
prod width 1 2
prod width 7 8
prod width 3 2
sum 0 0 @6
sum height 0 #2
prod @10 30573 4096
prod @11 2 1
sum height 0 @12
sum @11 #2 0
sum @11 height #1
sum height 0 #1
prod @16 1 2
sum @11 @17 0
sum @14 #1 height
sum #0 @5 0
sum width 0 @20
sum width 0 #0
sum @6 #0
ellipse @23 width @11
sum @24 height @11
sum @25 @11 @19
sum #2 @11 @19
prod @11 2391 32768
sum @6 @0 @20
eellipse @29 width @11
sum #1 @30 @11
sum @25 #1 height
sum height @30 @14
sum @11 @14 0
sum height 0 @34
sum @35 @19 @11
sum @10 @15 @11
sum @35 @15 @11
sum @28 @14 @18
sum height 0 @39
sum @19 0 @18
prod @41 2 3
sum #1 @0 @42
sum #2 0 @42
min @44 20925
prod width 3 8
sum @46 0 4

Adjustment Values 5400,5400,18900
Connector Locations @6,0@1;@5,0@40;@6,0@4;@7,0@40
Connector Angles 270,180,90,0
Text Box Rectangle @0,0@1,22,2@25
Vertical Scroll

Internal Name: VerticalScroll

Shaped Concentric Fill: No.

Joins: Rounded.

Endcaps: Rounded.

Adjustments: A left perimeter adjusting handle controls the size of the roll on both ends. It limo-stretches both vertically and horizontally, so that the roll is always circular at both ends.

Geometric properties:

Path:
```
m@5qx@1@0@2@0@0@2@0@0@2@0@7@21600@2@0@2@0@0@2@0@1@0@0@2@0@0@2@0@1@0
```

Guide Formulas:
```
sum height 0 #0
val #0
prod @1 1 2
prod @1 3 4
prod @1 5 4
prod @1 3 2
prod @1 2 1
sum height 0 @2
sum height 0 @3
sum width 0 @5
sum width 0 @1
sum width 0 @2
val height
prod height 1 2
prod width 1 2
```

Adjustment Values:
2700

Connector Locations:
```
@14,0;@1,13;@14,12;@10,13
```

Connector Angles:
```
270,180,90,0
```

Text Box Rectangle:
```
@1,1;@10,7
```

Handles:
```
position="topLeft,#0" yrange="0,5400"
```

Limo:
```
10800,10800
```
**Horizontal Scroll**

*Internal Name:* HorizontalScroll  
*Shaped Concentric Fill:* No.  
*Joins:* Rounded.  
*Endcaps:* Rounded.  
*Adjustments:* A top perimeter adjust handle controls the size of the roll on both ends. It limo-stretches both vertically and horizontally, so that the roll is always circular at both ends.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m0@5qy@2@1l@0@1@0@2qy@7,,21600@21600@9qy@7@10@1@10@1@11qy@2,21600,0@11xem0@5nfqy@2@6@5@3@4@2@5@2@6@em@1@5nf@1@10em21600@2nfqy@7@11@0@1em@0@2nfqy@8@3@7@21@7@1e</th>
</tr>
</thead>
</table>
| Guide Formulas        | sum width 0 #0  
val #0  
prod @1 1 2  
prod @1 3 4  
prod @1 5 4  
prod @1 3 2  
prod @1 2 1  
sum width 0 @2  
sum width 0 @3  
sum height 0 @5  
sum height 0 @2  
val width  
prod width @1  
prod height @2 |
| Adjustment Values     | 2700  
Connector Locations   | @13,@10014;@13,@10012;@14  
Connector Angles      | 270,180,90.0  
Text Box Rectangle    | @1.0,7.0  
Handles               | position="#0,topLeft" xrange="0,5400  
Limo                  | 10800,10800 |
Wave

*Internal Name:* Wave
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* A left perimeter adjust handle controls the amplitude of the wave. A bottom perimeter adjust handle controls the “skew” of the wave.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@28@0c@27@1@26@3@25@01@21@4c@22@5@23@6@24@4xe</th>
</tr>
</thead>
</table>

**Guide Formulas**

val #0
prod @0 41 9
prod @0 23 9
sum 0 0 @2
sum 21600 0 #0
sum 21600 0 @1
sum 21600 0 @3
sum #1 0 10800
sum 21600 0 @1
prod @8 2 3
prod @8 4 3
prod @8 2 1
sum 21600 0 @9
sum 21600 0 @10
sum 21600 0 @11
prod @1 2 3
prod @1 4 3
prod @1 2 1
sum 21600 0 @15
sum 21600 0 @16
sum 21600 0 @17
if @7 @14 0
if @7 @15 @15
if @7 @12 @16
if @7 21600 @17
if @7 0 @20
if @7 @9 @19
if @7 @10 @18
if @7 @11 21600
sum @24 0 @21
sum @4 0 @0
max @21 @25
min @24 @28
prod @0 2 1
sum 21600 0 @33
mid @26 @27
mid @24 @28
mid @22 @23
mid @21 @25

**Adjustment Values**

2809,10800

**Connector Locations**

@35,0;@38,10800;@37,4;@36,10800

**Connector Angles**

270,180,90,0

**Text Box Rectangle**

@31,33,32,34

**Handles**

position="topLeft,#0" yrange="0,4459"
position="#1,bottomRight" xrange="8640,12960"
Double Wave

*Internal Name:* Wave  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.

**Adjustments:** A left perimeter adjust handle controls the amplitude of the wave. A bottom perimeter adjust handle controls the “skew” of the wave.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@43@0c@42@1@41@3@40@0@39@1c@38@3@37@0l@30@4c@31@5@32@66@33@4@34@5@35@66@36@4xe</th>
</tr>
</thead>
</table>

**Guide Formulas**

| val @0 | prod @0 41 9 | prod @0 23 9 | sum 0 0 @2 | sum 21600 0 #0 | sum 21600 0 @1 | sum 21600 0 @3 | sum #1 0 10800 | sum 21600 0 @1 | prod @8 1 3 | prod @8 2 3 | prod @8 4 3 | prod @8 5 3 | prod @8 2 1 | sum 21600 0 @9 | sum 21600 0 @10 | sum 21600 0 @8 | sum 21600 0 @11 | sum 21600 0 @12 | sum 21600 0 @13 | prod #1 1 3 | prod #1 2 3 | prod #1 4 3 | prod #1 5 3 | prod #1 2 1 | sum 21600 0 @20 | sum 21600 0 @21 | sum 21600 0 @22 | sum 21600 0 @23 | sum 21600 0 @24 |
|------|------------------|------------------|-------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|

**Adjustment Values**

- Adjustments: 1404,10800

**Connector Locations**

- @40,0;@51,10800;@33,4;@50,10800

**Connector Angles**

- 270,180,90,0

**Text Box Rectangle**

- @46,48,47,49

**Handles**

- position="topLeft,#0" yrange="0,2229"
- position="#1,bottomRight" xrange="8640,12960"
### Rectangular Callout

**Internal Name:** WedgeRectCallout  
**Shaped Concentric Fill:** No.  
**Joins:** Mitered.  
**Endcaps:** Rounded.  
**Adjustments:** One adjust handle controls the position of the point. The wedge extends from the center of the edge that is closest to the point. If the point is placed inside the rectangle, no wedge is drawn. Note that part of the shape extends past the geometry box of the shape.

#### Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Guide Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>m,0@8@12@24,0@9@21600@6,21600@15@27@7</td>
<td>sum 10800 0 #0</td>
</tr>
<tr>
<td></td>
<td>sum 10800 0 #1</td>
</tr>
<tr>
<td></td>
<td>sum 0 @0 #1</td>
</tr>
<tr>
<td></td>
<td>sum @0 @1 0</td>
</tr>
<tr>
<td></td>
<td>sum 21600 0 #0</td>
</tr>
<tr>
<td></td>
<td>sum 21600 0 #1</td>
</tr>
<tr>
<td>@0</td>
<td>if @0 3600 12600</td>
</tr>
<tr>
<td>@0</td>
<td>if @0 9000 18000</td>
</tr>
<tr>
<td>@0</td>
<td>if @1 3600 12600</td>
</tr>
<tr>
<td>@0</td>
<td>if @1 9000 18000</td>
</tr>
<tr>
<td>@0</td>
<td>if @2 0 @0</td>
</tr>
<tr>
<td>@0</td>
<td>if @3 @10 0</td>
</tr>
<tr>
<td>@0</td>
<td>if @0 @11</td>
</tr>
<tr>
<td>@0</td>
<td>if @2 @6 #0</td>
</tr>
<tr>
<td>@0</td>
<td>if @3 @6 @13</td>
</tr>
<tr>
<td>@0</td>
<td>if @5 @6 @14</td>
</tr>
<tr>
<td>@0</td>
<td>if @2 @0 21600</td>
</tr>
<tr>
<td>@0</td>
<td>if @3 21600 @16</td>
</tr>
<tr>
<td>@0</td>
<td>if @4 21600 @17</td>
</tr>
<tr>
<td>@0</td>
<td>if @2 @0 @6</td>
</tr>
<tr>
<td>@0</td>
<td>if @3 @19 @6</td>
</tr>
<tr>
<td>@0</td>
<td>if @1 @6 @20</td>
</tr>
<tr>
<td>@0</td>
<td>if @2 @8 #1</td>
</tr>
<tr>
<td>@0</td>
<td>if @3 @22 @8</td>
</tr>
<tr>
<td>@0</td>
<td>if @0 @8 @23</td>
</tr>
<tr>
<td>@0</td>
<td>if @2 21600 @1</td>
</tr>
<tr>
<td>@0</td>
<td>if @3 21600 @25</td>
</tr>
<tr>
<td>@0</td>
<td>if @5 21600 @26</td>
</tr>
<tr>
<td>@0</td>
<td>if @2 @1 @8</td>
</tr>
<tr>
<td>@0</td>
<td>if @3 @8 @28</td>
</tr>
<tr>
<td>@0</td>
<td>if @4 @8 @29</td>
</tr>
<tr>
<td>@0</td>
<td>if @2 @1 0</td>
</tr>
<tr>
<td>@0</td>
<td>if @3 @31 0</td>
</tr>
<tr>
<td>@0</td>
<td>if @1 0 @32</td>
</tr>
<tr>
<td>#0</td>
<td>val #0</td>
</tr>
<tr>
<td>#0</td>
<td>val #1</td>
</tr>
</tbody>
</table>

| Adjustment Values | 1350,25920 |
| Connector Locations | 10800,0,0,10800,10800,21600,21600,10800,21600,0,0@34,0@35 |
| Handles | position=’#0,#1 |
Rounded Rectangular Callout

Internal Name: WedgeRRectCallout

Shaped Concentric Fill: No.

Joins: Mitered.

Endcaps: Rounded.

Adjustments: One adjust handle controls the position of the point. The wedge extends from the center of the edge that is closest to the point. If the point is placed inside the rectangle, no wedge is drawn. Note that part of the shape extends past the geometry box of the shape.

Geometric properties:

| Path | m3600.qx,3600000@8@12@24.0@9,.18000q3600,21600@6,21600@15@27 @7.21600,18000.21600q21600,18000021600@9@18@30,21600@8,21600,3 600p18000,1@7.0@21@33@6,xe |
|-----------------------------------------------|
| Guide Formulas | sum 10800 0 #0 sum 10800 0 #1 sum #0 0 #1 sum @0 @1 0 sum 21600 0 #0 sum 21600 0 #1 if @0 3600 12600 if @0 9000 18000 if @1 3600 12600 if @1 9000 18000 if @2 0 #0 if @3 @10 0 if @0 11 if @2 @6 @0 if @3 @6 @13 if @5 @6 @14 if @2 #0 21600 if @3 21600 @16 if @4 21600 @17 if @2 #0 @6 if @3 @19 @6 if @1 @6 @20 if @2 @8 #1 if @3 @22 @8 if @0 @8 @23 if @2 21600 @1 if @3 21600 @25 if @5 21600 @26 if @2 @1 @8 if @3 @8 @28 if @4 @8 @29 if @2 @1 0 if @3 @31 0 if @1 0 @32 val #0 val #1 |
| Adjustment Values | 1350.25920 |
| Connector Locations | 10800,0,10800,10800,21600,21600,10800;@34,35 |
| Text Box Rectangle | 791,791,20809,20809 |
| Handles | position="#0,#1" |
**Oval Callout**

*Internal Name:* WedgeEllipseCallout  
*Shaped Concentric Fill:* No.  
*Endcaps:* Mitered.  
*Adjustments:* One adjust handle controls position of the point. The wedge extends from the center of the ellipse. If the point is placed inside the ellipse, no wedge is drawn. Note that part of the shape extends past the geometry box of the shape.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>wr,21600,216000@15@16@17@18@21@22x</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
val #1  
sum 10800 0 #0  
sum 10800 0 #1  
atan2 @2 @3  
sumangle @4 11 0  
sumangle @4 0 11  
cos 10800 @4  
sin 10800 @4  
cos 10800 @5  
sin 10800 @5  
cos 10800 @6  
sin 10800 @6  
sum 10800 0 @7  
sum 10800 0 @8  
sum 10800 0 @9  
sum 10800 0 @10  
sum 10800 0 @11  
sum 10800 0 @12  
mod @2 @3 0  
sum @19 0 10800  
if @20 #0 @13  
if @20 #1 @14 |
| Adjustment Values | 1350,25920 |
| Connector Locations | 10800,0;3163,3163;0,10800;3163,18437;10800,21600;18437,18437;21600,10800;18437,3163;@21,@22 |
| Text Box Rectangle | 3163,3163,18437,18437 |
| Handles | position="#0,#1 |

**Cloud Callout**

*Internal Name:* CloudCallout  
*Shaped Concentric Fill:* No.  
*Endcaps:* Rounded.  
*Adjustments:* One adjust handle controls position of the point. The three bubbles extend from the perimeter of the cloud in-line with the center of the cloud. Note that part of the shape extends past the geometry box of the shape.

**Geometric properties:**

| Path | ar,7165,4345,13110,1950,7185,1080,12690,475,11732,4835,17650,10800,12690,0,2910,17640,2387,9757,10107,20300,2910,17640,8235,19545,7660,12382,14412,21597,8235,19545,14280,18330,12910,11080,18695,18947,14280,18330,0,18690,15045,14822,5862,21597,15082,18690,15045,20895,7665,15772,2597,21105,9685,20895,7665,19140,2715,14330,19187,6595,19140,2715,14910,1170,10992,15357,5945,14910,1170,11250,1665,6692,650,12025,7917,11125,0,1665,7005,2580,1912,1972,8665,11162,7005,2580,1950,7185,7165,4345,5,13110,1080,12690,2340,13080,1475,11732,4835,17650,2910,17640,346,5,17445,15045,17070,11475,15772,2592,21105,9685,20895,7665,19140,2715xear,7165,4345,13110,1950,7185,1080,12690,475,11732,4835,17650,10800,12690,0,2910,17640,2387,9757,10107,20300,2910,17640,8235,19545,7660,12382,14412,21597,8235,19545,14280,18330,12910,11080,18695,18947,14280,18330,0,18690,15045,14822,5862,21597,15082,18690,15045,20895,7665,15772,2597,21105,9685,20895,7665,19140,2715,14330,19187,6595,19140,2715,14910,1170,10992,15357,5945,14910,1170,11250,1665,6692,650,12025,7917,11125,0,1665,7005,2580,1912,1972,8665,11162,7005,2580,1950,7185,7165,4345,5,13110,1080,12690,2340,13080,1475,11732,4835,17650,2910,17640,346,5,17445,15045,17070,11475,15772,2592,21105,9685,20895,7665,19140,2715xem@23@37qx@35@334@36@34@24@23@37xem@16@334@51@17@16@32@30@17@16@33xem@38@29q@27@39@38@28@26@39@38@29xe |
| Guide Formulas | sum #0 0 10800  
sum #1 0 10800  
cosatan2 10800 0 @0 @1 |
Line Callout 1

**Internal Name:** BorderCallout90

**Shaped Concentric Fill:** Yes.

**Joins:** Mitered.

**Endcaps:** Rounded.

**Adjustments:** The position of both ends of the line are adjustable.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@20@3nfem,l21600,r,21600x1e</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0
val #1
val #2
val #3 |
| Adjustment Values | -1800,24300,-1800,4050 |
| Connector Locations | @0, @1:10800,0;10800,21600;0,10800,21600,10800 |
| Handles | position="#0,#1
position="#2,#3" |
Line Callout 2
Internal Name: BorderCallout1
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: The position of both ends of the line are adjustable.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Guide Formulas</th>
<th>Adjustment Values</th>
<th>Connector Locations</th>
<th>Handles</th>
</tr>
</thead>
<tbody>
<tr>
<td>m@0@0@110@2@3nfem,l21600,r,21600l,21600xe</td>
<td>val #0, val #1, val #2, val #3</td>
<td>-8280,24300,-1800,4050</td>
<td>@0,@1;10800,0;10800,21600;0,10800;21600,10800</td>
<td>position=&quot;#0,#1, position=&quot;#2,#3</td>
</tr>
</tbody>
</table>

Line Callout 3
Internal Name: BorderCallout2
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: The position of both ends of the line and the joint are adjustable.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Guide Formulas</th>
<th>Adjustment Values</th>
<th>Connector Locations</th>
<th>Handles</th>
</tr>
</thead>
<tbody>
<tr>
<td>m@0@0@110@2@3@4@5nfem,l21600,r,21600l,21600xe</td>
<td>val #0, val #1, val #2, val #3, val #4, val #5</td>
<td>-10080,24300,-3600,4050,-1800,4050</td>
<td>@0,@1;10800,0;10800,21600;0,10800;21600,10800</td>
<td>position=&quot;#0,#1, position=&quot;#2,#3, position=&quot;#4,#5</td>
</tr>
</tbody>
</table>

Line Callout 4
Internal Name: BorderCallout3
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: The position of both ends of the line and both joints are adjustable.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Guide Formulas</th>
<th>Adjustment Values</th>
<th>Connector Locations</th>
<th>Handles</th>
</tr>
</thead>
<tbody>
<tr>
<td>m@0@0@110@2@3@4@5@6@7nfem,l21600,r,21600l,21600xe</td>
<td>val #0, val #1, val #2, val #3, val #4, val #5, val #6, val #7</td>
<td>23400,24400,25200,21600,25200,4050,23400,4050</td>
<td>@0,@1;10800,0;10800,21600;0,10800;21600,10800</td>
<td>position=&quot;#0,#1, position=&quot;#2,#3, position=&quot;#4,#5, position=&quot;#6,#7</td>
</tr>
</tbody>
</table>
**Line Callout 1 (Accent Bar)**

*Internal Name:* AccentCallout90  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* The position of both ends of the line are adjustable.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@2@3nfem,j21660,r,21600l,21600nsxe</th>
</tr>
</thead>
</table>
| Guide Formulas| val #0  
|               | val #1  
|               | val #2  
|               | val #3  |
| Adjustment Values | -1800,24300, -1800,4050 |
| Connector Locations | @0, @1:10800,0;10800,21600;0,10800;21600,10800 |
| Handles       | position="#0,#1  
|               | position="#2,#3  

**Line Callout 2 (Accent Bar)**

*Internal Name:* AccentCallout1  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* The positions of both ends of the line are adjustable.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@2@3nfem@2,21600hnfem,j21660,r,21600l,21600nsxe</th>
</tr>
</thead>
</table>
| Guide Formulas| val #0  
|               | val #1  
|               | val #2  
|               | val #3  |
| Adjustment Values | -2880,24300, -1800,4050 |
| Connector Locations | @0, @1:10800,0;10800,21600;0,10800;21600,10800 |
| Handles       | position="#0,#1  
|               | position="#2,#3  

**Line Callout 3 (Accent Bar)**

*Internal Name:* AccentCallout2  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.  
*Adjustments:* The position of both ends of the line and the joint are adjustable.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@2@3@4@5nfem@4,4,21600hnfem,j21660,r,21600l,21600nsxe</th>
</tr>
</thead>
</table>
| Guide Formulas| val #0  
|               | val #1  
|               | val #2  
|               | val #3  
|               | val #4  
|               | val #5  |
| Adjustment Values | -10080,24300, -3600,4050, -1800,4050 |
| ConnectorLocations | 0, @1:10800,0;10800,21600;0,10800;21600,10800 |
| Handles       | position="#0,#1  
|               | position="#2,#3  
|               | position="#4,#5  

---
**Line Callout 4 (Accent Bar)**
*Internal Name:* AccentCallout3
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* The position of both ends of the line and both joints are adjustable.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@ 2@3@4@5@6@7nfem@6,l@6,21600nfem,l21600,r,21600l,21600nsxe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
val #1  
val #2  
val #3  
val #4  
val #5  
val #6  
val #7 |
| Adjustment Values | 23400,24400,25200,21600,25200,4050,23400,4050 |
| Connector Locations | @0, @1,10800,0,10800,21600,0,10800,21600,10800 |
| Handles | position="#0,#1  
position="#2,#3  
position="#4,#5  
position="#6,#7" |

**Line Callout 1 (No Border)**
*Internal Name:* Callout90
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* The position of both ends of the line are adjustable.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@ 2@3nfem,l21600,r,21600l,21600nsxe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
val #1  
val #2  
val #3  |
| Adjustment Values | -1800,24300,1800,4050 |
| Connector Locations | @0, @1,10800,0,10800,21600,0,10800,21600,10800 |
| Handles | position="#0,#1  
position="#2,#3" |

**Line Callout 2 (No Border)**
*Internal Name:* Callout1
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.
*Adjustments:* The positions of both ends of the line are adjustable.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@ 2@3nfem,l21600,r,21600l,21600nsxe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
val #1  
val #2  
val #3  |
| Adjustment Values | 8280,24300,1800,4050 |
| Connector Locations | @0, @1,10800,0,10800,21600,0,10800,21600,10800 |
| Handles | position="#0,#1  
position="#2,#3" |
Line Callout 3 (No Border)
Internal Name: Callout2
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: The position of both ends of the line and the joint are adjustable.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@2@3@4@5nfem,l21600,r,21600l,21600nsxe</th>
</tr>
</thead>
</table>

Guide Formulas
- val #0
- val #1
- val #2
- val #3
- val #4
- val #5

Adjustment Values
- -10080,24300,-3600,4050,-1800,4050

Connector Locations @0@1;10800;0@10800;21600;0@10800;21600@10800

Handles
- position="#0,#1"
- position="#2,#3"
- position="#4,#5"

Line Callout 4 (No Border)
Internal Name: Callout3
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: The position of both ends of the line and both joints are adjustable.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@2@3@4@5@6@7nfem,l21600,r,21600l,21600nsxe</th>
</tr>
</thead>
</table>

Guide Formulas
- val #0
- val #1
- val #2
- val #3
- val #4
- val #5
- val #6
- val #7

Adjustment Values
- 23400,24400,25400,21600,25200,4050,23400,4050

Connector Locations @0@1;10800;0@10800;21600;0@10800;21600@10800

Handles
- position="#0,#1"
- position="#2,#3"
- position="#4,#5"
- position="#6,#7"

Line Callout 1 (Border and Accent Bar)
Internal Name: AccentBorderCallout90
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: The position of both ends of the line are adjustable.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>m@0@110@2@3@4@5nfem,l21600,r,21600l,21600xe</th>
</tr>
</thead>
</table>

Guide Formulas
- val #0
- val #1
- val #2
- val #3

Adjustment Values
- -1800,24300,-1800,4050

Connector Locations @0@1;10800;0@10800;21600;0@10800;21600@10800

Handles
- position="#0,#1"
- position="#2,#3"
Line Callout 2 (Border and Accent Bar)
Internal Name: AccentBorderCallout1
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: The positions of both ends of the line are adjustable.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Guide Formulas</th>
<th>Adjustment Values</th>
<th>Connector Locations</th>
<th>Handles</th>
</tr>
</thead>
<tbody>
<tr>
<td>m@0@1l0@2r3nlfem@4l5]</td>
<td>val #0, val #1, val #2, val #3</td>
<td>8280,24300,1800,4050</td>
<td>@0, @1;10800,0;10800,21600;0,10800;21600,10800</td>
<td>position=&quot;#0,#1, #2,#3</td>
</tr>
</tbody>
</table>

Line Callout 3 (Border and Accent Bar)
Internal Name: AccentBorderCallout2
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: The position of both ends of the line and the joint are adjustable.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Guide Formulas</th>
<th>Adjustment Values</th>
<th>Connector Locations</th>
<th>Handles</th>
</tr>
</thead>
<tbody>
<tr>
<td>m@0@1l0@2r3nlfem@4l5]</td>
<td>val #0, val #1, val #2, val #3, val #4, val #5</td>
<td>10080,24300,3600,4050,1800,4050</td>
<td>@0, @1;10800,0;10800,21600;0,10800;21600,10800</td>
<td>position=&quot;#0,#1, #2,#3, #4,#5</td>
</tr>
</tbody>
</table>

Line Callout 4 (Border and Accent Bar)
Internal Name: AccentBorderCallout3
Shaped Concentric Fill: Yes.
Joins: Mitered.
Endcaps: Rounded.
Adjustments: The position of both ends of the line and both joints are adjustable.

Geometric properties:

<table>
<thead>
<tr>
<th>Path</th>
<th>Guide Formulas</th>
<th>Adjustment Values</th>
<th>Connector Locations</th>
<th>Handles</th>
</tr>
</thead>
<tbody>
<tr>
<td>m@0@1l0@2r3nlfem@4l5]</td>
<td>val #0, val #1, val #2, val #3, val #4, val #5, val #6, val #7</td>
<td>23400,24400,25000,21600,25200,4050,23400,4050</td>
<td>@0, @1;10800,0;10800,21600;0,10800;21600,10800</td>
<td>position=&quot;#0,#1, #2,#3, #4,#5, #6,#7</td>
</tr>
</tbody>
</table>
Action Buttons

Action Button: Custom
- **Internal Name:** ActionButtonBlank
- **Shaped Concentric Fill:** Yes.
- **Joins:** Mitered.
- **Endcaps:** Rounded.

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>m,1.21600,21600,1.21600,xem@0@0nfl@0@2@1@2@1@0xem,nfl@0@0em,21600nfl@0@2em21600,21600nfl@1@2em21600,nfl@1@0e</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Guide Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>val #0</td>
</tr>
<tr>
<td>sum width 0 #0</td>
</tr>
<tr>
<td>sum height 0 #0</td>
</tr>
<tr>
<td>prod width 1 2</td>
</tr>
<tr>
<td>prod height 1 2</td>
</tr>
<tr>
<td>prod #0 1 2</td>
</tr>
<tr>
<td>prod #0 3 2</td>
</tr>
<tr>
<td>sum @1 @5 0</td>
</tr>
<tr>
<td>sum @2 @5 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Adjustment Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1350</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connector Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,@4;@0,4;@3,21600,3,2,21600,4,1,4,3,0;@3,0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text Box Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>@0,@0,0,1,0,2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Handles</th>
</tr>
</thead>
<tbody>
<tr>
<td>position=&quot;#0,topLeft&quot; switch=&quot;true&quot; xrange=&quot;0.5400</td>
</tr>
<tr>
<td>Limo</td>
</tr>
<tr>
<td>10800,10800</td>
</tr>
</tbody>
</table>

Action Button: Home
- **Internal Name:** ActionButtonHome
- **Shaped Concentric Fill:** Yes.
- **Joins:** Mitered.
- **Endcaps:** Rounded.

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>m,1.21600,21600,1.21600,xem@0@0nfl@0@2@1@2@1@0xem,nfl@0@0em,21600nfl@0@2em21600,21600nfl@1@2em21600,nfl@1@0e</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Guide Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>val #0</td>
</tr>
<tr>
<td>sum width 0 #0</td>
</tr>
<tr>
<td>sum height 0 #0</td>
</tr>
<tr>
<td>prod width 1 2</td>
</tr>
<tr>
<td>prod height 1 2</td>
</tr>
<tr>
<td>prod #0 1 2</td>
</tr>
<tr>
<td>prod #0 3 2</td>
</tr>
<tr>
<td>sum @1 @5 0</td>
</tr>
<tr>
<td>sum @2 @5 0</td>
</tr>
<tr>
<td>sum @0 @4 8100</td>
</tr>
<tr>
<td>sum @2 8100 @4</td>
</tr>
<tr>
<td>sum @0 @3 8100</td>
</tr>
<tr>
<td>sum @1 8100 @3</td>
</tr>
<tr>
<td>sum @10 0 @9</td>
</tr>
<tr>
<td>prod @13 1 16</td>
</tr>
<tr>
<td>prod @13 1 8</td>
</tr>
<tr>
<td>prod @13 3 16</td>
</tr>
<tr>
<td>prod @13 5 16</td>
</tr>
<tr>
<td>prod @13 7 16</td>
</tr>
<tr>
<td>prod @13 9 16</td>
</tr>
<tr>
<td>Adjustment Values</td>
</tr>
<tr>
<td>Connector Locations</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
</tr>
<tr>
<td>Handles</td>
</tr>
<tr>
<td>Limo</td>
</tr>
</tbody>
</table>

**Action Button: Help**

*Internal Name:* ActionButtonHelp

*Shaped Concentric Fill:* Yes.

*Joins:* Mitered.

*Endcaps:* Rounded.

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

| Path | m,1,21600,21600,21600,xem@0,0nd@0,2@1,2@1,0xem,xem@0,0em,21600nfl@0,2em21600,21600nfl@1,2em21600,nfl@1,0xem,33,27nfaqy@3,0@40@27@39@40@27@39@40@27@39@40@27@39@40@27@39@40@27@39@40@27@39@40@27@39@40 |
| Guide Formulas | val #0 |
| | sum width 0 #0 |
| | sum height 0 #0 |
| | prod width 1 2 |
| | prod height 1 2 |
| | prod #0 1 2 |
| | prod #0 3 2 |
| | sum @ 1 5 0 |
| | sum @ 2 5 0 |
| | sum @ 0 4 8100 |
| | sum @ 2 8100 4 |
| | sum @ 0 3 8100 |
| | sum @ 1 8100 3 |
| | sum @ 10 0 9 |
| | prod @ 13 1 7 |
### Adjustment Values

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1350</td>
</tr>
</tbody>
</table>

### Connector Locations

- `0,0;0,4;0,4;3,21600;3,0;2,21600;0,4;1,0;4,3,0;3,0`  
- `@0,0;@1,0;@2,0`  

### Text Box Rectangle

- `@0,0;@1,2`  

### Handles

- `position="#0,0;#0,2" switch="true" xrange="0,5400"`
- `Limo 10800,1080`

### Action Button: Information

**Internal Name:** ActionButtonInformation  
**Shaped Concentric Fill:** Yes.  
**Joins:** Mitered.  
**Endcaps:** Rounded.

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

- **Path**:
  ```text
  m,21600,21600,21600,21600,xem@0@0,0,0,4,2,0,1,2,4,1,0,0,3,0,0,0,0,m,
  21600,21600,21600,21600,xem@0@0,0,0,4,2,0,1,2,4,1,0,0,3,0,0,0,0
  ```
### Guide Formulas

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>val #0</td>
<td></td>
</tr>
<tr>
<td>sum width 0 #0</td>
<td></td>
</tr>
<tr>
<td>sum height 0 #0</td>
<td></td>
</tr>
<tr>
<td>prod width 1 2</td>
<td></td>
</tr>
<tr>
<td>prod height 1 2</td>
<td></td>
</tr>
<tr>
<td>prod #0 1 2</td>
<td></td>
</tr>
<tr>
<td>prod #0 3 2</td>
<td></td>
</tr>
<tr>
<td>sum @1 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @2 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @0 @4 8100</td>
<td></td>
</tr>
<tr>
<td>sum @2 8100 @4</td>
<td></td>
</tr>
<tr>
<td>sum @0 @3 8100</td>
<td></td>
</tr>
<tr>
<td>sum @1 8100 @3</td>
<td></td>
</tr>
<tr>
<td>sum @10 @9</td>
<td></td>
</tr>
<tr>
<td>prod @13 1 32</td>
<td></td>
</tr>
<tr>
<td>prod @13 5 32</td>
<td></td>
</tr>
<tr>
<td>prod @13 9 32</td>
<td></td>
</tr>
<tr>
<td>prod @13 13 32</td>
<td></td>
</tr>
<tr>
<td>prod @13 19 32</td>
<td></td>
</tr>
<tr>
<td>prod @13 5 8</td>
<td></td>
</tr>
<tr>
<td>prod @13 11 16</td>
<td></td>
</tr>
<tr>
<td>prod @13 13 16</td>
<td></td>
</tr>
<tr>
<td>prod @13 7 8</td>
<td></td>
</tr>
<tr>
<td>sum @9 @14 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @15 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @16 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @17 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @18 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @23 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @24 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @17 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @18 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @19 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @20 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @21 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @22 0</td>
<td></td>
</tr>
<tr>
<td>sum @3 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @4 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @10 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @12 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @25 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @26 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @27 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @28 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @29 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @30 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @31 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @32 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @33 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @34 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @35 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @36 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @37 @5 0</td>
<td></td>
</tr>
</tbody>
</table>

### Adjustment Values

1350

### Connector Locations

0,0;0,0;3,21600;3,21600;2,21600,0;4;1,0;4;3,0;3,0

### Text Box Rectangle

@0,0,1,2

### Handles

position="@0,topLeft" switch="true" xrange="0,5400"

### Limo

10800,10800
Action Button: Back/Previous

*Internal Name:* ActionButtonBlank
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.

Adjustments: A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m,l,21600r21600,l21600,xem,0@0@0@0@0@2@1@2@1@0@0@0@0@21600nfl@0@0@21600,21600nfl@1@2em21600,nfl@1@0em@12@9nfl@11@4@12@10xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0&lt;br&gt;sum width 0 #0&lt;br&gt;sum height 0 #0&lt;br&gt;prod width 1 2&lt;br&gt;prod height 1 2&lt;br&gt;prod #0 1 2&lt;br&gt;prod #0 3 2&lt;br&gt;sum @1 1 @0 5 0&lt;br&gt;sum @2 1 @0 5 0&lt;br&gt;sum @0 1 @0 4 8100&lt;br&gt;sum @2 1 @0 8100 4&lt;br&gt;sum @2 1 @0 8100 3&lt;br&gt;sum @1 1 @0 8100 3&lt;br&gt;sum @4 1 @0 5 0&lt;br&gt;sum @9 1 @0 5 0&lt;br&gt;sum @10 1 @0 5 0&lt;br&gt;sum @11 1 @0 5 0&lt;br&gt;sum @12 1 @0 5 0</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>1350</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>0, @4; @0, @3, 21600; @3, 21600; @3, 21600; @4; @1, @4; @3, 0; @3, 0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@0, @0, @1, @2</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,topLeft&quot; switch=&quot;true&quot; xranges=&quot;0,5400&quot;</td>
</tr>
<tr>
<td>Limo</td>
<td>10800, 10800</td>
</tr>
</tbody>
</table>
**Action Button: Forward/Next**

*Internal Name:* ActionButtonBlank  
*Shaped Concentric Fill:* Yes.  
*Joins:* Mitered.  
*Endcaps:* Rounded.

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m,121600r21600,21600,xem@0@0nfl@0@2@1@2@1@0xem,nfl@0@0em,21600nfl@0@2em21600,21600nfl@1@2em21600,nfl@1@0em@11@9nfl@12@4@11@10xe</th>
</tr>
</thead>
</table>
| Guide Formulas                                   | val #0  
sum height 0 #0  
prod height 1 2  
prod #0 1 2  
prod #0 3 2  
sum @1 @5 0  
sum @2 @5 0  
sum @0 @4 8100  
sum @2 8100 @4  
sum @0 @3 8100  
sum @1 8100 @3  
sum @4 @5 0  
sum @9 @5 0  
sum @10 @5 0  
sum @11 @5 0  
sum @12 @5 0 |
| Adjustment Values                                | 1350 |
| Connector Locations                              | 0,4,0,4;3,21600,3,21600,4,2,21600,4,1,4,3,0,3,0 |
| Text Box Rectangle                               | @0,0,1,0,2 |
| Handles                                          | position="0,topLeft" switch="true" xrange="0,5400 |
| Limo                                             | 10800,10800 |
**Action Button: Beginning**

*Internal Name: ActionButtonBlank*

*Shaped Concentric Fill: Yes.*

*Joins: Mitered.*

*Endcaps: Rounded.*

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m.L21600r21600.L21600.xem@0@0nfl@0@0@2@1@2@<a href="mailto:1@0xem.nfl">1@0xem.nfl</a>@0@0em .21600nfl@0@0@2em21600,21600nfl@1@2em21600,nfl@1@0em@12@9f@1 7@4@12@10xem@11@9f@16@9@16@10@11@10xe</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
| | sum width 0 @0  
| | sum height 0 @0  
| | prod width 1 2  
| | prod height 1 2  
| | prod #0 1 2  
| | prod #0 3 2  
| | sum @1 @5 0  
| | sum @2 @5 0  
| | sum @0 @4 8100  
| | sum @2 8100 @4  
| | sum @0 @3 8100  
| | sum @1 8100 @3  
| | sum @10 0 @9  
| | prod @13 1 8  
| | prod @13 1 4  
| | sum @11 @14 0  
| | sum @11 @15 0  
| | sum @4 @5 0  
| | sum @9 @5 0  
| | sum @10 @5 0  
| | sum @11 @5 0  
| | sum @12 @5 0  
| | sum @16 @5 0  
| | sum @17 @5 0  |
| Adjustment Values | 1350 |
| Connector Locations | 0,0@0@0@4;@0@4;@3,21600;@3,21600,4;@1,4;@3,0;@3,0 |
| TextBox Rectangle | @0,0@0@1,02 |
| Handles | position="#0,topLeft" switch="true" xranges="0,5400 |
| Limo | 10800,10800 |
Action Button: End

**Internal Name:** ActionButtonBlank

**Shaped Concentric Fill:** Yes.

**Joins:** Mitered.

**Endcaps:** Rounded.

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m.121600r21600.121600.xem@00@0nfl@0@2@1@2@<a href="mailto:1@0xem.nfl">1@0xem.nfl</a>@0@0nem .21600nfl@0@0@2em21600,21600nfl@1@2em21600,nfl@1@0em@11@91@1 6@4@11@10xem@17@91@12@9@12@10@17@10xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0  sum width 0 #0  sum height 0 #0  prod width 1 2  prod height 1 2  prod #0 1 2  prod #0 3 2  sum @1 @5 0  sum @2 @5 0  sum @0 @4 8100  sum @2 8100 @4  sum @0 @3 8100  sum @1 8100 @3  sum @10 0 @9  prod @13 3 4  prod @13 7 8  sum @11 @14 0  sum @11 @15 0  sum @4 @5 0  sum @9 @5 0  sum @10 @5 0  sum @11 @5 0  sum @12 @5 0  sum @16 @5 0  sum @17 @5 0</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>1350</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>0, @4; @0, @4; @3, 21600; @3, 2; 21600, @4; @1, @4; @3, @3, @0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@0, @0, @1, @2</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,topLeft&quot; switch=&quot;true&quot; xranges=&quot;0,5400&quot;</td>
</tr>
<tr>
<td>Limo</td>
<td>10800, 10800</td>
</tr>
</tbody>
</table>
Action Button: Return

*Internal Name:* ActionButtonBlank

*Shaped Concentric Fill:* Yes.

*Joins:* Mitered.

*Endcaps:* Rounded.

*Adjustments:* A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m,121600,21600,21600,121600,xem@0@0nfl@0@2@1@2@1@0xem,nfl@0@0em,21600nfl@0@2em21600,21600nfl@1@2em21600,nfl@1@0em@12@21nfl@23@9@3@21@24@21@24@20xy@3@19l@25@19qx@26@20l@26@21@11@21@11@20qy@25@10l@3@10qx@22@20l@22@21xe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guide Formulas</td>
<td>val #0; sum width 0 #0; sum height 0 #0; prod width 1 2; prod height 1 2; prod #0 1 2; prod #0 3 2; sum @1 @5 0; sum @2 @5 0; sum @0 @4 8100; sum @2 8100 @4; sum @0 @3 8100; sum @1 8100 @3; sum @10 0 @9; prod @13 7 8; prod @13 3 4; prod @13 5 8; prod @13 3 8; prod @13 1 4; sum @9 @15 0; sum @9 @16 0; sum @9 @18 0; sum @11 @14 0; sum @11 @15 0; sum @11 @16 0; sum @11 @17 0; sum @11 @18 0; sum @3 @5 0; sum @9 @5 0; sum @10 @5 0; sum @11 @5 0; sum @12 @5 0; sum @19 @5 0; sum @20 @5 0; sum @21 @5 0; sum @22 @5 0; sum @23 @5 0; sum @24 @5 0; sum @25 @5 0; sum @26 @5 0;</td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>1350</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>@0,0;@4;@0,0;@3,21600;@3,0;2;21600,4;@1,4;@3,0;@3,0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@0,0,0,0,0,0</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,topLeft&quot; switch=&quot;true&quot; xrange=&quot;0,5400</td>
</tr>
<tr>
<td>Limo</td>
<td>10800,10800</td>
</tr>
</tbody>
</table>
**Action Button: Document**

*Internal Name:* ActionButtonBlank

*Shaped Concentric Fill:* Yes.

*Joins:* Mitered.

*Endcaps:* Rounded.

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
<th>m.21600r21600.21600.xem@0@0mfl@0@2@1@2@1@0xem_mfl@0@0em ,21600mfl@0@0@2em21600,21600mfl@1@0@2em21600,mfl@1@0em@12@9mfl@12@10@13@10@13@14@15@9xem@15@9mfl@15@14@13@14e</th>
</tr>
</thead>
</table>
| Guide Formulas | val #0  
sum width 0 #0  
sum height 0 #0  
prod width 1 2  
prod height 1 2  
prod #0 1 2  
prod #0 3 2  
sum @1 @5 0  
sum @2 @5 0  
sum @0 @4 8100  
sum @2 8100 @4  
prod #0 3 4  
sum @3 @11 6075  
sum @3 6075 @11  
sum @4 @5 4050  
sum @13 @5 4050  
sum @9 @5 0  
sum @10 @5 0  
sum @12 @5 0  
sum @13 @5 0  
sum @14 @5 0  
sum @15 @5 0 |
| Adjustment Values | 1350 |
| Connector Locations | 0,0;0,4;@3,21600;@3,2,21600,0;4;@1,4;3,0;3,0 |
| Text Box Rectangle | @0,0,1,2 |
| Handles | position="#0,0";topLeft="true";xrange="0,5400" |
| Limo | 10800,10800 |
Action Button: Sound

*Internal Name:* ActionButtonBlank
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

| Path | m.121600r21600.21600.xem@0@0nfl@0@2@1@2@1@0xem afl@0@0em ,21600nfl@0@0@2em21600,21600nfl@1@2em21600,nfl@1@0em@11@21nfl @11@22@24@22@25@10@25@9@24@21xem@26@21nfl@12@20em@ 26@4nfl@12@4em@26@22nfl@12@23e |
| Guide Formulas | val #0  sum width 0 #0  sum height 0 #0  prod width 1 2  prod height 1 2  prod #0 1 2  prod #0 3 2  sum @1 5 0  sum @2 5 0  sum @0 4 8100  sum @2 8100 @4  sum @0 3 8100  sum @1 8100 @3  sum @10 0 @9  prod @13 1 8  prod @13 5 16  prod @13 5 8  prod @13 11 16  prod @13 3 4  prod @13 7 8  sum @9 14 0  sum @9 15 0  sum @9 17 0  sum @9 19 0  sum @11 15 0  sum @11 16 0  sum @11 18 0  sum @4 5 0  sum @9 5 0  sum @10 5 0  sum @11 5 0  sum @12 5 0  sum @20 5 0  sum @21 5 0  sum @22 5 0  sum @23 5 0  sum @24 5 0  sum @25 5 0  sum @26 5 0 |

| Adjustment Values | 1350 |
| Connector Locations | 0, @0; @0, @4; @3, 21600; @3, @2; 21600, @4; @1, @4; @3, 0 |
| Text Box Rectangle | @0, @0, @1, @2 |
| Handles | position="0,topLeft" switch="true" xrange="0,5400" |
| Limo | 10800, 10800 |

Action Button: Movie

*Internal Name:* ActionButtonBlank
*Shaped Concentric Fill:* Yes.
*Joins:* Mitered.
*Endcaps:* Rounded.

**Adjustments:** A top or left perimeter adjust handle controls the amount of bevel. It limo-stretches both vertically and horizontally at the midpoint, so that the amount of bevel is the same horizontally and vertically. Also, the
image on the button face stays in proportion and centered. The adjust handle switches between the top and left sides depending on which dimension is smaller. The adjust handle can extend half the distance of the smaller dimension. When a hyperlink or Action Setting (PowerPoint) is attached to this shape, it will change to the second form when depressed.

**Geometric properties:**

<table>
<thead>
<tr>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>m,l,21600r21600,l21600,xem@0@0@0@2@1@2@1@0xem,nfl@0@0@0em ,21600nfl@0@0@0em21600,nfl@1@2em21600,nfl@1@0em@1@1@39nfl @11@44@31@44@32@43@33@43@33@47@35@47@35@45@36@45@36@45@ 38@46@12@46@12@41@38@41@37@42@35@42@35@41@34@40@3 2@40@31@39xe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Guide Formulas</th>
<th>val #0</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum width 0 #0</td>
<td></td>
</tr>
<tr>
<td>sum height 0 #0</td>
<td></td>
</tr>
<tr>
<td>prod width 1 2</td>
<td></td>
</tr>
<tr>
<td>prod height 1 2</td>
<td></td>
</tr>
<tr>
<td>prod #0 1 2</td>
<td></td>
</tr>
<tr>
<td>prod #0 3 2</td>
<td></td>
</tr>
<tr>
<td>sum @1 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @2 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @0 @4 8100</td>
<td></td>
</tr>
<tr>
<td>sum @2 8100 @4</td>
<td></td>
</tr>
<tr>
<td>sum @0 @3 8100</td>
<td></td>
</tr>
<tr>
<td>sum @1 8100 @3</td>
<td></td>
</tr>
<tr>
<td>sum @10 @0 @9</td>
<td></td>
</tr>
<tr>
<td>prod @13 1455 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 1905 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 2325 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 16155</td>
<td></td>
</tr>
<tr>
<td>prod @13 19725 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 20595 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 5280 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 5730 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 6630 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 7492 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 9067 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 9555 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 13342 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 14580 21600</td>
<td></td>
</tr>
<tr>
<td>prod @13 15592 21600</td>
<td></td>
</tr>
<tr>
<td>sum @11 @14 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @15 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @16 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @17 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @18 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @19 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @20 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @21 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @22 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @23 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @24 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @25 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @26 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @27 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @28 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @29 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @30 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @31 0</td>
<td></td>
</tr>
<tr>
<td>sum @4 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @9 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @10 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @11 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @12 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @31 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @32 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @33 @5 0</td>
<td></td>
</tr>
<tr>
<td>sum @34 @5 0</td>
<td></td>
</tr>
<tr>
<td>Adjustment Values</td>
<td>1350</td>
</tr>
<tr>
<td>-------------------</td>
<td>------</td>
</tr>
<tr>
<td>Connector Locations</td>
<td>0, @2; @0, @4; @0, @2; 3, 21600; 2, 21600, @4, @2; @0, @4; @0, @2; 3, @0</td>
</tr>
<tr>
<td>Text Box Rectangle</td>
<td>@0, @0, @1, @2</td>
</tr>
<tr>
<td>Handles</td>
<td>position=&quot;#0,topLeft&quot; switch=&quot;true&quot; xrange=&quot;0,5400&quot;</td>
</tr>
<tr>
<td>Limo</td>
<td>10800, 10800</td>
</tr>
</tbody>
</table>
Appendix E: Animation Records

This appendix documents animation information records that can exist at the slide level of a document. For example, PowerPoint XP and later uses these records to store information about animations in presentations.

Many of these records represent the same type of information as an XML type, element, or attribute in the Office Open XML format. The publically available documentation for this format provides additional context and details for how animations are represented. Where appropriate, record descriptions contain references to their corresponding representation in the Office Open XML format.

F123 msofbtTimeNodeContainer

Same format as msofbtExtTimeNodeContainer.

F124 msofbtTimeConditionList

List of msofbtTimeCondition records.

The instance field of each condition record contains the condition type, with the values defined by the following enum:

```c
enum ConditionType {
    _NA,       // obsolete identifier; not used
    Begin,
    End,
    Next,
    Previous,
    EndSync,
    ConditionType_MaxEnumIDs
};
```

ConditionType indicates a type of condition that is equivalent to a condition type represented in the Office Open XML format:

<table>
<thead>
<tr>
<th>ConditionType</th>
<th>Office Open XML equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin</td>
<td>A “cond” in stCondLst</td>
</tr>
<tr>
<td>End</td>
<td>A “cond” in endCondLst</td>
</tr>
<tr>
<td>Next</td>
<td>A “cond” in nextCondLst</td>
</tr>
<tr>
<td>Previous</td>
<td>A “cond” in prevCondLst</td>
</tr>
<tr>
<td>EndSync</td>
<td>“endSync” element</td>
</tr>
</tbody>
</table>

F125 msofbtTimeConditionContainer

<table>
<thead>
<tr>
<th>Record</th>
<th>Type</th>
</tr>
</thead>
</table>
Base record | msofbtTimeCondition (FTC)
Visual element (optional) | msofbtClientVisualElement if triggerType == totVisualElement

F126 msofbtTimeModifierList
List of msofbtTimeModifier records.

F127 msofbtTimeNode (FTN)
Represents the same information as the CT_TLCommonTimeNodeData type in the Office Open XML format.

```c
// FTN - File Time Node
struct FTN {
    ULONG masterID; // OBSOLETE: id of the master time node in the timing tree for a "subordinate" time node.
    ULONG restart;  // TLTimeNodeRestartType
    ULONG type;     // TLTimeNodeType
    ULONG fill;     // TLTimeNodeFillType
    ULONG syncBehavior; // TLTimeNodeSyncType
    BYTE fSyncMaster; // OBSOLETE: set to zero
    LONG duration;   // TIME, in milliseconds
    ULONG propertiesSet; // bit flag of which properties are used/set, as defined below
};
```

```c
eenum TLTimeNodeType {
    TLTimeNodeTypeParallel = 0,
    TLTimeNodeTypeSequential,
    TLTimeNodeTypeExclusive,
    TLTimeNodeTypeBehaviorType,
    TLTimeNodeTypeMediaType,
    TLTimeNodeType_MaxEnumIDs
};
```

```c
eenum TLTimeNodeRestartType {
    TLR_NoRestartType = 0,
    TLR_AlwaysRestart = 1,
    TLR_RestartWhenOff = 2,
    TLR_NeverRestart = 3,
    TLRTimeNodeRestartType_MaxEnumIDs = 4
};
```

```c
eenum TLTimeNodeFillType {
    TLF_NoFillType = 0,
    TLF_FillRemove = 1,
    TLF_FillFreeze = 2,
```
TLF_FillHold = 3,
TLF_FillTransition = 4,
TLTimeNodeFillType_MaxEnumIDs = 5
};

enum TLTimeNodeSyncType
{
    TLS_NoSyncType = 0,
    TLS_CanSlipSyncType = 1,
    TLS_LockedSyncType = 2,
    TLTimeNodeSyncType_MaxEnumIDs = 3
};

// property bit flag for propertiesSet
const int fillProperty         = (1 << 0);
const int restartProperty      = (1 << 1);
const int syncBehaviorProperty = (1 << 2);
const int groupingTypeProperty = (1 << 3);
const int durationProperty     = (1 << 4);

F128 msobufTTimeCondition (FTC)

Represents the same information as the CT_TLTimeCondition type in the Office Open XML format.
The instance field of the record contains the condition type, the same as described by the
"ConditionType" for record F124.

// FTC - File Time Condition
struct FTC
{
    ULONG triggerType;          // TriggerObjectType
    ULONG event;                // TriggerEventType
    ULONG id;                   // (1) RuntimeNodeReferenceType if triggerType ==
totRuntimeNodeRef; (2) index of timenode in tree where index is computed by depth-first
    // traversal of the tree
    LONG delay;                // TIME, in milliseconds
};

{  
totNone,
totVisualElement,
totTimeNode,
totRuntimeNodeRef,
    TriggerObjectType_MaxEnumIDs
};

enum TriggerEventType
{
    tetNone = 0,
tetOnBegin,
tetOnEnd,
tetBegin,
tetEnd,
tetOnClick,  
tetOnDoubleClick,  
tetOnMouseOver,  
tetOnMouseOut,  
tetOnNext,       // PPT-specific  
tetOnPrev,       // PPT-specific  
tetOnStopAudio,  
TriggerEventType_MaxEnumIDs
};

enum RuntimeNodeReferenceType
{
    rnrtWithFirstChild,  
rnrtWithLastChild,  
rnrtAllChildren,      
    RuntimeNodeReferenceType_MaxEnumIDs
};

F129  msofbtTimeModifier (FTM)

Time modifiers for a time node, used in a msofbtExtTimeNodeContainer.

The instance field of each record tells you the Modifier type:

enum Type
{
    RepeatCount = 0,    // The number of times to repeat  
    RepeatDur,          // Duration of repeats  
    Speed,              // Fraction indicating how much faster  
                         // or slower to go than the normal duration.  
    Accelerate,         // Fraction between 0 and 1 indicating  
                         // portion of the duration over which to accelerate.  
    Decelerate,         // Fraction between 0 and 1 indicating portion of the  
                         // duration over which to decelerate.  
    AutoReverse,        // The node should play forward then reverse.  
    Type_MaxEnumIDs
};

// FTM - File Time Modifier  
struct FTM
{
    ULONG  type;          // Type, as described above  
    ULONG  value;         // a float, written out as a ULONG
};

F12A  msofbtTimeBehaviorContainer

Represents the same information as the CT_TLCommonBehaviorData type in the Office Open XML format.

<table>
<thead>
<tr>
<th>Record</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Base record msofbtTimeBehavior FTB
Optional attribute names msofbtTimeVariantList (should all be strings)
optional additional properties msofbtTimePropertyList
optional "FROM" msofbtTimeVariant
optional "TO" msofbtTimeVariant
optional "BY" msofbtTimeVariant
Optional target element msofbtClientVisualElement

F12B msofbtTimeAnimateBehaviorContainer
Represents the same information as the CT_TLAnimateBehavior type in the Office Open XML format.

<table>
<thead>
<tr>
<th>Record</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base record</td>
<td>msofbtTimeAnimateBehavior (FTBA)</td>
</tr>
<tr>
<td>Optional animation value list</td>
<td>msofbtTimeAnimationValueList</td>
</tr>
<tr>
<td>&quot;BY&quot;</td>
<td>msofbtTimeVariant (instance 1)</td>
</tr>
<tr>
<td>&quot;FROM&quot;</td>
<td>msofbtTimeVariant (instance 2)</td>
</tr>
<tr>
<td>&quot;TO&quot;</td>
<td>msofbtTimeVariant (instance 3)</td>
</tr>
<tr>
<td>Base time behavior record</td>
<td>msofbtTimeBehaviorContainer (FTB)</td>
</tr>
</tbody>
</table>

F12C msofbtTimeColorBehaviorContainer
Represents the same information as the CT_TLAnimateColorBehavior type in the Office Open XML format.

| Base record | msofbtTimeColorBehavior (FTBC) |
| Base time behavior record | msofbtTimeBehaviorContainer (FTB) |

F12D msofbtTimeEffectBehaviorContainer
Represents the same information as the CT_TLAnimateEffectBehavior type in the Office Open XML format.
<table>
<thead>
<tr>
<th>Base record</th>
<th>msofbtTimeColorBehavior (FTBE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>msofbtTimeVariant (instance 1)</td>
</tr>
<tr>
<td>Progress</td>
<td>msofbtTimeVariant (instance 2)</td>
</tr>
<tr>
<td>Runtime context</td>
<td>msofbtTimeVariant (instance 3)</td>
</tr>
<tr>
<td>Base time behavior record</td>
<td>msofbtTimeBehaviorContainer (FTB)</td>
</tr>
</tbody>
</table>

**F12E**

msofbtTimeMotionBehaviorContainer

Represents the same information as the CT_TLAnimateMotionBehavior type in the Office Open XML format.

<table>
<thead>
<tr>
<th>Base record</th>
<th>msofbtTimeColorBehavior (FTBM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path</td>
<td>msofbtTimeVariant (instance 1)</td>
</tr>
<tr>
<td>Rotation</td>
<td>msofbtTimeVariant (instance 2)</td>
</tr>
<tr>
<td>Base time behavior record</td>
<td>msofbtTimeBehaviorContainer (FTB)</td>
</tr>
</tbody>
</table>

**F12F**

msofbtTimeRotationBehaviorContainer

Represents the same information as the CT_TLAnimateRotationBehavior type in the Office Open XML format.

<table>
<thead>
<tr>
<th>Base record</th>
<th>msofbtTimeRotationBehavior (FTBR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base time behavior record</td>
<td>msofbtTimeBehaviorContainer (FTB)</td>
</tr>
</tbody>
</table>

**F130**

msofbtTimeScaleBehaviorContainer

Represents the same information as the CT_TLAnimateScaleBehavior type in the Office Open XML format.

<table>
<thead>
<tr>
<th>Base record</th>
<th>msofbtTimeScaleBehavior (FTBS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base time behavior record</td>
<td>msofbtTimeBehaviorContainer (FTB)</td>
</tr>
</tbody>
</table>
F131 msofbtTimeSetBehaviorContainer  
Represents the same information as the CT_TLSetBehavior type in the Office Open XML format.

<table>
<thead>
<tr>
<th>Base record</th>
<th>msofbtTimeSetBehavior (FTBSet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>“To”</td>
<td>msofbtTimeVariant (instance 1)</td>
</tr>
<tr>
<td>Base time behavior record</td>
<td>msofbtTimeBehaviorContainer (FTB)</td>
</tr>
</tbody>
</table>

F132 msofbtTimeCommandBehaviorContainer  
Represents the same information as the CT_TLCommandBehavior type in the Office Open XML format.

<table>
<thead>
<tr>
<th>Base record</th>
<th>msofbtTimeCommandBehavior (FTBCom)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>msofbtTimeVariant (instance 1)</td>
</tr>
<tr>
<td>Base time behavior record</td>
<td>msofbtTimeBehaviorContainer (FTB)</td>
</tr>
</tbody>
</table>

F133 msofbtTimeBehavior (FTB)  
Represents part of the same information as the CT_TLBehavior type in the Office Open XML format.

```c
// FTB - File Time Behavior
struct FTB
{
    ULONG propertiesUsed; // bit flag of properties set, as defined below by the PropertyUsedFlag
    UINT tbaddAdditive;   // Additive
    UINT tbaccAccumulate; // Accumulate
    UINT tbttTransformType; // TransformType
};
```

```c
enum PropertyUsedFlag
{
    PUF_NONE             = 0,
    PUF_Additive         = 1 << 0,
    PUF_Accumulate       = 1 << 1,
    PUF_AttributeNames   = 1 << 2,
    PUF_TransformType    = 1 << 3,
    PUF_FromFormula      = 1 << 4,
    PUF_ToFormula        = 1 << 5,
    PUF_ByFormula        = 1 << 6
};
```

```c
enum Additive
```
F134 msofbtTimeAnimateBehavior (FTBA)

Represents part of the same information as the CT_TLAnimateBehavior type in the Office Open XML format.

// FTBA - File Time Behavior
struct FTBA
{
    UINT     tabcmCalcMode;   // CalcMode
    ULONG    propertiesUsed;  // AnimatePropertyUsedFlag
    UINT     tabvtValueType;  // ValueType
};

eenum CalcMode
{
    DiscreteMode,
    LinearMode,
    FormulaMode,
    CalcMode_MaxEnumIDs
};

eenum AnimatePropertyUsedFlag
{
    APUF_NONE          = 0,
    APUF_By            = 1 << 0,
    APUF_From          = 1 << 1,
    APUF_To            = 1 << 2,
    APUF_CalcMode      = 1 << 3,
APUF_AnimationValues = 1 << 4,
APUF_ValueType = 1 << 5
};

enum ValueType
{
    StringType,
    NumberType,
    ColorType,
    ValueType_MaxEnumIDs
};

F135 msofbtTimeColorBehavior (FTBC)

Represents part of the same information as the CT_TLAnimateColorBehavior type in the Office Open XML format.

// FTBC - File Time Behavior
struct FTBC
{
    ULONG       propertiesUsed;  // AnimateColorPropertyUsedFlag
    UINT        tabcmBy;         // ColorModel
    UINT        nColor1By;       // range is -255 to 255; could be R or H depending on color model specified by tabcmBy
    UINT        nColor2By;       // range is -255 to 255; G or S depending on tabcmBy
    UINT        nColor3By;       // range is -255 to 255; B or L depending on tabcmBy
    UINT        tabcmFrom;       // ColorModel
    UINT        nColor1From;     // range is 0 to 255
    UINT        nColor2From;     // range is 0 to 255
    UINT        nColor3From;     // range is 0 to 255
    UINT        tabcmTo;         // ColorModel
    UINT        nColor1To;       // range is 0 to 255
    UINT        nColor2To;       // range is 0 to 255
    UINT        nColor3To;       // range is 0 to 255
};

enum AnimateColorPropertyUsedFlag
{
    ACPUF_NONE = 0,
    ACPUF_By   = 1 << 0,
    ACPUF_From = 1 << 1,
    ACPUF_To   = 1 << 2,
    ACPUF_ColorSpace = 1 << 3,
    ACPUF_Direction = 1 << 4
};

enum ColorModel
{
    RGBColorModel   = 0,
    HSLColorModel   = 1,
    IndexColorModel = 2,
    ColorModel_MaxEnumIDs
};
F136 msofbtTimeEffectBehavior (FTBE)

Represents part of the same information as the CT_TLAnimateEffectBehavior type in the Office Open XML format.

// FTBE - File Time Behavior
struct FTBE
{
    ULONG propertiesUsed;  // AnimateEffectPropertyUsedFlag
    UINT taetTransition;  // Transition
};

enum AnimateEffectPropertyUsedFlag
{
    AEPUF_NONE                    = 0,
    AEPUF_Transition              = 1 << 0,
    AEPUF_Type                    = 1 << 1,
    AEPUF_Progress                = 1 << 2,
    AEPUF_RuntimeContext_Obsolete = 1 << 3 // Keep this for backward binary persistence
};

enum Transition
{
    TransitionIn,
    TransitionOut,
    TransitionNone,
    Transition_MaxEnumIDs
};

F137 msofbtTimeMotionBehavior (FTBM)

Represents part of the same information as the CT_TLAnimateMotionBehavior type in the Office Open XML format.

// FTBM - File Time Behavior Motion
struct FTBM
{
    ULONG propertiesUsed;  // AnimateMotionPropertyUsedFlag
    ULONG fXBy;            // a float, written out as a ULONG
    ULONG fYBy;            // a float, written out as a ULONG
    ULONG fXFrom;          // a float, written out as a ULONG
    ULONG fYFrom;          // a float, written out as a ULONG
    ULONG fXTo;            // a float, written out as a ULONG
    ULONG fYTo;            // a float, written out as a ULONG
    UINT tamboOrigin;      // Origin
};

enum AnimateMotionPropertyUsedFlag
{  
    AMPUF_NONE = 0,  
    AMPUF_By = 1 << 0,  
    AMPUF_From = 1 << 1,  
    AMPUF_To = 1 << 2,  
    AMPUF_Origin = 1 << 3,  
    AMPUF_Paths = 1 << 4,  
    AMPUF_Rotation = 1 << 5,  
    AMPUF_EditRotation = 1 << 6,  
    AMPUF_PointsTypes = 1 << 7  
};

enum Origin
{
    DefaultOrigin,  
    ParentOrigin,  
    LayoutOrigin,  
    Origin_MaxEnumIDs
};

F138 msOfficeBinaryFormat Extension (FTBR)

Represents part of the same information as the CT_TLAnimateRotationBehavior type in the Office Open XML format.

// FTBR - File Time Behavior Rotation
struct FTBR
{
    ULONG propertiesUsed; // AnimateRotationPropertyUsedFlag
    ULONG fBy; // a float, written out as a ULONG
    ULONG fFrom; // a float, written out as a ULONG
    ULONG fTo; // a float, written out as a ULONG
    UINT tabrd; // Direction
};

enum AnimateRotationPropertyUsedFlag
{
    ARPUF_NONE = 0,  
    ARPUF_By = 1 << 0,  
    ARPUF_From = 1 << 1,  
    ARPUF_To = 1 << 2,  
    ARPUF_Direction = 1 << 3  
};

enum Direction
{
    Clockwise,  
    CounterClockwise,  
    Direction_MaxEnumIDs
};
F139 msobtTimeScaleBehavior (FTBS)

Represents part of the same information as the CT_TLAnimateScaleBehavior type in the Office Open XML format.

// FTBS - File Time Behavior Scale
struct FTBS
{
  ULONG propertiesUsed;    // AnimateScalePropertyUsedFlag
  ULONG fXBy;              // a float, written out as a ULONG
  ULONG fYBy;              // a float, written out as a ULONG
  ULONG fXFrom;            // a float, written out as a ULONG
  ULONG fYFrom;            // a float, written out as a ULONG
  ULONG fXTo;              // a float, written out as a ULONG
  ULONG fYTo;              // a float, written out as a ULONG
  BYTE fZoomContents;      // true or false
};

enum AnimateScalePropertyUsedFlag
{
  ASPUF_NONE                = 0,
  ASPUF_By                  = 1 << 0,
  ASPUF_From                = 1 << 1,
  ASPUF_To                  = 1 << 2,
  ASPUF_ZoomContents        = 1 << 3,
};

F13A msobtTimeSetBehavior (FTBSet)

Represents part of the same information as the CT_TLSetBehavior type in the Office Open XML format.

// FTBSet - File Time Behavior Set
struct FTBSet
{
  ULONG propertiesUsed;    // SetPropertyUsedFlag
  UINT tabvtValueType;     // ValueType
};

enum SetPropertyUsedFlag
{
  SPUF_NONE                = 0,
  SPUF_To                  = 1 << 0,
  SPUF_ValueType           = 1 << 1
};

enum ValueType
{
  StringType,
  NumberType,
  ColorType,
  ValueType_MaxEnumIDs
};
F13B msisoftTimeCommandBehavior (FTBCom)

Represents part of the same information as the CT_TLCommandBehavior type in the Office Open XML format.

```c
// FTBCom - File Time Behavior Command
struct FTBCom
{
    ULONG    propertiesUsed;     // CommandPropertyUsedFlag
    UINT     tcbtType;           // CommandType
};
```

```c
enum CommandPropertyUsedFlag
{
    CPUF_NONE                = 0,
    CPUF_Type                = 1 << 0,
    CPUF_Command             = 1 << 1
};
```

```c
enum CommandType
{
    EventType,
    CallType,
    OlVerbType,
    CommandType_MaxEnumIDs
};
```

F13C msisoftClientVisualElement

msisoftClientVisualElement is a container for either (1) PSR_VisualPageAtom or (2) PSR_VisualShapeAtom defined in serial.h. The PSR_VisualXXX records are documented in the PowerPoint file format documentation.

F13D msisoftTimePropertyList

A list of TimeVariant objects, defined by msisoftTimeVariant records. The instance field for each TimeVariant record gives you the property ID for the property.

```c
enum TLTimePropertyID
{
    tpidUnknown             = 0,
    tpidID                  = 1, // string
    tpidDisplay             = 2,
    tpidMasterPos           = 5,
    tpidSubNodeType         = 6,
    tpidParagraphLevel      = 7,
    tpidGraphLevel          = 8,
```
tpidEffectID = 9,
tpidEffectDir = 10,
tpidEffectType = 11,
tpidAfterEffect = 13,
tpidDiagramLevel = 14,
tpidSlideCount = 15, // integer
tpidTimeFilter = 16, // string
tpidEventFilter = 17, // string
tpidHideWhenStopped = 18, // boolean
tpidGroupID = 19,
tpidPPTType = 20,
tpidPlaceholderNode = 21,
tpidMediaVolume = 22,
tpidMediaMute = 23,
tpidXMLUnknownAttribs = 24,
tpidXMLAttributesUnknownValues = 25,
tpidZoomToFullScreen = 26,
tpidShowControls = 27,
tpidDVDTitle = 28,
tpidDVDStartTime = 29,
tpidDVDEndTime = 30
};

F13E msofbtTimeVariantList

A list of msofbtTimeVariant records. Each record is a string value that is an attribute name that indicates an attribute that the Behavior modifies. The possible attribute names are the same as those listed in the Office Open XML Format for the “attrName” element.

F13F msofbtTimeAnimationValueList

A list of msofbtTimeAnimationValue records. Represents the same information as the CT_TLTimeAnimateValueList in the Office Open XML Format.

F140 msofbtTimeIterateData

FTID

Represents the same information as the CT_TLIterateData in the Office Open XML Format.

// FTID - File Time Iterate Data
struct FTID
{
    ULONG interval; // a float, written out as a ULONG
    ULONG type;     // IterationType, defined below
    ULONG direction; // Direction, defined below
    ULONG intervaltype; // IntervalType, defined below
    ULONG used;     // PropertyUsed, defined below
};

eenum IterationType
{
    AllAtOnce,
    ByWord,
ByLetter,
   IntervalType_MaxEnumIDs
\
    enum IntervalType
    {   Seconds,
        Percentage,
        IntervalType_MaxEnumIDs
    };

    enum Direction
    {   Backwards,
        Forwards,
        Direction_MaxEnumIDs
    };

    enum PropertyUsed
    {   DirectionProperty      = 1 << 0,
        IterationTypeProperty  = 1 << 1,
        IntervalProperty       = 1 << 2,
        IntervalTypeProperty   = 1 << 3,
    };

    F141 msobftTimeSequenceData FTSD

    Represents part of the same information as the CT_TLTimeNodeSequence in the Office Open XML Format.

    // FTSD - File Time Sequence Data
    struct FTSD
    {
        ULONG concurrency;           // ConcurrencyType
        ULONG nextAction;            // NextActionType
        ULONG previousAction;        // PreviousActionType
        ULONG enableNext;            // EnableNextType
        ULONG used;                  // see "TLTimeSequenceData" flags below
    };

    enum ConcurrencyType
    {   Disabled,
        Enabled,
        ConcurrencyType_MaxEnumIDs
    };

    enum NextActionType
    {   NoNextActionType,
        Seek,
NextActionType_MaxEnumIDs
{
  NoPreviousActionType,
  SkipTimed,
  PreviousActionType_MaxEnumIDs
};

enum PreviousActionType
{
  NoPreviousActionType,
  SkipTimed,
  PreviousActionType_MaxEnumIDs
};

enum EnableNextType
{
  End,
  Begin,
  EnableNextType_MaxEnumIDs
};

// TLTimeSequenceData flags:
const int SF_Concurrency    = (1 << 0);
const int SF_NextAction     = (1 << 1);
const int SF_PreviousAction = (1 << 2);
const int SF_EnableNext     = (1 << 3);

F142 msofbtTimeVariant

Represents the same information as the CT_TLAnimVariant in the Office Open XML Format.

<table>
<thead>
<tr>
<th>Type</th>
<th>Byte – values defined by AnimVariantType, defined below</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>bool, int, float, or string depending on the Type</td>
</tr>
</tbody>
</table>

denum Type
{
  None     = -1,
  Bool     = 0,
  Int      = 1,
  Float    = 2,
  String   = 3,
};

F143 msofbtTimeAnimationValue

Represents the same information as the CT_TLTimeAnimateValue in the Office Open XML Format.

<table>
<thead>
<tr>
<th>Time, in milliseconds</th>
<th>LONG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>msofbtTimeVariant</td>
</tr>
<tr>
<td>Formula</td>
<td>string</td>
</tr>
</tbody>
</table>
F144  msofbtExtTimeNodeContainer

Represents the same information as a CT_TLTimeNodeSequence, CT_TLMediaNodeAudio, or CT_TLMediaNodeVideo type in the Office Open XML Format.

<table>
<thead>
<tr>
<th>Base record</th>
<th>msofbtTimeNode (FTN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior (optional)</td>
<td>Any of the msofbtTimeXXXBehavior records</td>
</tr>
<tr>
<td>Media (optional)</td>
<td>msofbtClientVisualElement</td>
</tr>
<tr>
<td>Iterate data (optional)</td>
<td>msofbtTimeIterateData</td>
</tr>
<tr>
<td>Sequence data (optional)</td>
<td>msofbtTimeSequenceData</td>
</tr>
<tr>
<td>Begin condition list</td>
<td>msofbtTimeConditionList</td>
</tr>
<tr>
<td>End condition list</td>
<td>msofbtTimeCondition</td>
</tr>
<tr>
<td>End sync</td>
<td>msofbtTimeConditionList</td>
</tr>
<tr>
<td>Time modifiers</td>
<td>msofbtTimeModifierList</td>
</tr>
<tr>
<td>Subordinate time nodes (0-n occurences)</td>
<td>msofbtSubNodeContainer</td>
</tr>
<tr>
<td>Child time nodes (0-n occurences)</td>
<td>msofbtExtTimeNodeContainer</td>
</tr>
</tbody>
</table>

F145  msofbtSubNodeContainer

Same format as msofbtExtTimeNodeContainer. This represents the same information as the “subTnLst” element of the CT_TLCommonTimeNodeData type in the Office Open XML Format.
Appendix F: Shape XML Data

The 2007 Microsoft Office System for Windows introduced a new XML-based file format. This format is fully documented in the publicly available Office Open XML documentation. While this is the default format for documents saved by Office 2007, Office 2007 also provides the capability to save files to the binary file format used in previous versions and described by this document.

Some new shape properties and property groups were added to the binary file format in Office 2007, and those properties are documented above. In addition, some XML information from the Office Open XML format is embedded in binary files. This is done to preserve data in the file, so that a file saved by Office 2007 to the binary format and later reopened in Office 2007 will retain document features available only in Office 2007 and later.

The shape property metroBlob (PID 937) stores a package in Office Open XML format. As described in the Office Open XML format, this data is equivalent to a zip file. In addition to storing the shape’s Office Open XML representation, with root element “sp”, the package may contain a second file named downrev.xml.

The downrev.xml file stores additional information about how the shape was converted from its native XML representation to the older binary representation. This information allows Office 2007 to determine what properties of the shape were changed when the file was saved by a previous version of Office. When re-opening the file in Office 2007, the application uses this information to determine whether to read the binary or XML representation of the shape. For example, a shape that was moved but not otherwise changed in Office 2003 could be rendered in Office 2007 using the full-fidelity XML data, but with adjusted coordinates.

This file does not store information affecting the appearance or meaning of a shape; it is used solely as a mechanism for converting shape information between the XML and binary representations. The following section provides the schema for this file and description of the relevant types, elements, and attributes.

Schema for downrev.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<!---------------------------------------------------------------------
 Module : oartdownrev.xsd
 Owner : [OFFICE]

 Copyright (c) Microsoft Corporation. All rights reserved.

---------------------------------------------------------------------
 Schema definition for down-rev roundtriping
--------------------------------------------------------------------->

<xsd:schema
 targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/main"
 elementFormDefault="qualified"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns="http://schemas.openxmlformats.org/drawingml/2006/main">

<!---- Schema includes ------------------------------------------------->
<xsd:include schemaLocation="oartbasetypes.xsd"/>
```
<xsd:complexType name="CT_BoundRect">
    <xsd:attribute name="l" type="ST_Coordinate" use="required"/>
    <xsd:attribute name="t" type="ST_Coordinate" use="required"/>
    <xsd:attribute name="r" type="ST_Coordinate" use="required"/>
    <xsd:attribute name="b" type="ST_Coordinate" use="required"/>
</xsd:complexType>

<xsd:complexType name="CT_PreciseRelativeRect">
    <xsd:sequence>
        <xsd:element name="l" type="CT_Ratio" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="t" type="CT_Ratio" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="r" type="CT_Ratio" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="b" type="CT_Ratio" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CT_XLLegacyAnchorPoint">
    <xsd:attribute name="row" type="xsd:int" use="required"/>
    <xsd:attribute name="col" type="xsd:int" use="required"/>
    <xsd:attribute name="rowOffset" type="ST_Coordinate" use="required"/>
    <xsd:attribute name="colOffset" type="ST_Coordinate" use="required"/>
</xsd:complexType>

<xsd:complexType name="CT_XLLegacyAnchor">
    <xsd:sequence>
        <xsd:element name="from" type="CT_XLLegacyAnchorPoint" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="to" type="CT_XLLegacyAnchorPoint" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="fMove" type="xsd:boolean" use="optional" default="false"/>
    <xsd:attribute name="fResize" type="xsd:boolean" use="optional" default="false"/>
</xsd:complexType>

<xsd:complexType name="CT_DownRevStg">
    <xsd:sequence>
        <xsd:element name="from" type="CT_XLLegacyAnchorPoint" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="to" type="CT_XLLegacyAnchorPoint" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="fMove" type="xsd:boolean" use="optional" default="false"/>
    <xsd:attribute name="fResize" type="xsd:boolean" use="optional" default="false"/>
</xsd:complexType>
Description of Types in Schema

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT_BoundsRect</td>
<td>Represents the transform bounds rectangle.</td>
</tr>
<tr>
<td>CT_PreciseRelativeRect</td>
<td>Represents a rectangle by the ratio of the offsets from the respective sides of some other rectangle. Left and right are the ratio of the rectangle width, and top and bottom are the ratio of the rectangle height. This is similar to RelativeRect, except we use Ratio instead of Percentage, so there won't be data loss after roundtrip.</td>
</tr>
<tr>
<td>CT_XLLegacyAnchorPoint</td>
<td>Represents the legacy Excel anchor point.</td>
</tr>
<tr>
<td>CT_XLLegacyAnchor</td>
<td>Represents the legacy Excel anchor.</td>
</tr>
<tr>
<td>CT_DownRevStg</td>
<td>Represents all information about how a shape was saved down-level.</td>
</tr>
</tbody>
</table>

**Elements**

- **bounds**: Represents the transform bounds of the shape.
- **effectOffsets**: Obsolete; no longer written.
- **preciseEffectOffsets**: Represents shape effect offsets.
- **xlAnchor**: Represents shape anchoring if the shape is in an Excel document.

**Attributes**

- **shapeCheckSum**: Checksum computed on the shape. If the checksum doesn’t match the binary data when the file is loaded back into Office 2007, the XML data is considered out of date and the binary representation is used.
- **textCheckSum**: Checksum computed on the text of the shape. If the checksum doesn’t match the binary data when the file is loaded back into Office 2007, the XML data is considered out of date and the binary representation of the text is used.
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>shapeId</td>
<td>Stores the shape ID; useful if the shape’s full XML is not included.</td>
</tr>
<tr>
<td>fHybridRaster</td>
<td>Flag which is set on the group when rasterizing a shape into a legacy group of a picture and a textbox.</td>
</tr>
<tr>
<td>rot</td>
<td>Rotation of the shape; used to detect whether the shape’s rotation has changed since this property is not computed in the shape checksum.</td>
</tr>
<tr>
<td>convChart</td>
<td>Whether a native chart converted to an OLE object when saved by Office 2007 should be converted back to a native object when reopened by Office 2007.</td>
</tr>
</tbody>
</table>

**Schema for oartbasetypes.xsd**

```xml
<?xml version="1.0" encoding="utf-8"?>
<!---------------------------------------------------------------------
 Module : oartbasetypes.xsd
 Owner   : [OFFICE]
 Copyright (c) Microsoft Corporation. All rights reserved.
--------------------------------------------------------------------->

<xsd:schema targetNamespace="http://schemas.openxmlformats.org/drawingml/2006/main" elementFormDefault="qualified"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns="http://schemas.openxmlformats.org/drawingml/2006/main">
 <xsd:simpleType name="ST_Coordinate">
  <xsd:annotation>
   <xsd:documentation>Coordinate</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:long">
   <xsd:minInclusive value="-27273042329600" /> 
   <xsd:maxInclusive value="27273042316900" /> 
  </xsd:restriction>
 </xsd:simpleType>
 <xsd:simpleType name="ST_Percentage">
  <xsd:annotation>
   <xsd:documentation>Percentage</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:int" />
 </xsd:simpleType>
 <xsd:complexType name="CT_Ratio">
  <xsd:attribute name="n" type="xsd:long" use="required">
   <xsd:annotation>
    <xsd:documentation>Numerator</xsd:documentation>
   </xsd:annotation>
  </xsd:attribute>
 </xsd:complexType>
</xsd:schema>
```
<xsd:complexType name="CT_RelativeRect">
  <xsd:attribute name="l" type="ST_Percentage" use="optional" default="0">
    <xsd:annotation>
      <xsd:documentation>Left Offset</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="t" type="ST_Percentage" use="optional" default="0">
    <xsd:annotation>
      <xsd:documentation>Top Offset</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="r" type="ST_Percentage" use="optional" default="0">
    <xsd:annotation>
      <xsd:documentation>Right Offset</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="b" type="ST_Percentage" use="optional" default="0">
    <xsd:annotation>
      <xsd:documentation>Bottom Offset</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
Appendix G: Miscellaneous Enumerated Types and Structures

// MSOSHAPEPATH
typedef enum
{
    msoshapeLines,    // A line of straight segments
    msoshapeLinesClosed, // A closed polygonal object
    msoshapeCurves,   // A line of Bezier curve segments
    msoshapeCurvesClosed, // A closed shape with curved edges
    msoshapeComplex,  // pSegmentInfo must be non-empty
} MSOSHAPEPATH;

// MSOWRAPMODE
typedef enum
{
    msowrapSquare,
    msowrapByPoints,
    msowrapNone,
    msowrapTopBottom,
    msowrapThrough,
} MSOWRAPMODE;

// MSOBWMODE
typedef enum
{
    msobwColor,       // only used for predefined shades
    msobwAutomatic,   // depends on object type
    msobwGrayScale,   // shades of gray only
    msobwLightGrayScale, // shades of light gray only
    msobwInverseGray, // dark gray mapped to light gray, etc.
    msobwGrayOutline, // pure gray and white
    msobwBlackTextLine, // black text and lines, all else grayscale
    msobwHighContrast, // pure black and white mode (no grays)
    msobwBlack,       // solid black
    msobwWhite,       // solid white
    msobwDontShow,    // object not drawn
    msobwNumModes     // number of Black and white modes
} MSOBWMODE;

// MSOANCHOR
typedef enum
{
    msoanchorTop,
    msoanchorMiddle,
    msoanchorBottom,
    msoanchorTopCentered,
    msoanchorMiddleCentered,
    msoanchorBottomCentered,
    msoanchorTopBaseline,
    msoanchorBottomBaseline,
    msoanchorTopCenteredBaseline,
}
typedef enum 
{
    msocdir0, // Right
    msocdir90, // Down
    msocdir180, // Left
    msocdir270 // Up
} MSOCDIR;

// MSOCXSTYLE - connector style
typedef enum 
{
    msocxstyleStraight = 0,
    msocxstyleBent,
    msocxstyleCurved,
    msocxstyleNone
} MSOCXSTYLE;

// MSOCXX - the kind of connection sites
typedef enum 
{
    msocxkNone = 0, // No connection sites
    msocxkSegments = 1, // Connection sites at the segments
    msocxkCustom = 2, // Sites defined by msopidPConnectionSites
    msocxkRect = 3, // Use the connection sites for a rectangle
} MSOCXK;

// MSOTXFL - text flow
typedef enum 
{
    msotxflHorzN, // Horizontal non-@
    msotxflTtoBA, // Top to Bottom @-font
    msotxflBtoT, // Bottom to Top non-@
    msotxflTtoBN, // Top to Bottom non-@ 
    msotxflHorzA, // Horizontal @-font
    msotxflVertN, // Vertical, non-@
} MSOTXFL;

// MSOTXDIR - text direction (needed for Bi-Di support)
typedef enum 
{
    msotxdirLTR, // left-to-right text direction
    msotxdirRTL, // right-to-left text direction
    msotxdirContext, // context text direction
} MSOTXDIR;

// MSOSPCOT - Callout Type
typedef enum 
{
    msospcotRightAngle = 1,
    msospcotOneSegment = 2,
typedef enum
{
    msoalignTextStretch,  /* Stretch each line of text to fit width. */
    msoalignTextCenter,  /* Center text on width. */
    msoalignTextLeft,    /* Left justify. */
    msoalignTextRight,   /* Right justify. */
    msoalignTextLetterJust, /* Spread letters out to fit width. */
    msoalignTextWordJust, /* Spread words out to fit width. */
    msoalignTextInvalid   /* Invalid */
} MSOGEOTEXTALIGN;

typedef enum
{
    msoblipflagDefault = 0,
    msoblipflagComment = 0, // Blip name is a comment
    msoblipflagFile,       // Blip name is a file name
    msoblipflagURL,        // Blip name is a full URL
    msoblipflagType = 3,   // Mask to extract type
    /* Or the following flags with any of the above. */
    msoblipflagDontSave    = 4, // A "dont" is the depression in the metal
    msoblipflagDoNotSave   = 4, // For those who prefer English
    msoblipflagLinkToFile  = 8,
} MSOBLIPFLAGS;
---

```c
// MSO3DRENDERMODE
typedef enum
{
msoFullRender,       // Generate a full rendering
msoWireframe,       // Generate a wireframe
msoBoundingCube,    // Generate a bounding cube
} MSO3DRENDERMODE;

// MSOXFORMTYPE
typedef enum
{
msoxformAbsolute,   // Apply transform in absolute space centered on shape
msoxformShape,      // Apply transform to shape geometry
msoxformDrawing     // Apply transform in drawing space
} MSOXFORMTYPE;

// MSOSHADOWTYPE
typedef enum
{
msoshadowOffset,    // N pixel offset shadow
msoshadowDouble,    // Use second offset too
msoshadowRich,      // Rich perspective shadow (cast relative to shape)
msoshadowShape,     // Rich perspective shadow (cast in shape space)
msoshadowDrawing,  // Perspective shadow cast in drawing space
msoshadowEmbossOrEngrave,
} MSOSHADOWTYPE;

// MSODZTYPE - the type of a (length) measurement
typedef enum
{
msodztypeMin         = 0,
msodztypeDefault     = 0,  // Default size, ignore the values
msodztypeA           = 1,  // Values are in EMUs
msodztypeV           = 2,  // Values are in pixels
msodztypeShape       = 3,  // Values are 16.16 fractions of shape size
msodztypeFixedAspect = 4,  // Aspect ratio is fixed
msodztypeAFixed      = 5,  // EMUs, fixed aspect ratio
msodztypeVFixed      = 6,  // Pixels, fixed aspect ratio
msodztypeShapeFixed  = 7,  // Proportion of shape, fixed aspect ratio
msodztypeFixedAspectEnlarge = 8,  // Aspect ratio is fixed, favor larger size
msodztypeAFixedBig   = 9,  // EMUs, fixed aspect ratio
msodztypeVFixedBig   = 10, // Pixels, fixed aspect ratio
msodztypeShapeFixedBig= 11,// Proportion of shape, fixed aspect ratio
msodztypeMax         = 11
} MSODZTYPE;

// MSOFILLTYPE
typedef enum
{
msofillSolid,       // Fill with a solid color
msofillPattern,     // Fill with a pattern (bitmap)
msofillTexture,     // A texture (pattern with its own color map)
msofillPicture,     // Center a picture in the shape
msofillShade,       // Shade from start to end points
```
msofillShadeCenter,       // Shade from bounding rectangle to end point
msofillShadeShape,       // Shade from shape outline to end point
msofillShadeScale,       // Similar to msofillShade, but the fillAngle
  // is additionally scaled by the aspect ratio of
  // the shape. If shape is square, it is the
  // same as msofillShade.
msoshowShadeTitle,        // special type - shade to title --- for PP
msofillBackground         // Use the background fill color/pattern
} MSOFILLTYPE;

// MSOSHADETYPE - how to interpret the colors in a shaded fill.
typedef enum
{
  msoshadeNone  = 0,        // Interpolate without correction between RGBs
  msoshadeGamma = 1,        // Apply gamma correction to colors
  msoshadeSigma = 2,        // Apply a sigma transfer function to position
  msoshadeBand  = 4,        // Add a flat band at the start of the shade
  msoshadeOneColor = 8,     // This is a one color shade

  /* A parameter for the band or sigma function can be stored in the top
   16 bits of the value - this is a proportion of *each* band of the
   shade to make flat (or the approximate equal value for a sigma
   function). NOTE: the parameter is not used for the sigma function,
   instead a built in value is used. This value should not be changed
   from the default! */
  msoshadeParameterShift = 16,
  msoshadeParameterMask  = 0xffff0000,

  msoshadeDefault = (msoshadeGamma|msoshadeSigma|
  (16384<<msoshadeParameterShift))
} MSOSHADETYPE;

// MSOLINESTYLE - compound line style
typedef enum
{
  msolineSimple,     // Single line (of width lineWidth)
  msolineDouble,     // Double lines of equal width
  msolineThickThin,  // Double lines, one thick, one thin
  msolineThinThick,  // Double lines, reverse order
  msolineTriple      // Three lines, thin, thick, thin
} MSOLINESTYLE;

// MSOLINETYPE - how to "fill" the line contour
typedef enum
{
  msolineSolidType,         // Fill with a solid color
  msolinePattern,           // Fill with a pattern (bitmap)
  msolineTexture,           // A texture (pattern with its own color map)
  msolinePicture            // Center a picture in the shape
} MSOLINETYPE;

// MSOLINEDASHING - dashed line style
typedef enum
{
msolineSolid,    // Solid (continuous) pen
msolineDashSys,  // PS_DASH system dash style
msolineDotSys,   // PS_DOT system dash style
msolineDashDotSys,  // PS_DASHDOT system dash style
msolineDashDotDotSys, // PS_DASHDOTDOT system dash style
msolineDotGEL,   // square dot style
msolineDashGEL,  // dash style
msolineLongDashGEL, // long dash style
msolineDashDotGEL, // dash short dash
msolineLongDashDotGEL, // long dash short dash
msolineLongDashDotDotGEL // long dash short dash short dash
} MSOLINEDASHING;

// MSOLINEEND - line end effect
typedef enum
{
    msolineNoEnd,
    msolineArrowEnd,
    msolineArrowStealthEnd,
    msolineArrowDiamondEnd,
    msolineArrowOvalEnd,
    msolineArrowOpenEnd,
} MSOLINEEND;

// MSOLINEENDWIDTH - size of arrowhead
typedef enum
{
    msolineNarrowArrow,
    msolineMediumWidthArrow,
    msolineWideArrow
} MSOLINEENDWIDTH;

// MSOLINEENDDLENGTH - size of arrowhead
typedef enum
{
    msolineShortArrow,
    msolineMediumLenArrow,
    msolineLongArrow
} MSOLINEENDDLENGTH;

// MSOLINEJOIN - line join style.
typedef enum
{
    msolineJoinBevel,  // Join edges by a straight line
    msolineJoinMiter,  // Extend edges until they join
    msolineJoinRound   // Draw an arc between the two edges
} MSOLINEJOIN;

// MSOLINECAP - line cap style (applies to ends of dash segments too).
typedef enum
{
    msolineEndCapRound, // Rounded ends - the default
    msolineEndCapSquare, // Square protrudes by half line width
    msolineEndCapFlat   // Line ends at end point
struct ADJH {
    ULONG f;          // type of adjust handle
    ADJPOS apX;      // X position of the handle
    ADJPOS apY;      // Y position of the handle
    LONG xRange;     // Map values
    LONG yRange;
    LONG xMin;       // Pin values
    LONG xMax;
    LONG yMin;
    LONG yMax;
};

// SGF - Shape Guide Formula
enum SGF {
    // Arithmetic
    sgfSum,             // guide := value + parameter1 - parameter2
    sgfProduct,         // guide := value * parameter1 / parameter2 (MulDiv)
    sgfMid,             // guide := (value + param1)/2

    // Logical
    sgfAbsolute,        // guide := fabs(value)
    sgfMin,             // guide := min(value, param1)
    sgfMax,             // guide := max(value, param1)
    sgfIf,              // guide := vaule > 0 ? param1 : param2

    // Polar arithmetic - angles are 16.16 degrees
    sgfMod,              // guide := sqrt(value^2 + param1^2 + param2^2)
    sgfATan2,            // guide := atan2(param1, value) [param2 ignored]
    sgfSin,              // guide := value * sin(param1) [param2 ignored]
    sgfCos,              // guide := value * cos(param1) [param2 ignored]
    sgfCosATan2,         // guide := value * cos(atan2(param2, param1))
    sgfSinATan2,         // guide := value * sin(atan2(param2, param1))
    sgfSqrt,             // guide := sqrt(value) [param1 and param2 ignored]
    sgfSumAngle,
    sgfEllipse,
    sgfTan,              // guide := value * tan(param1) [param2 ignored]
};

// SGVT - Shape guide parameter type
enum SGVT {
    sgvtV = 0x2000,     // Value is adjust value or guide
    sgvtP1 = 0x4000,    // Parameter1 is adjust value or guide
    sgvtP2 = 0x8000     // Parameter2 is adjust value or guide
};

// Value index: The first set define the value as an adjust handle property.
const USHORT sgvAdjust1 = msopidAdjustValue;
const USHORT sgvAdjust2 = msopidAdjust2Value;
const USHORT sgvAdjust3 = msopidAdjust3Value;
const USHORT sgvAdjust4 = msopidAdjust4Value;
const USHORT sgvAdjust5 = msopidAdjust5Value;
const USHORT sgvAdjust6 = msopidAdjust6Value;
const USHORT sgvAdjust7 = msopidAdjust7Value;
const USHORT sgvAdjust8 = msopidAdjust8Value;

// Geometry width and height
const USHORT sgvWidth   = msopidGeoRight;
const USHORT sgvHeight  = msopidGeoBottom;

// Other geometry properties
const USHORT sgvXCenter = msopidGeoLeft;
const USHORT sgvYCenter = msopidGeoTop;
const USHORT sgvXLimo   = msopidXLimo;
const USHORT sgvYLimo   = msopidYLimo;

// The guide properties
#define sgvGuide_(n) (USHORT(msopidLast + 1 + (n)))

/* View coordinate properties in V units (pixels) and EMU, the EMU scaling
   is the average of the width/height scaling */
const USHORT sgvLineWidth = sgvGuide_(247);  // line width in pixels
const USHORT sgvDxvAnchor = sgvGuide_(248);  // width in pixels
const USHORT sgvDyvAnchor = sgvGuide_(249);  // height in pixels
const USHORT sgvDxeAnchor = sgvGuide_(252);  // width of a shape in the view in emus
const USHORT sgvDyeAnchor = sgvGuide_(253);  // Height of a shape in the view in emus
const USHORT sgvXeAnchorCenter = sgvGuide_(254);
const USHORT sgvYeAnchorCenter = sgvGuide_(255);

// SG - Shape Guide
struct SG
{
   ULONG sgf:16;
   ULONG iValue:16;
   ULONG a:16;
   ULONG b:16;
};

// MSOGV - Generic Value
typedef void *MSOGV;

// DiaGraM Node Kind
d enum DGMNK
{
   dgmnkMin = 0,
   dgmnkNode = dgmnkMin,
   dgmnkRoot,
   dgmnkAssistant,
   dgmnkCoWorker,
   dgmnkSubordinate,
   dgmnkAuxNode,
   dgmnkDefault, // used for the orgchart split bar
dnmknMax, dgmknLast = dgmknMax - 1,
    dgmknNil = 0xFFFF,
);  

/* MSODGMLO = DiaGraG LayOut */
typedef enum
{
    msodgmloFirst = 0, msodgmloMin = msodgmloFirst, msodgmloMinLessOne = msodgmloMin - 1,

    // OrgChart layout
    msodgmloOrgChartMin,
    msodgmloOrgChartStd = 0,
    msodgmloOrgChartBothHanging,
    msodgmloOrgChartRightHanging,
    msodgmloOrgChartLeftHanging,
    msodgmloOrgChartMax, msodgmloOrgChartLast = msodgmloOrgChartMax - 1,

    // Cycle layout
    msodgmloCycleMin, msodgmloCycleMinLessOne = msodgmloCycleMin - 1,
    msodgmloCycleStd,
    msodgmloCycleMax, msodgmloCycleLast = msodgmloCycleMax - 1,

    // Radial layout
    msodgmloRadialMin, msodgmloRadialMinLessOne = msodgmloRadialMin - 1,
    msodgmloRadialStd,
    msodgmloRadialMax, msodgmloRadialLast = msodgmloRadialMax - 1,

    // Stacked layout
    msodgmloStackedMin, msodgmloStackedMinLessOne = msodgmloStackedMin - 1,
    msodgmloStackedStd,
    msodgmloStackedMax, msodgmloStackedLast = msodgmloStackedMax - 1,

    // Venn layout
    msodgmloVennMin, msodgmloVennMinLessOne = msodgmloVennMin - 1,
    msodgmloVennStd,
    msodgmloVennMax, msodgmloVennLast = msodgmloVennMax - 1,

    // BullsEye layout
    msodgmloBullsEyeMin, msodgmloBullsEyeMinLessOne = msodgmloBullsEyeMin - 1,
    msodgmloBullsEyeStd,
    msodgmloBullsEyeMax, msodgmloBullsEyeLast = msodgmloBullsEyeMax - 1,

    msodgmloMax, msodgmloLast = msodgmloMax - 1,
    msodgmloNil = 0xFF,
} MSODGMLO;

// MSODGMT -- DiaGraM Type
typedef enum
{
    msodgmtMin = 0,
    msodgmtCanvas = msodgmtMin,
    msodgmtFirstDiagramType = 1,
    msodgmtOrgChart = msodgmtFirstDiagramType,
    msodgmtRadial = 2,
msodgmtCycle = 3,
msodgmtStacked = 4,
msodgmtVenn = 5,
msodgmtBullsEye = 6,
msodgmtMax, msodgmtLast = msodgmtMax - 1,
msodgmtNil = 0x0FFF,
}\ MSODGMT;

/* MSODGMST = DiaGraM STryle */
typedef enum
{
    /** WARNING: This is written out to the file format! **/
    /** NOTE: If you add a new style enum, you need to update vrgdgmstdesc **/

    msodgmstMin = 0, msodgmstFirst = msodgmstMin,

    // OrgChart styles
    msodgmstOrgChartFirst = msodgmstFirst,
    msodgmstOrgChart2,
    msodgmstOrgChart3,
    msodgmstOrgChart4,
    msodgmstOrgChart5,
    msodgmstOrgChart6,
    msodgmstOrgChart7,
    msodgmstOrgChart8,
    msodgmstOrgChart9,
    msodgmstOrgChart10,
    msodgmstOrgChart11,
    msodgmstOrgChart12,
    msodgmstOrgChart13,
    msodgmstOrgChart14,
    msodgmstOrgChart15,
    msodgmstOrgChart16,
    msodgmstOrgChart17,
    msodgmstOrgChartMax,
    msodgmstOrgChartLast = msodgmstOrgChartMax - 1,

    // Radial styles
    msodgmstRadialFirst = msodgmstFirst,
    msodgmstRadial2,
    msodgmstRadial3,
    msodgmstRadial4,
    msodgmstRadial5,
    msodgmstRadial6,
    msodgmstRadial7,
    msodgmstRadial8,
    msodgmstRadial9,
    msodgmstRadial10,
    msodgmstRadialMax,
    msodgmstRadialLast = msodgmstRadialMax - 1,

    // Cycle styles
    msodgmstCycleFirst = msodgmstFirst,
    msodgmstCycle2,
msodgmstCycle3,
msodgmstCycle4,
msodgmstCycle5,
msodgmstCycle6,
msodgmstCycle7,
msodgmstCycle8,
msodgmstCycle9,
msodgmstCycle10,
msodgmstCycle2First,
msodgmstCycle11 = msodgmstCycle2First,
msodgmstCycle12,
msodgmstCycle13,
msodgmstCycle14,
msodgmstCycle15,
msodgmstCycle16,
msodgmstCycle17,
msodgmstCycle18,
msodgmstCycle19,
msodgmstCycle20,
msodgmstCycle21,
msodgmstCycleMax,
msodgmstCycleLast = msodgmstCycleMax - 1,

// Stacked styles
msodgmstStackedFirst = msodgmstFirst,
msodgmstStacked2,
msodgmstStacked3,
msodgmstStacked4,
msodgmstStacked5,
msodgmstStacked6,
msodgmstStacked7,
msodgmstStacked8,
msodgmstStacked9,
msodgmstStacked10,
msodgmstStackedMax,
msodgmstStackedLast = msodgmstStackedMax - 1,

// Venn styles
msodgmstVennFirst = msodgmstFirst,
msodgmstVenn2,
msodgmstVenn3,
msodgmstVenn4,
msodgmstVenn5,
msodgmstVenn6,
msodgmstVenn7,
msodgmstVenn8,
msodgmstVenn9,
msodgmstVenn10,
msodgmstVennMax,
msodgmstVennLast = msodgmstVennMax - 1,

// BullsEyeChart styles
msodgmstBullsEyeFirst = msodgmstFirst,
msodgmstBullsEye2,
msodgmstBullsEye3,
msodgmstBullsEye4,
msodgmstBullsEye5,
msodgmstBullsEye6,
msodgmstBullsEye7,
msodgmstBullsEye8,
msodgmstBullsEye9,
msodgmstBullsEye10,
msodgmstBullsEyeMax,
msodgmstBullsEyeLast = msodgmstBullsEyeMax - 1,
msodgmstNil = 0xFFFF,
} MSODGMST;

// DiaGraM ReLationship
//-----------------------------------------------------------------------------
struct DGMRL
{
    ULONG spidSrc; // spid of the source node.
    ULONG spidDest; // spid of the destination node.
    ULONG spidCntr; // spid of the connector.
};