[MS-VSDX]:

Visio Graphics Service VSDX File Format

Revision Summary

Date	Revision History	Revision Class	Comments
1/20/2012	0.1	New	Released new document.
4/11/2012	0.1	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	0.1	No Change	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	0.1	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	1.0	Major	Significantly changed the technical content.
2/11/2013	2.0	Major	Significantly changed the technical content.
7/30/2013	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
9/4/2015	3.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Visio Graphics Service VSDX File Format describes a Web Drawing, which is a collection of Drawing Pages, Masters, Shapes, Images, Comments, Data Connections, and recalculation information that can be rendered as a drawing.

Sections 1.7 and 2 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

add-in: Supplemental functionality that is provided by an external application or macro to extend the capabilities of an application.

American National Standards Institute (ANSI) character set: A **character set** defined by a code page approved by the American National Standards Institute (ANSI). The term "ANSI" as used to signify Windows code pages is a historical reference and a misnomer that persists in the Windows community. The source of this misnomer stems from the fact that the Windows code page 1252 was originally based on an ANSI draft, which became International Organization for Standardization (ISO) Standard 8859-1 [ISO/IEC-8859-1]. In Windows, the ANSI character set can be any of the following code pages: 1252, 1250, 1251, 1253, 1254, 1255, 1256, 1257, 1258, 874, 932, 936, 949, or 950. For example, "ANSI application" is usually a reference to a non-**Unicode** or code-page-based application. Therefore, "ANSI character set" is often misused to refer to one of the character sets defined by a Windows code page that can be used as an active system code page; for example, character sets defined by code page 1252 or character sets defined by code page 950. Windows is now based on **Unicode**, so the use of ANSI character sets is strongly discouraged unless they are used to interoperate with legacy applications or legacy data.

assembly name: The name of a collection of one or more files that is versioned and deployed as a unit. See also assembly.

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

bitmap (BMP): A representation of characters or graphics by individual pixels. The pixels can be arranged in rows (horizontal) and columns (vertical). Each pixel can be represented by one or more bits.

Boolean: An operation or expression that can be evaluated only as either true or false.

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

class name: The name that is used to refer to a class module that provides an implementation of a behavior.

color space: A system that describes color numerically by mapping color components to a multidimensional coordinate system. The number of dimensions is typically two, three, or four. For example, if colors are expressed as a combination of the three components red, green, and blue, a three-dimensional space can describe all possible colors. Grayscale colors can be mapped

to a two-dimensional color space. If transparency is considered a component, four dimensions are appropriate. Also referred to as color model.

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

culture name: A part of a language identification tagging system, as described in [RFC1766]. Culture names adhere to the format "<languagecode2>-<country/regioncode2>." If a two-letter language code is not available, a three-letter code that is derived from [ISO-639] is used.

data provider: A known data source that is specific to a target type and that provides data to a collector type.

data source: A database, web service, disk, file, or other collection of information from which data is queried or submitted. Supported data sources vary based on application and data provider.

drawing: A collection of drawing objects, such as shapes, curves, or WordArt, that are viewed together as a single image.

embedded image: An image that is stored within a document rather than being linked to a source file that is outside the document.

embedded object: An object that is created by using one application and is hosted in a document that was created by using another application. Embedding an object, rather than inserting or pasting it, ensures that the object retains its original format. Users can double-click an embedded object and edit it with the toolbars and menus from the application that was used to create it. See also Object Linking and Embedding (OLE).

enhanced metafile format (EMF): A file format that supports the device-independent definitions of images.

field: An element or attribute (1) in a data source that can contain data.

floating-point number: A number that is represented by a mantissa and an exponent according to a given base. The mantissa is typically a value between "0" and "1". To find the value of a floating-point number, the base is raised to the power of the exponent, and the mantissa is multiplied by the result.

font: An object that defines the graphic design, or formatting, of a collection of numbers, symbols, and letters. A font specifies the style (such as bold and strikeout), size, family (a typeface such as Times New Roman), and other qualities to describe how the collection is drawn.

gamma correction: In digital imaging, the process of changing the brightness, contrast, or color balance of an image by assigning new values (different colors) to gray or color tones.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

Graphics Interchange Format (GIF): A compression format that supports device-independent transmission and interchange of bitmapped image data. The format uses a palette of up to 256 distinct colors from the 24-bit **RGB** color space. It also supports animation and a separate palette of 256 colors for each frame. The color limitation makes the GIF format unsuitable for reproducing color photographs and other images with gradients of color, but it is well-suited for simpler images such as graphics with solid areas of color.

header row: A row in a table, typically the first row, that contains labels for columns (2) in the table.

hue-saturation-luminance (HSL): A color model that defines a color by using three dimensions: hue, the color itself; saturation, the purity of the color; and luminance, the amount of light that is either reflected or absorbed by the color. See also color scheme and **color space**.

hyperlink location: A portion of a hyperlink that specifies the location of a specific item, such as a bookmark (1), within a document, object, or other type of resource; for example "#bookmark" in the hyperlink location C:\Documents\Document.docx#bookmark.

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Joint Photographic Experts Group (JPEG): A raster graphics file format for displaying high-resolution color graphics. JPEG graphics apply a user-specified compression scheme that can significantly reduce the file sizes of photo-realistic color graphics. A higher level of compression results in lower quality, whereas a lower level of compression results in higher quality. JPEG-format files have a .jpg or .jpeg file name extension.

language code identifier (LCID): A 32-bit number that identifies the user interface human language dialect or variation that is supported by an application or a client computer.

list: (1) A container within a SharePoint site that stores list items. A list has a customizable schema that is composed of one or more fields.

(2) An organization of a region of cells into a tabular structure in a workbook.

metafile: A file that stores an image as graphical objects, such as lines, circles, and polygons, instead of pixels. A metafile preserves an image more accurately than pixels when an image is resized.

Office data connection (ODC) file: A file that stores information about a connection to a data source, such as an Access database, worksheet, or text file. This file facilitates data source administration.

OLE DB: A set of interfaces that are based on the Component Object Model (COM) programming model and expose data from a variety of sources. These interfaces support the amount of Database Management System (DBMS) functionality that is appropriate for a data store and they enable a data store to share data.

Open Database Connectivity (ODBC): A standard software API method for accessing data that is stored in a variety of proprietary personal computer, minicomputer, and mainframe databases. It is an implementation of [ISO/IEC9075-3:2008] and provides extensions to that standard.

Portable Network Graphics (PNG): A bitmap graphics file format that uses lossless data compression and supports variable transparency of images (alpha channels) and control of image brightness on different computers (gamma correction). PNG-format files have a .png file name extension.

primary key: A field or set of fields that uniquely identifies each record in a table. A primary key cannot contain a null value.

query: A formalized instruction to a data source to either extract data or perform a specified action. A query can be in the form of a query expression, a method-based query, or a combination of the two. The data source can be in different forms, such as a relational database, XML document, or in-memory object. See also search query.

red-green-blue (RGB): A color model that describes color information in terms of the red (R), green (G), and blue (B) intensities in a color.

row: A single set of data that is displayed horizontally in a worksheet or a table.

Tagged Image File Format (TIFF): A high-resolution, tag-based graphics format. TIFF is used for the universal interchange of digital graphics.

text run: A string of characters that represents a discrete span of text with the same formatting properties.

token: A word in an item or a search query that translates into a meaningful word or number in written text. A token is the smallest textual unit that can be matched in a search query. Examples include "cat", "AB14", or "42".

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

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

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

UTF-16: A standard for encoding Unicode characters, defined in the Unicode standard, in which the most commonly used characters are defined as double-byte characters. Unless specified otherwise, this term refers to the UTF-16 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

view: See form view (Microsoft InfoPath), list view (SharePoint Products and Technologies), or View (Microsoft Business Connectivity Services).

whitespace: A character that can be found between words, including a space (" "), a carriage return in combination with a line feed (newline), and a tab character.

workbook: A container for a collection of sheets (1).

zero-based index: An index in which the first item has an index of "0" (zero).

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

1.2 References

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

1.2.1 Normative References

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

[GIF89a] CompuServe Incorporated, "Graphics Interchange Format(sm)", Graphics Interchange Format Programming Reference, July 1990, http://www.w3.org/Graphics/GIF/spec-gif89a.txt

[IEEE754] IEEE, "IEEE Standard for Binary Floating-Point Arithmetic", IEEE 754-1985, October 1985, http://ieeexplore.ieee.org/servlet/opac?punumber=2355

[ISO-15924] International Organization for Standardization, "ISO 15924 Registration Authority", http://www.unicode.org/iso15924/

[ISO-8601] International Organization for Standardization, "Data Elements and Interchange Formats - Information Interchange - Representation of Dates and Times", ISO/IEC 8601:2004, December 2004, http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874&ICS1=1&ICS2=140&ICS3=30

Note There is a charge to download the specification.

[ISO/IEC29500-1:2011] ISO/IEC, "Information Technology -- Document description and processing languages -- Office Open XML File Formats -- Part 1: Fundamentals and Markup Language Reference", ISO/IEC 29500-1:2011, 2011, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59575

[ISO/IEC29500-2:2011] ISO/IEC, "Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 2: Open Packaging Conventions", ISO/IEC 29500-2:2011, 2011, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59576

[ISO/IEC29500-3:2011] ISO/IEC, "Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 3: Markup Compatibility and Extensibility", ISO/IEC 29500-3:2011, 2011, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59577

[JFIF] Hamilton, E., "JPEG File Interchange Format, Version 1.02", September 1992, http://www.w3.org/Graphics/JPEG/jfif.txt

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

[MS-OAUT] Microsoft Corporation, "OLE Automation Protocol".

[MS-ODBCSTR] Microsoft Corporation, "ODBC Connection String Structure".

[MSDN-BMPST] Microsoft Corporation, "Bitmap Storage", http://msdn.microsoft.com/en-us/library/dd183391(VS.85).aspx

[RFC2083] Boutell, T., et al., "PNG (Portable Network Graphics) Specification Version 1.0", RFC 2083, March 1997, http://www.ietf.org/rfc/rfc2083.txt

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

[RFC3302] Parsons, G., and Rafferty, J., "Tag Image File Format (TIFF) - image/tiff MIME Sub-Type Registration", RFC 3302, September 2002, http://www.ietf.org/rfc/rfc3302.txt

[RFC3629] Yergeau, F., "UTF-8, A Transformation Format of ISO 10646", STD 63, RFC 3629, November 2003, http://www.ietf.org/rfc/rfc3629.txt

[RFC4646] Phillips, A., and Davis, M., Eds., "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006, http://www.rfc-editor.org/rfc/rfc4646.txt

[RFC4647] Phillips, A., and Davis, M., Eds., "Matching of Language Tags", BCP 47, RFC 4647, September 2006, http://www.rfc-editor.org/rfc/rfc4647.txt

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, http://www.rfc-editor.org/rfc/rfc5234.txt

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

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

[MSDN-CompareOptions] Microsoft Corporation, "CompareOptions Enumeration", .NET Framework Class Library, http://msdn.microsoft.com/en-us/library/system.globalization.compareoptions.aspx

[MSDN-ENCLOC] Microsoft Corporation, "Encoding and Localization", .NET Framework Developer's Guide, http://msdn.microsoft.com/en-us/library/h6270d0z.aspx

[MSDN-ToDouble] Microsoft Corporation, "Convert.ToDouble Method", .NET Framework Class Library, http://msdn.microsoft.com/en-us/library/system.convert.todouble.aspx

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.rfc-editor.org/rfc/rfc2616.txt

1.3 Overview

This structure describes a ZIP archive that stores all the information needed to describe a web drawing.

A Document XML Part in the ZIP archive describes the properties of the web drawing.

A collection of Visio parts and Shared XML parts in the ZIP archive describes the graphical elements displayed in the web drawing. These graphical elements are presented as Shapes on Drawing Pages. Shapes are described by the Master XML Part, Page XML Part, and Themes XML Part. Drawing Pages are described by the Masters XML Part and Pages XML Part.

Graphical elements can be static or dynamic. Dynamic graphical elements have visual properties that are bound to data in a **data source**, and the appearance of these elements changes as data in the data source refreshes (section 2.2.10). A collection of Visio parts in the ZIP archive describes the Data Connections, bindings (section 2.2.10.2.1) between data and shapes, and recalculation information necessary to update (section 2.2.11) visual properties. Data connections are described by the Connections XML part. Data bindings are described by the Recordsets XML part. Recalculation information is described by a grammar (section 2.2.11.2.1) for Formula Evaluation that describes how changes in the data are translated into changes in properties of graphical elements. This grammar is described by the Master XML part and Page XML part.

Additional items in the ZIP archive describe the Images and Comments in the web drawing.

1.4 Relationship to Protocols and Other Structures

This specification is dependent on the structures and concepts defined in [ISO/IEC29500-2:2011], [ISO/IEC29500-3:2011] and [ISO/IEC29500-1:2011] section 9 for Open Packaging Conventions.

1.5 Applicability Statement

This document specifies a persistence format for Web Drawing content, which can include Drawing Pages, Masters, Shapes, Images, Comments, Data Connections, and recalculation information, as specified in Section 2.2.1. The persistence format is applicable when the document content is graphical in nature.

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

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

1.6 Versioning and Localization

This document covers versioning issues in the following areas:

- App XML Part
- Custom XL Part
- Version

This document covers localization in the Core XML Part.

Local overrides to document language are specified in attributes, properties, and function arguments as described in the Conceptual Overview, Visio XML Schema, ShapeSheet Properties, and Formula Expressions and Evaluation sections.

1.7 Vendor-Extensible Fields

Persistence format can be extended by storing information in Parts that are not specified in Section 2. Implementations are not required to preserve or remove additional Parts when modifying an existing document.

2 Structures

This section specifies the overall structure of a file that conforms to this specification.

2.1 File Structure Overview

A file of the type specified by this specification MUST be a Package that is a ZIP archive.

The ZIP Package is used to persist information that is necessary to fully represent a web Drawing. This package contains a collection of Parts that are used to persist data in XML or standard binary formats, and to specify various aspects of the Web Drawing as well as the structure of the Package.

2.1.1 Package

A file of the type specified by this document MUST be a Package that is a ZIP archive and that conforms to the Open Packaging Conventions as specified in [ISO/IEC29500-2:2011], the further packaging restrictions specified in [ISO/IEC29500-1:2011] section 9, and this specification.

2.1.2 Parts

A Package is composed of multiple parts as specified in [ISO/IEC29500-2:2011] section 9.1. Each part has an associated content type that specifies the format it is persisted in. Each part can also be the target or the source of a connection between two parts called a relationship (section 2.1.3), as specified in [ISO/IEC29500-2:2011] section 9.3.

The valid parts, content types, required relationships, and optional relationships between all parts in this package are specified in Part Enumeration.

2.1.3 Relationship

A relationship specifies a connection between a source and a target resource as specified in [ISO/IEC29500-1:2011] section 9.2. Relationship identifiers are used in binary and XML part (section 2.1.2) content to reference unique relationship elements in relationship parts that in turn target other resources.

There are several different types of relationships:

- A Package relationship is a relationship where the target is a part and the source is the package as a whole.
- A part-to-part relationship is a relationship where the target is a part and the source is a part in the package.
- An explicit relationship is a relationship where a resource is referenced from the contents of a source part by referencing the **ID** attribute value of a relationship element.
- An implicit relationship is a relationship where a resource is not referenced from the contents of a source part by referencing the **ID** attribute value of a relationship element.
- An internal relationship is a relationship where the target is a part in the package.
- An external relationship is a relationship where the target is an external resource, not part of the package.

2.1.4 Markup Compatibility

A markup specification defines a set of elements and attributes within one or more namespaces. A characteristic of an application that consumes the markup is that it can recognize the elements and attributes within understood namespaces, including those containing elements and attributes defined in the markup specification. Markup consumers MUST treat all recognized elements and attributes of any understood namespace according to the requirements of the markup specifications defining those elements or attributes. A markup specification MAY require that the presence of unrecognized elements or attributes in an understood namespace be treated as an error condition; however, markup consumers MUST always treat the presence of an unrecognized element or attribute from the Markup Compatibility namespace as an error condition.

If a markup consumer encounters an element or attribute from a non-understood namespace, the markup consumer MUST treat the presence of that element or attribute as an error condition, unless the markup producer has embedded in the markup document explicit Markup Compatibility elements or attributes that override that behavior.

The valid Markup Compatibility elements and attributes in a Web Drawing are specified in the Markup Compatibility Schema (section 2.3.5).

2.2 Conceptual Overview

The Conceptual Overview sections that follow specify how higher-level features of the file format are represented by combinations of parts and XML elements.

2.2.1 Web Drawing

A web drawing is a collection of Drawing Pages, Masters, Shapes, Images, Comments, Data Connections, and recalculation information that can be rendered as a **drawing** in a web browser.

A web drawing is specified by a Package as specified in the File Structure Overview. The contents of a web drawing are specified by the Parts in the Part Enumeration section.

For examples of various web drawings, see Structure Examples.

2.2.2 Drawing Page

A drawing page is a collection of Shapes that are viewed together.

A collection of drawing pages in a web drawing is specified by a Pages XML Part.

2.2.2.1 Page Identification

A Page_Type element in a Pages XML Part specifies a single drawing page. A drawing page is uniquely identified by the **ID**, **Name**, and **NameU** attributes in a Page_Type element. The following elements in parts of the web drawing have attributes that are equal to **ID**, **Name**, or **NameU** and specify supplementary information about the drawing page.

- A PublishedPage_Type element in a Document XML Part has an **ID** attribute that is equal to the **ID** attribute of the Page_Type element, and specifies that the drawing page is viewable in the web drawing.
- A **TitlesOfParts** element in an App XML Part contains an **lpstr** element with contents equal to the **Name** attribute of the Page_Type element and specifies the name of the drawing page.
- A Page_Type element in a Pages XML Part can have a **BackPage** attribute that is equal to the **ID** attribute of the Page_Type element, and specifies that the latter drawing page is to be used as the background page for the former drawing page.

- A RowMap_Type element in the Recordsets XML Part contains a **PageID** attribute that is equal to the **ID** attribute of the Page_Type element, and specifies the data binding between a **row** of a Recordset and a Shape on the drawing page.

The graphical information necessary to render a drawing page is specified by the PageSheet_Type and ShapeSheet_Type elements in a Pages XML Part and a Shapes_Type element in a Page XML Part.

A drawing page is also associated with a Master. The graphical information about a Master is specified by the PageSheet_Type and Shapes_Type elements in a Master XML Part.

A drawing page can contain **embedded images**. Each Image used in a drawing page is specified by an Image Part. The Fallback Image section explains how some embedded image formats and embedded objects are rendered using Fallback Images which are also specified by Image Parts.

2.2.2.2 Coordinate System

A point on a drawing page is specified by coordinates on a two-dimensional Cartesian plane, where the x-coordinate specifies the horizontal position and the y-coordinate specifies the vertical position.

The origin of a drawing page is the lower-left corner of the drawing page.

Increasing the x-coordinate specifies the position of a Shape, group or object rightward, while increasing the y-coordinate specifies the position upward.

Every drawing page defines its own coordinate system.

2.2.2.3 Drawing Scale

The drawing scale of a drawing page is the ratio of the values of the PageScale Cell_Type element to the value of the DrawingScale Cell_Type element.

Drawing units specify size or position of objects on the drawing page. Page units specify measurements on the printed page.

The drawing scale multiplied by the drawing units will result in a scaled object.

The following cells are not expressed in drawing units and are not scaled. All other vLengths are expressed in drawing units and will be scaled.

- BeginArrowSize, EndArrowSize
- GlowSize
- ReflectionBlur
- ReflectionDist
- SoftEdgesSize
- LineWeight
- FontScale, Size
- AsianFont
- Case
- Color, ColorTrans
- DblUnderline

- ComplexScriptFont, ComplexScriptSize
- DoubleStrikethrough
- Overline
- Pos
- Strikethru
- Style
- BevelTopWidth, BevelTopHeight, BevelBottomWidth, BevelBottomHeight
- BevelDepthSize, BevelContourSize
- ShdwOffsetX, ShdwOffsetY
- DistanceFromGround

2.2.2.4 Foreground Page

A drawing page can be a foreground page. A web drawing contains at least one foreground page.

The Page_Type element specifies whether a page is a foreground page in a web drawing. If a Page_Type element in a Pages XML Part contains a **Background** attribute equal to zero, it is a foreground page.

A foreground page in a web drawing has zero or one background pages as specified by the **BackPage** attribute of the Page_Type element associated with the page.

A foreground page can be published or unpublished. Published pages are viewable in a web drawing while unpublished pages are hidden. The PublishSettings_Type child element of the VisioDocument_Type element for the web drawing determines whether a page is published or unpublished.

2.2.2.5 Background Page

A background page is a drawing page that can appear behind foreground pages and other background pages in a web drawing.

A background page can have a different drawing scale than a foreground page.

A background page in a web drawing is specified by the Page_Type element associated with the page.

If the Page_Type element associated with the page contains a **Background** attribute equal to one, it is a background page.

A background page in a web drawing has zero or one background pages as specified by the **BackPage** attribute of the Page_Type element associated with the page.

2.2.2.6 Layer

A web drawing can have layers. A shape belongs to zero or more layers. A layer can contain zero or more shapes. A layer specifies additional information about the shapes that it contains such as color, color transparency, and visibility.

A layer in a web drawing is specified by the Row_Type child element of a Layer Section_Type element. A Layer Section_Type element is a child of a PageSheet_Type element associated with the page.

Each Row_Type child element of the Layer Section_Type element contains information for a single layer. A layer is uniquely identified by the **IX** attribute of that layer's Row_Type.

A collection of Cell_Type elements that define a layer's properties is composed of Color, Visible, Lock, and ColorTrans.

The layer membership of a shape is specified by the LayerMember Cell_Type element in the ShapeSheet_Type element of the shape.

2.2.3 Shape

A shape is a collection of Geometry Visualization, Format, Text, Images, and Shape Data in a Drawing Page.

2.2.3.1 Shape Identification

A Shape in a Web Drawing is specified by a ShapeSheet_Type child element of a Shapes_Type descendant element of either a PageContents element in a Page_XML_Part, or a MasterContents element in a Master XML Part.

A Shape is uniquely identified within a Drawing Page by the **ID** attribute of its ShapeSheet_Type element. The following elements in other Parts of the document have attributes that reference shapes by their **ID** attributes to specify supplementary information about them.

- A RowMap_Type element specifies the shape it is bound to in its **ShapeID** attribute.
- A CommentEntry_Type element specifies the shape it relates to in its **ShapeID** attribute.
- The ShapeSheetRef Reference Token references a shape.

2.2.3.1.1 One-Dimensional Shape

A Shape is one-dimensional if its ShapeSheet_Type element has BeginX, BeginY, EndX, and EndY child elements of the type Cell_Type.

2.2.3.1.2 Two-Dimensional Shape

A Shape is two-dimensional if its ShapeSheet_Type element has no BeginX, BeginY, EndX, or EndY child elements of the type Cell_Type.

2.2.3.2 Geometry Visualization

Geometry on Shapes in a Web Drawing can be visualized.

The following sections specify the concepts and elements of geometry visualization.

2.2.3.2.1 Coordinate System

A point on a Drawing Page or a Shape is specified by coordinates on a two-dimensional Cartesian plane, where the x-coordinate specifies the horizontal position and the y-coordinate specifies the vertical position.

Every Shape defines a local coordinate system. A point on a shape is specified either in its local coordinates or in the coordinate system of the shape's Parent, depending on the **N** attribute of the Cell_Type element specifying this point.

A point specified in local coordinates can be converted into parent coordinates by applying the following transformations in the following order:

1. Subtract the value of the LocPinX property of the Cell_Type element from the x-coordinate.
2. Subtract the value of the LocPinY property of the Cell_Type from the y-coordinate.
3. Mirror the point about the y-axis if the value of the FlipX property of the Cell_Type is equal to one.
4. Mirror the point about the x-axis if the value of the FlipY property of the Cell_Type is equal to one.
5. Rotate the point counterclockwise around the origin by the value of the Angle property of the Cell_Type.
6. Add the value of the PinX Cell_Type to the x-coordinate.
7. Add the value of the PinY Cell_Type to the y-coordinate.

2.2.3.2.1.1 Relative Coordinate System

A relative coordinate system is a Coordinate System where the coordinates are determined by multiplying a scalar value by the width or height of the Shape.

It is used to represent x-coordinate or y-coordinate by the Cell_Type element that has a RelCubBezTo, RelEllipticalArcTo, RelLineTo, RelMoveTo or RelQuadBezTo properties of the Row_Type parent element.

It is also used to represent formula by E Cell_Type element that has a NURBSTo Row_Type parent element and A Cell_Type element that has a PolylineTo Row_Type parent element.

The width and height are specified by the Width and Height Cell_Type elements.

2.2.3.2.2 Geometry Path

A path is a collection of vertices and line or curve segments that specifies an enclosed area. The geometry of a shape is specified by a collection of paths.

Each Geometry Section_Type element specifies a path. Each Row_Type child element specifies a vertex of that path, a segment of that path, or both.

If the Row_Type element is of type Ellipse or InfiniteLine, it specifies the only segment of the path.

Otherwise, if the Row_Type element is of type MoveTo or RelMoveTo, it specifies the first vertex in the path or the first vertex after a break in the path.

Otherwise, the Row_Type element specifies a vertex and a segment that connects the vertex of the previous Row_Type element to the vertex specified in the current Row_Type element.

For a path to be visible, the following conditions are necessary.

- The shape containing the path is not on a layer whose Visible Cell_Type element has a value equal to zero.
- The value of the NoShow Cell_Type child of the path's Geometry Section_Type element is not equal to one.

The visibility of the path's line and the visibility of the path's fill are specified, respectively, by the NoLine and NoFill Cell_Type child elements of the path's Geometry Section_Type element.

The format of the path's line and the format of the path's fill are specified, respectively, by the line property and fill property of the shape containing the path.

2.2.3.2.3 Display Order

The display order of shapes in a drawing page is a strict ordering that is used to determine their visualization behavior. If one shape has a higher position than another in the display order, the former shape is displayed in front of the latter.

If one ShapeSheet_Type element appears before another ShapeSheet_Type element in the same XML Part, the shape specified by the latter ShapeSheet_Type element has a higher position in the display order.

If one shape is a member shape of another shape, the relative positions of the two shapes in the display order are specified by the DisplayMode Cell_Type of the latter shape.

2.2.3.3 Shape Hierarchy

Shapes can be hierarchically grouped. A shape contains zero or more subshapes.

2.2.3.3.1 Parent

If a ShapeSheet_Type element has a parent Shapes_Type element whose parent is a ShapeSheet_Type element, the shape specified by the latter ShapeSheet_Type element is called that shape's parent.

If a ShapeSheet_Type element has an ancestor Shapes_Type element whose parent is a ShapeSheet_Type element, the shape specified by the latter ShapeSheet_Type element is called an ancestor shape of the shape specified by the former ShapeSheet_Type element.

2.2.3.3.2 Top-Level Shape

Top-level shapes are ShapeSheet_Type elements that have no ShapeSheet_Type ancestors. The parent of top-level shapes is the drawing page.

2.2.3.3.3 Subshape

If a ShapeSheet_Type element has a parent Shapes_Type element whose parent is a ShapeSheet_Type element, the shape specified by the former ShapeSheet_Type element is called a subshape of the shape specified by the latter ShapeSheet_Type element.

If a ShapeSheet_Type element has an ancestor Shapes_Type element whose parent is a ShapeSheet_Type element, the shape specified by the former ShapeSheet_Type element is called a member shape of the shape specified by the latter ShapeSheet_Type element.

2.2.3.4 Shape Selection

Shapes in a Web drawing can be selected.

For a shape to be selectable, all the following conditions are necessary.

- The value of the LockSelect Cell_Type element of the shape is equal to zero, or value of the ProtectShape Cell_Type element of the shape is equal to one.
- The shape is not on a layer whose Visible Cell_Type element has a value equal to zero.
- The shape is not on a layer whose Lock Cell_Type element has a value equal to zero.
- None of the ancestor shapes of the shape has a SelectMode cell whose value is equal to zero.
- The shape has at least one visible geometry path that is not obscured by shapes with a higher display order.
- The shape is on a foreground page.

2.2.3.5 Shape Hyperlinks

A shape has zero or more hyperlinks associated with it. Hyperlinks point to drawing pages within the Web drawing, shapes within the Web drawing, or destinations outside the Web drawing.

The set of hyperlinks associated with a shape is specified by the Hyperlink Section_Type element.

Each hyperlink is specified by a Row_Type child element of the Hyperlink Section_Type element for the shape. This Row_Type element specifies the information about the hyperlink properties using a collection of Cell_Type elements. It is either contained under a ShapeSheet_Type element for the shape or inherited.

A collection of Cell_Type elements that define the properties of the hyperlink is composed of Description, Address, SubAddress, ExtraInfo, Default, Invisible, and SortKey Cell_Type elements.

2.2.3.6 Shape Data

A shape can have data associated with it that provides information about its meaning. A shape's data is stored as a set of shape data **fields**.

Each shape data field is specified by a Row_Type child element of the Property Section_Type element for the shape. This Row_Type element specifies the information about the shape data field properties using a collection of Cell_Type elements. It is either contained under a ShapeSheet_Type element for the shape or it is inherited.

A collection of Cell_Type elements that define the properties of the shape data field is composed of Calendar, DataLinked, Format, Invisible, Label, LangID, Type, and Value Cell_Type elements.

The name of a shape data field is specified by the **N** attribute of the Row_Type element for the field. The value of a shape data field is specified by the Value Cell_Type element. The data type of a shape data field is specified by the Type Cell_Type element.

2.2.4 Master

Masters specify shapes that can be reused throughout a web drawing.

A shape on a drawing page can be linked to a master, which can affect various properties of the shape including its visual appearance. A relationship to such a master is called master-to-shape inheritance.

2.2.4.1 Master Identification

A master is specified by the combination of a Master_Type element in a Masters XML Part, and the ShapeSheet_Type elements in the Master XML Part specified by the Master_Type element's Rel_Type child element. These ShapeSheet_Type elements are called master shapes.

The following elements in other parts of the document have attributes that reference masters.

- A ShapeSheet_Type element in a Page XML Part can specify with its **Master** attribute the master it inherits from.
- The Use function token accepts as its argument the name or **GUID** of a master.
- The MasterSheetRef reference token references a master.

2.2.5 Sheet

A sheet is a collection of properties that specify information for a shape, master, drawing page, style, or web drawing.

2.2.5.1 Sheet Identification

A sheet for a shape is a collection of sections, rows, and cells contained in a ShapeSheet_Type element in a Page XML Part. A sheet for a shape is uniquely identified by the **ID** attribute in a Shapes_Type element.

A sheet for a master is a collection of sections, rows, and cells contained in a ShapeSheet_Type element in a Master XML Part. A sheet for a master is uniquely identified by the **UniqueID** attribute in a Shapes_Type element.

A sheet for a drawing page is a collection of sections, rows, and cells contained in a PageSheet_Type element in a Masters XML Part or Pages XML Part. A sheet for a drawing page is uniquely identified by the **ID**, **Name,** and **NameU** attributes in a Pages_Type element. A sheet for a drawing page is unique in a Master_Type or Page_Type element.

A sheet for a style is a collection of sections, rows, and cells contained in a StyleSheet_Type element in a Document XML Part. A sheet for a style is uniquely identified by the **ID** attribute in a StyleSheets_Type element.

A sheet for a web drawing is a collection of sections, rows, and cells contained in a DocumentSheet_Type element in a Document XML Part. A sheet for a Web drawing is unique in a VisioDocument_Type element.

2.2.5.2 Sheet Types

A sheet is specified by a Sheet_Type abstract complex type. A sheet in a web drawing can be one of four distinct types that extend the Sheet_Type. The distinct types are shape sheet, page sheet, style sheet, and document sheet (section 2.2.5.2.1).

2.2.5.2.1 Document Sheet

A document sheet specifies information pertaining to a web drawing. It is a collection of sections, rows, and cells in a DocumentSheet_Type child element of the VisioDocument_Type element in the Document XML Part.

2.2.5.2.2 Page Sheet

A page sheet specifies information pertaining to a drawing page. It is a collection of sections, rows, and cells contained in a Pages XML Part or Masters XML Part. Each page sheet is specified by a PageSheet_Type child element of a Page_Type child element of a Pages_Type element in either a Pages XML Part or PageSheet_Type child element of a Master_Type child element of a Masters_Type element in a Masters XML Part.

2.2.5.2.3 Shape Sheet

A shape sheet specifies information pertaining to a shape or master.

A shape sheet pertaining to a shape in a web drawing is a collection of sections, rows, and cells contained in a Page_XML_Part. Each shape sheet is specified by a ShapeSheet_Type child element of a Shapes_Type descendant element of a PageContents element in a part.

A shape sheet pertaining to a master in a web drawing is a collection of sections, rows, and cells contained in a Master XML Part. Each shape sheet is specified by a ShapeSheet_Type child element of a Shapes_Type descendant element of a MasterContents element in a part.

2.2.5.2.4 Style Sheet

A style sheet specifies information pertaining to a style and is used in inheritance.

A style sheet in a web drawing is a collection of sections, rows, and cells contained in a Document XML Part. Each style sheet is specified by a StyleSheet_Type child element of the StyleSheets_Type child element of the VisioDocument element in a part.

2.2.5.2.4.1 Root Style Sheet

The root style sheet is a style sheet in a web drawing that all other style sheets inherit from.

The root style sheet is specified by the StyleSheet_Type element whose **ID** attribute value is equal to zero and whose **NameU** attribute value is equal to "No Style".

2.2.5.3 Sheet Structures

A sheet structure is where the property information of a sheet has been hierarchically grouped into sections, rows, and cells.

2.2.5.3.1 Section

A section specifies a collection of related properties of a sheet. A section contains cells and rows.

Sections are specified by Section_Type child elements of the ShapeSheet_Type, PageSheet_Type, StyleSheet_Type, and DocumentSheet_Type elements. The **N** attribute of a Section_Type element specifies the name of the section that identifies the collection of properties that it pertains to. The properties specified by a section are specified by the Cell_Type and Row_Type child elements of the Section_Type element.

2.2.5.3.2 Row

A row specifies a subset of the properties in a section. A row contains cells.

Rows are specified by Row_Type child elements of the Section_Type child elements of the ShapeSheet_Type, PageSheet_Type, StyleSheet_Type, and DocumentSheet_Type elements. The **N** attribute of a Row_Type element specifies the name of the row that identifies the subset of properties that it pertains to. The properties specified by a row are specified by the Cell_Type child elements of the Row_Type element.

2.2.5.3.3 Cell

A cell specifies a single property in a row, section, or sheet.

Cells are specified by Cell_Type child elements of the Section_Type, Row_Type, ShapeSheet_Type, PageSheet_Type, StyleSheet_Type, and DocumentSheet_Type elements. The **N** attribute of a Cell_Type element specifies the name of the cell that identifies the property that it pertains to.

The **V** attribute of a Cell_Type element specifies the value of the property of the cell. The **F** attribute of a Cell_Type element specifies the formula expression of the property of the cell.

If the **F** attribute is present, the value of the property is used until it is replaced by a value from the most recent formula evaluation that does not result in an error value.

2.2.5.3.3.1 Cell Default Values

The property value assigned to a missing or malformed cell is called a cell default value. If the Cell_Type element of a cell in a web drawing is not specified directly in a sheet or through inheritance, the cell is called a missing cell. If the Cell_Type element of a cell in a web drawing does not specify a **V** attribute, the cell is called a malformed cell.

The cell default value of a missing cell depends on its parse token, custom structure, or custom token grouping. The cell default value for parse tokens and custom structures is specified in the following table. Where the default value for all the parse tokens in a custom token grouping is the same, the custom token grouping is specified in the table instead of individual parse tokens.

Parse token, custom structure, or custom token grouping	Cell default value
PtgNum	0.00
PtgBool	0
PtgString	""
PtgByte	0
PtgColorRGB	#000000
PtgShort	0
PtgDate	0.00 days
PtgInt	0
PtgUnsShort	0
PtgNumI	0.00 inches
vLanguageString	""
vFont	"0"
vAny	0.00 days
vAngle	0.00 radians
vLength	0.00 inches
vColor	#000000
vFormatString	""

The cell default value of a malformed cell depends on the **U** attribute value of its Cell_Type element. If the **U** attribute of the Cell_Type element of a malformed cell is not specified, the cell default value is specified in the previous table.

If the **U** attribute of the Cell_Type element of a malformed cell is specified, the cell default value is specified in the following table.

U attribute value	Cell default value
AC	0.00 inches
DEG	0.00 radians
DA	0.00 radians
AD	0.00 radians
RAD	0.00 radians
BOOL	0
COLOR	#000000

U attribute value	Cell default value
CY	0.00
DATE	0.00
ED	0.00 days
EH	0.00 days
EM	0.00 days
ES	0.00 days
EW	0.00 days
HA	0.00 inches
CM	0.00 inches
DL	0.00 inches
FT	0.00 inches
F_I	0.00 inches
IN	0.00 inches
IN_F	0.00 inches
KM	0.00 inches
M	0.00 inches
MI	0.00 inches
MI_F	0.00 inches
MM	0.00 inches
NM	0.00 inches
PER	0.00
YD	0.00 inches
DP	0.00 inches
PNT	PNT(0.00, 0.00)
STR	""
DE	0.00 days
C_D	0.00 inches
C	0.00 inches
D	0.00 inches
DT	0.00 inches
P	0.00 inches
P_PT	0.00 inches

U attribute value	Cell default value
PT	0.00 inches

2.2.5.4 Inheritance

This section describes how properties are inherited in sheets in a web drawing.

2.2.5.4.1 Master-to-Shape Inheritance

A shape on a drawing page can be linked to a master, which can affect various properties of the shape including its visual appearance. A relationship to such a master is called master-to-shape inheritance, and the shape is called an instance of that master. A shape has zero or one master-to-shape inheritance relationships.

If the **Master** attribute of the ShapeSheet_Type element of a shape on a drawing page is equal to the **ID** attribute of a Master_Type element of a master, the shape is an instance of the master. Any sections, rows, cells, or subshapes not specified in the instance are inherited from the master.

An instance can modify the sections, rows, and cells taken on from inheritance by specifying local properties. In addition, if an instance contains a subshape whose ShapeSheet_Type element has a **MasterShape** attribute that matches the **ID** attribute of a subshape of the master, the local properties specified in this subshape will override those of the corresponding subshape in the master.

If a master has one top-level shape, a shape that inherits from that master inherits the descendant elements of that master shape. If a master has more than one master shape, a shape that inherits from that master inherits those master shapes as subshapes.

2.2.5.4.2 Style-to-Shape Inheritance

Shapes in a web drawing can be linked to a style, which can affect various properties of the shape including its visual appearance. A relationship to such a style is called style-to-shape inheritance. A style-to-shape inheritance allows a shape to take on properties from the style it inherits from. A shape can have zero to three style-to-shape inheritance relationships.

The style-to-shape inheritances in a web drawing are specified by the Page XML Part. Each style-to-shape inheritance is specified by the attributes of a ShapeSheet_Type child element of the Shapes_Type descendant element of the PageContents element in a part. Style-to-shape inheritance information is specified by the ShapeSheet_Type element and a StyleSheet_Type child element of a StyleSheets_Type child element of the VisioDocument element in the Document XML Part.

If the **LineStyle**, **FillStyle**, and **TextStyle** attributes of the ShapeSheet_Type element are empty, a style-to-shape inheritance is not specified. If the **LineStyle**, **FillStyle**, or **TextStyle** attributes of the ShapeSheet_Type element are not empty, a style-to-shape inheritance exists individually for each attribute between the ShapeSheet_Type element and the StyleSheet_Type element whose **ID** attribute value is equal to the value of a **LineStyle**, **FillStyle**, or **TextStyle** attribute of the ShapeSheet_Type element.

The **LineStyle**, **FillStyle**, and **TextStyle** attributes of a ShapeSheet_Type element each specify a set of Cell_Type child elements of the StyleSheet_Type element as specified in the following table.

Attribute	Cell_Type elements
LineStyle	Specifies Cell_Type elements related to line properties except for Cell_Type child elements of a FillGradient Section_Type.

Attribute	Cell_Type elements
FillStyle	Specifies Cell_Type elements related to fill properties and effect properties including Cell_Type child elements of a FillGradient Section_Type.
TextStyle	Specifies Cell_Type elements related to text.

2.2.5.4.3 Style-to-Master Inheritance

Masters in a web drawing can be linked to a style, which can affect various properties of the master including its visual appearance. A relationship to such a style is called style-to-master inheritance. A style-to-master inheritance allows a master to take on properties from the style it inherits from. A master can have zero to three style-to-master inheritance relationships.

The style-to-master inheritances in a web drawing are specified by the Master XML Part. Each style-to-master inheritance is specified by the attributes of a ShapeSheet_Type child element of the Shapes_Type descendant element of the MasterContents element in a part. Style-to-master inheritance information is specified by the ShapeSheet_Type element and a StyleSheet_Type child element of a StyleSheets_Type child element of the VisioDocument element in the Document XML Part.

If the **LineStyle**, **FillStyle**, and **TextStyle** attributes of the ShapeSheet_Type element are empty, a style-to-master inheritance is not specified. If the **LineStyle**, **FillStyle**, or **TextStyle** attributes of the ShapeSheet_Type element are not empty, a style-to-master inheritance exists individually for each attribute between the ShapeSheet_Type element and the StyleSheet_Type element whose **ID** attribute value is equal to the value of a **LineStyle**, **FillStyle**, or **TextStyle** attribute of the ShapeSheet_Type element.

The **LineStyle**, **FillStyle**, and **TextStyle** attributes of a ShapeSheet_Type element each specify a set of Cell_Type child elements of the StyleSheet_Type element as specified in the table found in section 2.2.5.4.2.

2.2.5.4.4 Style-to-Style Inheritance

Styles in a web drawing can be linked to other styles, which can affect various properties of the style. A relationship to such a style is called style-to-style inheritance. A style-to-style inheritance allows a style sheet to take on properties from the style it inherits from. A style can have zero to three style-to-style inheritance relationships.

The style-to-style inheritances in a web drawing are specified by the Document XML Part. Each style-to-style inheritance is specified by the attributes of a StyleSheet_Type child element of the StyleSheets_Type child element of the VisioDocument element in a part. Style-to-style inheritance information is specified by the StyleSheet_Type element and other StyleSheet_Type elements in the Document XML Part.

If the **LineStyle**, **FillStyle**, and **TextStyle** attributes of the StyleSheet_Type element are empty, a style-to-style inheritance is not specified. If the **LineStyle**, **FillStyle**, or **TextStyle** attributes of the StyleSheet_Type element are not empty, a style-to-style inheritance exists individually for each attribute between the StyleSheet_Type element and another StyleSheet_Type element whose **ID** attribute value is equal to the value of a **LineStyle**, **FillStyle**, or **TextStyle** attribute of the StyleSheet_Type element.

The **LineStyle**, **FillStyle**, and **TextStyle** attributes of a StyleSheet_Type element each specify a set of Cell_Type child elements of the StyleSheet_Type element as specified in the table found in section 2.2.5.4.2.

2.2.5.4.5 Theme Inheritance

A shape in a web drawing takes on the format properties specified by its dynamic theme components and quick style slices through inheritance. Format properties from the dynamic theme are represented as a dynamic theme style sheet that is linked to from other style sheets, masters, and shapes. A relationship to a dynamic theme style sheet is called theme inheritance.

Theme inheritance allows a style sheet, master, or shape to take on properties from the cells of the dynamic theme style sheet it inherits from. These inheritances in a web drawing are specified as style-to-shape inheritance, style-to-master inheritance, and style-to-style inheritance.

A dynamic theme style sheet in a web drawing is specified by a StyleSheet_Type child element of the StyleSheets_Type child element of the VisioDocument element in the Document XML Part. A dynamic theme style sheet is uniquely identified by a StyleSheet_Type element whose **NameU** attribute is equal to "Theme".

2.2.5.4.6 Local Properties

Sheets corresponding to styles, masters, and shapes in a web drawing can specify that their own properties replace properties taken on from inheritance. These properties are called local properties.

Local properties are specified by Cell_Type, Row_Type, or Section_Type descendant elements of Sheet_Type elements. A local property replaces the properties of an inherited Cell_Type, Row_Type, or Section_Type element, if the value of the local property's **N** attribute is equal to the value of the **N** attribute of the inherited Cell_Type, Row_Type, or Section_Type element.

2.2.5.5 Sheet Extensibility

Sheet extensibility is a mechanism whereby a web drawing specifies extensions to the rules about sections, rows, cells, and function tokens as defined in this specification. Such extensions are specified in SectionDef_Type, RowDef_Type, CellDef_Type, and FunctionDef_Type descendant elements of the Extensions element of the Extensions XML Part.

The valid **N** attributes of a Section_Type element are specified in section 2.4.1. However, SectionDef_Type elements can specify additional valid **N** attributes.

The valid **N** attributes of a Cell_Type element and the locations where a Cell_Type element with a given **N** attribute can occur are specified in section 2.4.4. However, CellDef_Type elements can specify additional valid **N** attributes. Additionally, a CellDef_Type element specifies the valid locations where a CellDef_Type element with a given **N** attribute can occur, based on the CellDef_Type element's ancestor SectionDef_Type and RowDef_Type elements. Cells defined through sheet extensibility are used for formula evaluation only.

The valid function tokens are specified in section 2.5.3. However, FunctionDef_Type elements can specify additional valid function tokens. A function token defined through sheet extensibility consumes all argument, and returns a PtgErr parse token with an error code equal to #VALUE!.

2.2.6 Image

A web drawing can have embedded images. Each embedded image is associated with a shape, which provides information about the image placement, size, and properties.

The ShapeSheet_Type element of a shape that specifies an image MUST have its **Type** attribute equal to "Foreign" and MUST have a ForeignData_Type child element.

The ShapeSheet_Type element specifies the position, width, and height of the image using the ImgOffsetX, ImgOffsetY, ImgWidth, and ImgHeight Cell_Type child elements.

The ShapeSheet_Type element specifies image formatting properties using the Blur, Brightness, Contrast, Denoise, Gamma, and Sharpen, and Transparency Cell_Type child elements.

Additional image properties, such as format and compression, are specified by attributes of the ForeignData_Type child element. This element MUST have a Rel_Type child element that specifies an explicit relationship to another part. The following embedded image formats are supported:

- **bitmap (BMP)**
- **enhanced metafile format (EMF)**
- **Graphics Interchange Format (GIF)**
- **Joint Photographic Experts Group (JPEG)**
- **Portable Network Graphics (PNG)**
- **TIFF**

For these formats, the Rel_Type child element of the ForeignData_Type element MUST specify an Image part that contains the embedded image.

Other embedded image formats and embedded objects are supported using fallback images.

2.2.6.1 Fallback Image

If an Image Part (section 2.3.3.5) that is in a format that is not supported (section 2.2.6) has a relationship to another Image part that is in a supported format, the latter Image part is called a fallback image and is rendered in place of the former. Otherwise, neither Image part is rendered.

2.2.7 Format

A format is a collection of properties that affect the visual appearance of shapes in a web drawing.

2.2.7.1 Fill Properties

A shape, master, or style in a web drawing can possess a variety of properties relating to the visual appearance of fills in closed geometry paths. A collection of properties defining the visual appearance of a shape, master, or style's fill is called a fill property. Each shape, master, or style has one fill property.

Fill properties allow a shape, master, or style to take on a variety of fill styles, including full transparency, solid colors, gradients, and patterns. These properties can be combined with a line property and an effect property.

The fill properties of shapes in a web drawing are specified in the Page XML Part. Each fill property is specified in a ShapeSheet_Type child element of the Shapes_Type descendant element of the PageContents element in a part.

The fill properties of masters in a web drawing are specified in the Master XML Part. Each fill property is specified in a ShapeSheet_Type child element of the Shapes_Type descendant element of the MasterContents element in a part.

The fill properties of styles in a web drawing are specified in the Document XML Part. Each fill property is specified in a StyleSheet_Type child element of the StyleSheets_Type child element of the VisioDocument element in a part.

Fill property information in shapes, masters, and styles is specified by the FillForegnd, FillForegndTrans, FillBkgnd, FillBkgndTrans, FillPattern, FillGradientDir, FillGradientAngle, FillGradientEnabled, RotateGradientWithShape, and UseGroupGradientCell_Type elements, and the Cell_Type elements belonging to the FillGradient Section_Type.

2.2.7.2 Line Properties

A shape, master, or style in a web drawing can possess a variety of properties relating to the visual appearance of lines. A collection of properties defining the visual appearance of a shape, master, or style's line is called a line property. Each shape, master, or style has one line property.

Line properties allow a shape, master, or style to take on a variety of line styles, including full transparency, solid colors, gradients, and strokes. These properties can be combined with a fill property and an effect property.

The line properties of shapes in a web drawing are specified in the Page XML Part. Each line property is specified in a ShapeSheet_Type child element of the Shapes_Type descendant element of the PageContents element in a part.

The line properties of masters in a web drawing are specified in the Master XML Part. Each line property is specified in a ShapeSheet_Type child element of the Shapes_Type descendant element of the MasterContents element in a part.

The line properties of styles in a web drawing are specified in the Document XML Part. Each line property is specified in a StyleSheet_Type child element of the StyleSheets_Type child element of the VisioDocument element in a part.

Line property information in shapes, masters, and styles is specified by the LineColor, LinePattern, LineWeight, LineCap, BeginArrow, EndArrow, LineColorTrans, CompoundType, BeginArrowSize, EndArrowSize, Rounding, LineGradientDir, LineGradientAngle, and LineGradientEnabled Cell_Type elements, and the Cell_Type elements belonging to the LineGradient Section_Type.

2.2.7.3 Effect Properties

A shape, master, or style in a web drawing can possess a variety of properties relating to effects which can affect the visual appearance of the web drawing. Each distinct effect is called an effect set. A collection of properties defining the effect sets of a shape, master, or style is called an effect property. Each shape, master, or style has one effect property consisting of distinct effect sets.

Effect properties allow a shape, master, or style to take on a variety of distinct effect sets, including shadows, bevels, glows, reflections, soft edges, sketch, and 3D rotation. These properties can be combined with a fill property and a line property.

The effect properties of shapes in a web drawing are specified in the Page XML Part. Each effect property is specified in a ShapeSheet_Type child element of the Shapes_Type descendant element of the PageContents element in a part.

The effect properties of masters in a web drawing are specified in the Master XML Part. Each effect property is specified in a ShapeSheet_Type child element of the Shapes_Type descendant element of the MasterContents element in a part.

The effect properties of styles in a web drawing are specified in the Document XML Part. Each effect property is specified in a StyleSheet_Type child element of the StyleSheets_Type child element of the VisioDocument element in a part.

2.2.7.3.1 Shadow Effect Set

A shadow effect set allows a shape, master, or style to take on one of a variety of shadows as cast by light sources of different orientations and brightness. It can be combined with other distinct effect sets. Each shape, master, or style has at most one shadow effect set.

Shadow effect set information in shapes, masters, and styles is specified by the ShdwForegnd, ShdwForegndTrans, ShdwPattern, ShapeShdwType, ShapeShdwOffsetX, ShapeShdwOffsetY, ShapeShdwObliqueAngle, ShapeShdwScaleFactor, and ShapeShdwBlur Cell_Type elements.

Shadow effect set information is partially specified in the Cell_Type elements ShdwType, ShdwOffsetX, ShdwOffsetY, ShdwObliqueAngle, and ShdwScaleFactor, in page sheets.

2.2.7.3.1.1 Shadow Distance

The Euclidean distance between the point specified by the x and y coordinates of a shadow effect set and its origin is called a shadow distance.

The x-coordinate of a shadow effect set applied to a shape, master, or style is specified by the ShapeShdwOffsetX Cell_Type element. The y-coordinate of a shadow effect set applied to a shape, master, or style is specified by the ShapeShdwOffsetY Cell_Type element.

The x-coordinate of a shadow effect set applied to a page sheet is specified by the ShdwOffsetX Cell_Type element. The y-coordinate of a shadow effect set applied to a page sheet is specified by the ShdwOffsetY Cell_Type element.

2.2.7.3.1.2 Page Default Shadow

The parts of a shadow effect set specified in Cell_Type elements in a page sheet are called a page default shadow. A shadow effect set of a shape, master, or style can be partially specified by the page default shadow specified in the page sheet of the drawing page that the shape, master, or style resides on.

If the value of the structure of a ShapeShdwType Cell_Type element in a shape, master, or style is zero, the values of the structures of the ShapeShdwType, ShapeShdwOffsetX, ShapeShdwOffsetY, ShapeShdwObliqueAngle, and ShapeShdwScaleFactor Cell_Type elements of the shape, master, or style are specified by the page default shadow during rendering. These elements are not modified.

The values of the structures of Cell_Type elements of a page sheet relating to a page default shadow specify the values of the structures of Cell_Type elements of a shape, master, or style relating to a shadow effect set according to the following table:

Page sheet Cell_Type elements	Shape, master, or style Cell_Type elements
ShdwType	ShapeShdwType
ShdwOffsetX	ShapeShdwOffsetX
ShdwOffsetY	ShapeShdwOffsetY
ShdwObliqueAngle	ShapeShdwObliqueAngle
ShdwScaleFactor	ShapeShdwScaleFactor

2.2.7.3.2 Bevel Effect Set

A bevel effect set allows a shape, master, or style to take on three-dimensional sloping edges of various types on its top and bottom faces. It can be combined with other distinct effect sets. Each shape, master, or style has at most one bevel effect set.

Bevel effect set information in shapes, masters, and styles is specified by the BevelTopType, BevelTopWidth, BevelTopHeight, BevelBottomType, BevelBottomWidth, BevelBottomHeight, BevelDepthColor, BevelDepthSize, BevelContourColor, BevelContourSize, BevelMaterialType, BevelLightingType, and BevelLightingAngle Cell_Type elements.

2.2.7.3.3 Glow Effect Set

A glow effect set allows a shape, master, or style to take on a colored, blurred outline surrounding the outer edges of the shape, master, or style. It can be combined with other distinct effect sets. Each shape, master, or style has at most one glow effect set.

Glow effect set information in shapes, masters, and styles is specified by the GlowColor, GlowColorTrans, and GlowSize Cell_Type elements.

2.2.7.3.4 Reflection Effect Set

A reflection effect set allows a shape, master, or style to take on a duplicate image of its self, reflected across its bottom edge. Transparency and blur can be applied to the duplicate image to convey the reflective properties of various surfaces. A reflection effect set can be combined with other distinct effect sets. Each shape, master, or style has at most one reflection effect set.

Reflection effect set information in shapes, masters, and styles is specified by the ReflectionSize, ReflectionTrans, ReflectionDist, and ReflectionBlur Cell_Type elements.

2.2.7.3.5 Soft Edges Effect Set

A soft edges effect set allows a shape, master, or style to take on a blur affecting its outer edges. It can be combined with other distinct effect sets. Each shape, master, or style has at most one soft edges effect set.

Soft edges effect set information in shapes, masters, and styles is specified by the SoftEdgesSize Cell_Type element.

2.2.7.3.6 Sketch Effect Set

A sketch effect set allows a shape, master, or style to take on a less polished appearance as if drawn by hand. It cannot be combined with other distinct effect sets. Each shape, master, or style has at most one sketch effect set.

Sketch effect set information in shapes, masters, and styles is specified by the SketchEnabled, SketchSeed, SketchAmount, SketchLineWeight, SketchLineChange, and SketchFillChange Cell_Type elements.

A sketch effect set renders a new geometry path for a shape, master, or style's fill and distorts both the shape, master, or style's geometry path and the fill's geometry by rendering each path segment with randomized perturbations. The geometry section of the shape, master, or style is not modified.

The value of the structure of a SketchSeed Cell_Type element is used to randomize path segment perturbations in both the geometry path and the fill's geometry. If the value of the structure of a SketchSeed Cell_Type element is equivalent for shapes, masters, or styles with identical geometry paths, the shapes, masters, or styles render identical sketch effect sets.

If a sketch effect set is active on a shape, master, or style, other effect sets do not render.

2.2.7.3.7 3D Rotation Effect Set

A 3D rotation effect set allows a shape, master, or style to take on rotations in the z-axis and perspective rotations. A 3D rotation effect set can be combined with other distinct effect sets. Each shape, master, or style has at most one 3D rotation effect set.

3D rotation effect set information in shapes, masters, and styles is specified by the RotationXAngle, RotationYAngle, RotationZAngle, RotationType, Perspective, DistanceFromGround, and KeepTextFlat Cell_Type elements.

2.2.7.4 Dynamic Theme

A shape, master, or style in a Web drawing can specify pre-defined, dynamic sets of properties which can affect its visual appearance. A set of pre-defined, dynamic properties specified in this manner is called a dynamic theme.

A dynamic theme defines properties that specify properties for color, font, fill, line properties, and effect. The properties of a dynamic theme are separated into five distinct groupings called dynamic theme components. A unique set of properties is specified by the combination of the five dynamic theme components. A shape, master, or style specifies distinct dynamic theme components from one or more dynamic themes.

The specified dynamic theme components of a shape, master, or style define more properties than the shape, master, or style can visually express at any one time. A shape, master, or style further specifies subsets of properties which actively affect its visual appearance from its specified dynamic theme components. These subsets of properties are called a quick style.

A quick style defines seven distinct subsets of properties from a shape, master, or style's specified dynamic theme components. A subset is called a quick style slice. The combination of the specified quick style slices and the specified dynamic theme components directly determines the visual appearance of the shape, master, or style.

A dynamic theme defines four distinct sets of pre-defined properties used to indirectly specify the values of properties in quick style slices in a shape, master, or style in a Web drawing. A set is called a dynamic theme variant.

2.2.7.4.1 Dynamic Theme Components

A dynamic theme defines properties that specify color, font, fill, line, and effect. It is composed of multiple parts as specified in [ISO/IEC29500-1:2011] section 20.1.6.9 and this specification.

The properties of a dynamic theme are grouped into five distinct dynamic theme components that are specified in the following table.

Dynamic theme component	Description	Location
Color scheme	Specifies a set of twelve color properties, as specified in [ISO/IEC29500-1:2011] section 20.1.6.2, and one additional color property extension, as specified in [ISO/IEC29500-1:2011] section 18.2.10.	Specified by a **clrScheme** child element as specified by the **CT_ColorScheme** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.2) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part.
Font scheme	Specifies a set of six font properties, as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.18.	Specified by a **fontScheme** child element as specified by the **CT_FontScheme** type (specified in [ISO/IEC29500-1:2011] section 20.1.4.1.18) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part.
Effect scheme	Specifies a set of six quick style slices of fill, line, and effect properties, as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.14. These formats are used in	Specified by an **fmtScheme** child element as specified by the **CT_StyleMatrix** type (specified in [ISO/IEC29500-1:2011] section 20.1.4.1.14) of a **themeElements** child

Dynamic theme component	Description	Location
	non-connector shapes, masters, and styles.	element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part. Additional line properties and sketch effect set information are specified by CT_LineStyle child elements of a CT_SchemeLineStyles child element of a CT_LineStyles child element of an **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an **extLst** child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part. Additional font information is specified by CT_FontProps child elements of a CT_FontStyles child element of a CT_FontStylesGroup child element of an **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an **extLst** child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part.
Connector scheme	Specifies a set of six quick style slices of fill, line, and effect properties, as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.14. These formats are used in connector shapes, masters, and styles.	Specified by an **fmtConnectorScheme** child element as specified by the **CT_StyleMatrix type** (specified in [ISO/IEC29500-1:2011] section 20.1.4.1.14) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part. Additional line properties and sketch effect set information are specified by CT_LineStyle child elements of a CT_SchemeLineStyles child element of a CT_LineStyles child element of an **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an **extLst** child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **themeElements**

Dynamic theme component	Description	Location
		child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part. Additional font information is specified by CT_FontProps child elements of a CT_FontStyles child element of a CT_FontStylesGroup child element of an **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an **extLst** child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part.
Primary scheme	Used in formula evaluation only.	Specified by a CT_ThemeScheme child element of a CT_LineStyles child element of an **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an extLst child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part.

The additional complex types in the following table that are not specified in [ISO/IEC29500-1:2011] partially specify a dynamic theme.

The additional complex types that partially specify a dynamic theme and are not specified in [ISO/IEC29500-1:2011] are listed in the following table.

Complex Type	Description
CT_LineEx	Specifies line properties information in an effect or connector scheme dynamic theme component.
CT_Sketch	Specifies sketch effect set information in an effect or connector scheme dynamic theme component.
CT_SchemeID	Specifies the index of a color, font, effect, connector or primary scheme dynamic theme component, or the GUID of a custom dynamic theme color scheme dynamic theme component.
CT_LineStyle	Specifies line properties and sketch effect set information in an effect or connector scheme dynamic theme component.

Complex Type	Description
CT_LineStyles	Specifies a set of line properties and sketch effect set information in an effect and connector scheme dynamic theme component.
CT_ThemeScheme	Specifies the primary scheme dynamic theme component.
CT_FmtSchemeEx	Specifies the index of an effect scheme dynamic theme component or a connector scheme dynamic theme component.
CT_SchemeLineStyles	Specifies a set of line properties and sketch effect set information in an effect or connector scheme dynamic theme component.
CT_FontProps	Specifies properties used to format a **text run**.
CT_FontStyles	Specifies a set of properties used to format a text run.
CT_FontStylesGroup	Specifies the properties used to format a text run in shapes.
CT_VarClrScheme	Specifies a color scheme list of a dynamic theme variant.
CT_VariationClrSchemeLst	Specifies four distinct color scheme lists of four distinct dynamic theme variants in a dynamic theme.
CT_VariationStyle	Specifies a style property of a style scheme list of a dynamic theme variant.
CT_VariationStyleScheme	Specifies a style scheme list of a dynamic theme variant.
CT_VariationStyleSchemeLst	Specifies four distinct style scheme lists of four distinct dynamic theme variants in a dynamic theme.

2.2.7.4.2 Dynamic Theme Identification

A shape, master, or style in a web drawing can specify distinct dynamic theme components.

The dynamic theme components used in a shape are specified in the Page_XML_part. Each dynamic theme component is specified by a Cell_Type child element of a ShapeSheet_Type child element of a Shapes_Type descendant element of the PageContents element in a part.

The dynamic theme components used in a master are specified in the Master XML part. Each dynamic theme component is specified by a Cell_Type child element of a ShapeSheet_Type child element of a Shapes_Type descendant element of the MasterContents element in a part.

The dynamic theme components used in a style are specified in the Document XML part. Each dynamic theme component is specified by a Cell_Type child element of a StyleSheet_Type child element of a StyleSheets_Type child element of the VisioDocument element in a part.

The location of a dynamic theme component in a shape, master, or style is specified in the following table.

Dynamic theme components	Location
Color scheme	For a shape or master, the color scheme is specified by a ColorSchemeIndex Cell_Type child element of a ShapeSheet_Type element. For a style, specified by a ColorSchemeIndex Cell_Type child element of a StyleSheet_Type element.

Dynamic theme components	Location
Font scheme	For a shape or master, the color scheme is specified by a FontSchemeIndex Cell_Type child element of a ShapeSheet_Type element. For a style, specified by a FontSchemeIndex Cell_Type child element of a StyleSheet_Type element.
Effect scheme	For a shape or master, the effect scheme is specified by an EffectSchemeIndex Cell_Type child element of a ShapeSheet_Type element. For a style, the effect scheme is specified by an EffectSchemeIndex Cell_Type child element of a StyleSheet_Type element.
Connector scheme	For a shape or master, the connector scheme is specified by a ConnectorSchemeIndex Cell_Type child element of a ShapeSheet_Type element. For a style, the connector scheme is specified by a ConnectorSchemeIndex Cell_Type child element of a StyleSheet_Type element.
Primary scheme	For a shape or master, the primary scheme is specified by a ThemeIndex Cell_Type child element of a ShapeSheet_Type element. For a style, the primary scheme is specified by a ThemeIndex Cell_Type child element of a StyleSheet_Type element.

2.2.7.4.3 Quick Style Slices

Quick style slices define properties that specify color, font, fill, line, and effect properties that directly affect the visual appearance of a shape, master, or style. These properties are subsets of the properties provided by the dynamic theme components specified by the shape, master, or style, and are grouped into the seven distinct quick style slices specified in the following table.

Quick style slice	Description
Line matrix	Specifies one of the six quick style slices of line properties from the effect scheme dynamic theme component for non-connector shapes, masters or styles, or from the connector scheme dynamic theme component for connector shapes, masters or styles.
Fill matrix	Specifies one of the six quick style slices of fill properties from the effect scheme dynamic theme component for non-connector shapes, masters or styles, or from the connector scheme dynamic theme component for connector shapes, masters or styles.
Effect matrix	Specifies one of the six quick style slices of effect properties from the effect scheme dynamic theme component for non-connector shapes, masters or styles, or from the connector scheme dynamic theme component for connector shapes, masters or styles.

Quick style slice	Description
Font matrix	Specifies one of six quick style slices of fonts for the font scheme dynamic theme component for shapes, masters, or styles.
Line color	Specifies one of nine colors from the color scheme dynamic theme component.
Fill color	Specifies one of nine colors from the color scheme dynamic theme component.
Shadow color	Specifies one of nine colors from the color scheme dynamic theme component.
Font color	Specifies one of nine colors from the color scheme dynamic theme component.

2.2.7.4.4 Quick Style Identification

A shape, master, or style in a web drawing can specify distinct quick style slices.

The quick style slices of a shape are specified in the Page_XML_part. Each quick style slice is specified by a Cell_Type child element of a ShapeSheet_Type child element of a Shapes_Type descendant element of the PageContents element in a part.

The quick style slices of a master are specified in the Master XML part. Each quick style slice is specified by a Cell_Type child element of a ShapeSheet_Type child element of a Shapes_Type descendant element of the MasterContents element in a part.

The quick style slices of a style are specified in the Document XML part. Each quick style slice is specified by a Cell_Type child element of a StyleSheet_Type child element of a StyleSheets_Type child element of the VisioDocument element in a part.

The location of a quick style slice in a shape, master, or style is specified in the following table.

Quick style slices	Location
Line matrix	For a shape or master, the line matrix is specified by a QuickStyleLineMatrix Cell_Type child element of a ShapeSheet_Type element. For a style, the line matrix is specified by a QuickStyleLineMatrix Cell_Type child element of a StyleSheet_Type element.
Fill matrix	For a shape or master, the fill matrix is specified by a QuickStyleFillMatrix Cell_Type child element of a ShapeSheet_Type element. For a style, the fill matrix is specified by a QuickStyleFillMatrix Cell_Type child element of a StyleSheet_Type element.
Effect matrix	For a shape or master, the effect matrix is specified by a QuickStyleEffectsMatrix Cell_Type child element of a ShapeSheet_Type element.

Quick style slices	Location
	For a style, the effect matrix is specified by a QuickStyleEffectsMatrix Cell_Type child element of a StyleSheet_Type element.
Font matrix	For a shape or master, the font matrix is specified by a QuickStyleFontMatrix Cell_Type child element of a ShapeSheet_Type element. For a style, the font matrix is specified by a QuickStyleFontMatrix Cell_Type child element of a StyleSheet_Type element.
Line color	For a shape or master, the line color is specified by a QuickStyleLineColor Cell_Type child element of a ShapeSheet_Type element. For a style, the line color is specified by a QuickStyleLineColor Cell_Type child element of a StyleSheet_Type element.
Fill color	For a shape or master, the fill color is specified by a QuickStyleFillColor Cell_Type child element of a ShapeSheet_Type element. For a style, the fill color is specified by a QuickStyleFillColor Cell_Type child element of a StyleSheet_Type element.
Shadow color	For a shape or master, the shadow color is specified by a QuickStyleShadowColor Cell_Type child element of a ShapeSheet_Type element. For a style, the shadow color is specified by a QuickStyleShadowColor Cell_Type child element of a StyleSheet_Type element.
Font color	For a shape or master, the font color is specified by a QuickStyleFontColor Cell_Type child element of a ShapeSheet_Type element. For a style, the font color is specified by a QuickStyleFontColor Cell_Type child element of a StyleSheet_Type element.

A QuickStyleType Cell_Type element of a shape, master, or style specifies whether the QuickStyleLineMatrix, QuickStyleFillMatrix, and QuickStyleEffectsMatrix Cell_Type elements of the shape, master, or style refer to the effect or connector scheme dynamic theme component regardless of whether the shape, master, or style is a connector.

2.2.7.4.5 Dynamic Theme Variants

A dynamic theme (section 2.2.7.4) variant defines properties used to indirectly specify the values of properties in quick style slices.

A dynamic theme variant defines properties used to indirectly specify the value of the structure of the QuickStyleLineMatrix, QuickStyleFillMatrix, QuickStyleEffectsMatrix, QuickStyleLineColor, QuickStyleFillColor, QuickStyleShadowColor, QuickStyleFontColor, and QuickStyleFontMatrix Cell_Type elements of a shape, master, or style in a Web drawing. A dynamic theme variant also specifies embellishment and multiformat information.

The properties of a dynamic theme variant are specified in the following table.

Dynamic theme variant property	Description	Location
Color scheme list	Specifies a set of seven color properties used to indirectly specify the value of the structure of the QuickStyleLineColor, QuickStyleFillColor, QuickStyleShadowColor, and QuickStyleFontColor Cell_Type elements of a shape, master, or style in a Web drawing. Specifies multiformat information.	Specified by a CT_VariationClrScheme child element of a CT_VariationClrSchemeLst child element of an **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an **extLst** child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **CT_ColorScheme** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.2) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part. Each color property is specified by a **srgbClr** child element as specified by the **CT_SRGBClr** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.3.32) of a **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 child element of a CT_VariationClrScheme element.
Style scheme list	Specifies a set of four style properties used to indirectly specify the value of the structure of the QuickStyleLineMatrix, QuickStyleFillMatrix, QuickStyleEffectsMatrix, and QuickStyleFontMatrix Cell_Type elements of a shape, master, or style in a Web drawing. Specifies embellishment information.	Specified by a CT_VariationStyleScheme child element of a CT_VariationStyleSchemeLst child element of an **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an **extLst** child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part. Each style property is specified by a CT_VarStyle child element of a CT_VariationStyleScheme type.

2.2.7.4.6 Dynamic Theme Variants Identification

A shape, master, or style in a web drawing can specify distinct dynamic theme variants.

The dynamic theme variants used in a shape are specified in the Page_XML_part. Each dynamic theme variant is specified by a Cell_Type child element of a ShapeSheet_Type child element of a Shapes_Type descendant element of the PageContents element in a part.

The dynamic theme variants used in a master are specified in the Master XML part. Each dynamic theme variant is specified by a Cell_Type child element of a ShapeSheet_Type child element of a Shapes_Type descendant element of the MasterContents element in a part.

The dynamic theme variants used in a style are specified in the Document XML part. Each dynamic theme variant is specified by a Cell_Type child element of a StyleSheet_Type child element of a StyleSheets_Type child element of the VisioDocument element in a part.

The location of the properties of a dynamic theme variant in a shape, master, or style is specified in the following table.

Dynamic theme variant property	Location
Color scheme list	For a shape or master, the color scheme list is specified by a VariationColorIndex Cell_Type child element of a ShapeSheet_Type element. For a style, specified by a VariationColorIndex Cell_Type child element of a StyleSheet_Type element.
Style scheme list	For a shape or master, the style scheme list is specified by a VariationStyleIndex Cell_Type child element of a ShapeSheet_Type element. For a style, specified by a VariationStyleIndex Cell_Type child element of a StyleSheet_Type element.

2.2.7.4.7 Dynamic Theme Functions

The properties specified by a dynamic theme of a shape (section 2.2.3), master, or style are referenced through two function tokens persisted in formula expressions in a web drawing.

The ThemeVal function token, when called without argument, returns the property value from the dynamic theme for the Cell_Type child element that it resides in directly without invoking theme inheritance. The ThemeVal function token, when called with an argument, returns the property value from the dynamic theme specified by the argument directly without invoking theme inheritance.

The ThemeProp function token accepts an argument to retrieve the multiformat and embellishment property values from a dynamic theme (section 2.2.7.4) as specified by the argument.

2.2.7.4.8 Custom Dynamic Theme Color Scheme

The set of color properties in a dynamic theme can be specified by a master instead of a color scheme dynamic theme component. A set of color properties specified in this manner is called a custom dynamic theme color scheme.

The set of color property values in a custom dynamic theme color scheme is specified by the Value Cell_Type child elements of the msvThemeDarkColor, msvThemeLightColor, msvThemeAccentColor, msvThemeAccentColor2.4.3.6, msvThemeAccentColor2.4.3.7, msvThemeAccentColor2.4.3.8, msvThemeAccentColor2.4.3.9, msvThemeAccentColor2.4.3.5 and msvThemeBackgroundColor Row_Type child elements of a User Section_Type element descendant of a MasterContents element of a master.

A custom dynamic theme color scheme is specified by a CT_SchemeID child element of a **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an **extLst** child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **themeElements** element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) in a Theme_XML_part.

If the value of the **schemeEnum** attribute of the CT_SchemeID element is equal to 65535 and the value of the **schemeGUID** attribute of the CT_SchemeID element is equal to the value of the **UniqueID** attribute of the Master_Type child element of a Masters_Type element in the Masters XML part, the custom dynamic theme color scheme is specified by the master with the matching **UniqueID** attribute.

2.2.7.4.9 Connector

A shape, master, or style in a web drawing can be either a connector or a non-connector in terms of a dynamic theme.

If a shape, master, or style inherits from a style sheet whose **NameU** attribute value is equal to "Connector", the shape, master, or style is a connector; otherwise, the shape, master, or style is a non-connector.

2.2.7.4.10 Embellishment and Multiformat

A dynamic theme variant of a dynamic theme specifies two PtgByte parse tokens that are used in formula evaluation only. One is called **embellishment**, and the other is called **multiformat**.

If the value of the structure of the EmbellishmentIndex Cell_Type element of a shape is equal to 0, embellishment is specified by the value of the **embellishment** attribute of a CT_VariationStyleScheme child element, specified by the VariationStyleIndex Cell_Type element of a shape, of a CT_VariationClrSchemeLst child element of an **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an **extLst** child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part. Otherwise, embellishment is specified by the value of the structure of the EmbellishmentIndex Cell_Type element.

Multiformat is specified by the value of the **monotone** attribute of a CT_VariationClrScheme child element, specified by the VariationColorIndex Cell_Type element of a shape, of a CT_VariationClrSchemeLst child element of an **ext** child element as specified by the **CT_OfficeArtExtension** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14) of an **extLst** child element as specified by the **CT_OfficeArtExtensionList** type (specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15) of a **CT_ColorScheme** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.2) of a **themeElements** child element as specified by the **CT_BaseStyles** type (specified in [ISO/IEC29500-1:2011] section 20.1.6.10) of a CT_OfficeStyleSheet element in a Theme_XML_Part.

2.2.7.5 Fixed Theme

A shape, master, or style in a web drawing can specify pre-defined, fixed sets of properties which can affect its visual appearance. A set of pre-defined, fixed properties specified in this manner is called a fixed theme.

A fixed theme defines properties that specify color, font, fill, line, and effect properties. The properties of a fixed theme are separated into two groupings: a fixed color scheme and a fixed effect scheme. A shape, master, or style specifies a fixed theme by specifying a fixed color scheme and a fixed effect scheme.

The set of property values in a fixed color scheme is specified by the vThemeColor custom structure. The set of property values in a fixed effect scheme is specified by the vThemeEffect custom structure.

A fixed color scheme is specified by the Value Cell_Type child element of an msvThemeColors Row_Type child element of a User Section_Type element in a shape, master, or style. A fixed effect

scheme is specified by the Value Cell_Type child element of an msvThemeEffects Row_Type child element of a User Section_Type element in a shape, master, or style.

A Theme function token returns the property values from the set of properties specified by a fixed color scheme and a fixed effect scheme of a shape, master, or style.

2.2.7.5.1 Custom Fixed Color and Effect Schemes

The set of property values specified by a fixed color scheme or a fixed effect scheme can be specified by a master instead of a vThemeColor or vThemeEffect custom structure. A fixed color scheme of this description is called a custom fixed color scheme. A fixed effect scheme of this description is called a custom fixed effect scheme. A shape, master, or style in a web drawing can specify a custom fixed color scheme in place of a fixed color scheme and/or a custom fixed effect scheme in place of a fixed effect scheme.

The set of property values in a custom fixed color scheme is specified by the Value Cell_Type child elements of the msvThemeTextColor, msvThemeFillColor, msvThemeFillColor2.4.3.26, msvThemeLineColor, msvThemeConnectorColor, msvThemeShadowColor, msvThemeAccentColor, msvThemeAccentColor2.4.3.6, msvThemeAccentColor2.4.3.7, msvThemeAccentColor2.4.3.8, msvThemeAccentColor2.4.3.9, and msvThemeBackgroundColor Row_Type child elements of a User Section_Type element in a master.

The set of property values in a custom fixed effect scheme is specified by the Value Cell_Type child elements of the msvThemeLatinFont, msvThemeAsianFont, msvThemeComplexFont, msvThemeLineTransparency, msvThemeLineWeight, msvThemeLinePattern, msvThemeLineRounding, msvThemeConnectorTransparency, msvThemeConnectorPattern, msvThemeConnectorWeight, msvThemeConnectorRounding, msvThemeConnectorBegin, msvThemeConnectorEnd, msvThemeConnectorEnd2.4.3.18, msvThemeConnectorBeginSize, msvThemeConnectorEndSize, msvThemeFillTransparency, msvThemeFillPattern, msvThemeShadowTransparency, msvThemeShadowPattern, msvThemeShadowStyle, msvThemeShadowXOffset, msvThemeShadowYOffset, msvThemeShadowMagnification, and msvThemeShadowDirection Row_Type child elements of a User Section_Type element in a master.

A custom fixed color scheme is specified by the Value Cell_Type child element of an msvThemeColors Row_Type child element of a User Section_Type element in a shape, master, or style. If the **V** attribute of the Value Cell_Type element is equal to 254 and the argument of the USE function token of the **F** attribute of the Value Cell_Type element is equal to the **UniqueID** attribute of the master specified by the Master_Type child element of a Masters_Type element in the Masters XML part, the custom fixed color scheme of the shape, master, or style is specified by the master.

A custom fixed effect scheme is specified by the Value Cell_Type child element of an msvThemeEffects Row_Type child element of a User Section_Type element in a shape, master, or style. If the **V** attribute of the Value Cell_Type element is equal to 254 and the argument of the USE function token of the **F** attribute of the Value Cell_Type element is equal to the **UniqueID** attribute of a master specified by the Master_Type child element of a Masters_Type element in the Masters XML Part, the custom fixed effect scheme of the shape, master, or style is specified by the master.

A Theme function token returns the property values from the set of properties specified by a custom fixed color scheme and/or a custom fixed effect scheme of a shape, master, or style.

2.2.7.6 Color Table

A color value in a web drawing can be specified as either a PtgColorRGB parse token or an unsigned long integer.

If a color value specified as an unsigned long integer is greater than or equal to zero and less than or equal to 23, the color-value of the specified color is specified by the following table.

Unsigned Long Integer	Color-value
0	#000000
1	#FFFFFF
2	#FF0000
3	#00FF00
4	#0000FF
5	#FFFF00
6	#FF00FF
7	#00FFFF
8	#800000
9	#008000
10	#000080
11	#808000
12	#800080
13	#008080
14	#C0C0C0
15	#E6E6E6
16	#CDCDCD
17	#B3B3B3
18	#9A9A9A
19	#808080
20	#666666
21	#4D4D4D
22	#333333
23	#1A1A1A

If a color value specified as an unsigned long integer is greater than 23, the **RGB** value of the specified color is specified by a ColorEntry_Type child element of a Colors_Type child element of a VisioDocument element in a Document XML part. If the value of an **IX** attribute of a ColorEntry_Type element is equal to the specified unsigned long integer, the RGB value of the specified color is equal to the **RGB** attribute specified by the ColorEntry_Type element.

2.2.7.7 Font Table

A font table specifies the **fonts** used in a web drawing. It is specified by a FaceNames_Type child element of a VisioDocument element in a Document XML part. Each font is specified by a FaceName_Type child element of the FaceNames_Type element.

A shape, master, or style specifies its fonts using the Font, AsianFont, and ComplexScriptFont Cell_Type elements. If the value of the **V** attribute of a Font, AsianFont, or ComplexFont Cell_Type element of a shape, master, or style is equal to the value of the **NameU** attribute of a FaceName_Type element, the shape, master, or style specifies the font specified by the FaceName_Type element.

2.2.7.8 Custom Pattern

A fill, line, or line end in a shape, master, or style in a web drawing can be specified by a master. A fill, line, or line end specified in this way is called a custom pattern.

Custom patterns are specified by a Masters XML part. Each custom pattern is specified in a Master_Type child element of a Masters_Type element in a part. The **PatternFlags** attribute of a Master_Type element specifies the type and behavior of the custom pattern.

The visual appearance of a custom pattern is specified by the shapes in the Master XML part that corresponds to the master. Each shape is specified by a ShapeSheet_Type child element of the Shapes_Type descendant element of the MasterContents element in a part.

A shape, master, or style in a web drawing specifies a custom pattern according to the following table.

Custom pattern	Description
Fill	Specified by a FillPattern Cell_Type element whose **V** attribute value is equal to 254. If the argument of the USE function token of the **F** attribute of the FillPattern Cell_Type element is equal to the **NameU** attribute of a master specified by a Master_Type element, the shape, master or style specifies a fill custom pattern specified by a Master_Type element.
Line	Specified by a LinePattern Cell_Type element whose **V** attribute value is equal to 254. If the argument of the USE function token of the **F** attribute of the LinePattern Cell_Type element is equal to the **NameU** attribute of a master specified by a Master_Type element, the shape, master, or style specifies a line custom pattern specified by a Master_Type element.
Line end	Specified by a BeginArrow or EndArrow Cell_Type element whose **V** attribute value is equal to 254. If the argument of the USE function token of the **F** attribute of the BeginArrow or EndArrow Cell_Type element is equal to the **NameU** attribute of a master specified by a Master_Type element, the shape, master, or style specifies a line end custom pattern specified by a Master_Type element.

2.2.7.9 Data Formatting

The text field or shape data in a shape or master in a web drawing can specify a format that affects the visual appearance of its field. Formatting specified in this manner is called a data format.

2.2.7.9.1 Text Field Data Formatting

The text field in a shape or master in a web drawing can specify a data format that affects the visual appearance of its field that is used in a text run.

A text field data format is specified in a Field Section_Type element in a shape or master. Each data format is specified by a Row_Type child element of the Field Section_Type element. A Value Cell_Type

child element of the Row_Type element specifies a value to be formatted. A Format Cell_Type child element of the Row_Type element specifies the data format to apply to the value specified by its sibling Value Cell_Type.

The **V** attribute of the Format Cell_Type element is a vFormatString custom structure that specifies the data format information designating how the value is displayed.

2.2.7.9.2 Shape Data Formatting

The shape data in a shape or master in a web drawing can specify a data format that affects the visual appearance of its shape data field.

The shape data formats are specified in a Property Section_Type element in a shape or master. The data format for a shape data field is specified by its corresponding Row_Type child element of the Property Section_Type element. A Value Cell_Type child element of the Row_Type element specifies a shape data field value to be formatted.

The **V** attribute of the Value Cell_Type element specifies the shape data field value. A Type Cell_Type child element of the Row_Type element specifies the type of shape data field value that is stored in its sibling Value Cell_Type element. The **V** attribute of the Type Cell_Type element is a vDataType custom structure that specifies the shape data field value type. A Format Cell_Type child element of the Row_Type element specifies the data format to apply to shape data field value specified by its sibling Value Cell_Type. The **V** attribute of the Format Cell_Type element is a vFormatString custom structure that specifies the data format information designating how the shape data field value is displayed.

2.2.8 Text

A shape or master can contain text that is specified by one or more text runs. The text runs associated with a shape are specified by the contents of a Text_Type element contained in the ShapeSheet_Type element of the shape. The characters in a text run can be specified explicitly or can be a reference to a text field.

A text run has characters and properties of character, paragraph, and tabs specified as follows:

- Character properties are specified by cp_Type elements.
- Paragraph properties are specified by pp_Type elements.
- Tabs properties are specified by tp_Type elements.
- Text fields are specified by fld_Type elements.

The content of a Text_Type element is composed of the text characters associated with the shape, interspersed with cp_Type, pp_Type, tp_Type, and fld_Type elements.

The beginning of a text run on a shape is specified by a Text_Type, cp_Type, pp_Type, or tp_Type element.

2.2.8.1 Character Properties

The cp_Type element in a shape or master specifies the beginning of a text run and the set of character properties used for the text run. These character properties are used until the end of the Text_Type element, or until another cp_Type element specifies new character properties.

The cp_Type element specifies the index of a Row_Type element that is contained in a Character Section_Type element. This Row_Type element specifies the information about the character properties using a collection of Cell_Type elements. It is either contained under the ShapeSheet_Type element for the shape or it is inherited.

A collection of Cell_Type elements that define the character properties are composed of AsianFont, Case, Color, ColorTrans, ComplexScriptFont, ComplexScriptSize, DblUnderline, DoubleStrikethrough, Font, FontScale, LangID, Letterspace, Overline, Pos, Size, Strikethru, and Style Cell_Type elements.

2.2.8.2 Paragraph Properties

The pp_Type element in a shape or master specifies the beginning of a text run and the set of paragraph properties used for the text run. These paragraph properties are used until the end of the Text_Type element, or until another pp_Type element specifies new paragraph properties.

The pp_Type element specifies the index of a Row_Type element that is contained in a Paragraph Section_Type element. This Row_Type element specifies the information about the paragraph properties using a collection of Cell_Type elements. It is either contained under the ShapeSheet_Type element for the shape or it is inherited.

A collection of Cell_Type elements that define the paragraph properties are composed of Bullet, BulletFont, BulletFontSize, BulletStr, Flags, HorzAlign, IndFirst, IndLeft, IndRight, SpAfter, SpBefore, SpLine, and TextPosAfterBullet Cell_Type elements.

2.2.8.3 Tabs Properties

The tp_Type element in a shape or master specifies the beginning of a text run and the set of tab stops used for the text run. These tab stops are used until the end of the Text_Type element, or until another tp_Type element specifies new tab stops.

The tp_Type element specifies the index of a Row_Type element that is contained in a Tabs Section_Type element. This Row_Type element specifies the information about the tab stops using a collection of Cell_Type elements. It is either contained under the ShapeSheet_Type element for the shape or it is inherited.

A Row_Type element in a Tabs Section_Type element contains a series of Position and Alignment Cell_Type element pairs with **N** attributes equal to Position# and Alignment#, where the # represents the tab stop index. A Position and Alignment pair specifies the stop position and alignment for a single tab stop.

2.2.8.4 Text Fields

The fld_Type element in a shape or master specifies a field that is used in a text run. It specifies the index of a Row_Type element that is contained in a Field Section_Type element. This Row_Type element specifies the information about the field using a collection of Cell_Type elements. It is either contained under the ShapeSheet_Type element for the shape or it is inherited.

If the value of the **IX** attribute of a fld_Type element is equal to the value of the **IX** attribute of a Row_Type element that is contained in a Field Section_Type element in the shape or master, the Cell_Type elements contained under the Row_Type element specify information about the field of the fld_Type element.

A collection of Cell_Type elements that define a text field composed of Calendar, Format, ObjectKind, Type, and Value Cell_Type elements. The Value Cell_Type element specifies the value of the field. The Calendar, Format, ObjectKind, and Type Cell_Type elements specify how the value of the field is displayed in the text run.

2.2.8.5 Text Block

The text runs associated with a shape are rendered using a rectangular composition area called a text block. A text block specifies information related to the visual appearance of the text runs as a whole.

A text block uses a collection of Cell_Type elements contained under the ShapeSheet_Type element for a shape to specify position, transform, margin, alignment, direction, and background information for the text runs associated with the shape. A collection of Cell_Type elements that define how text is arranged in the composition area of a text block that is detailed in the following table.

Cell_Type element(s)	Description
TxtPinX, TxtPinY, TxtLocPinX, and TxtLocPinY	Specifies the text block coordinate system.
TxtAngle	Specifies the angle of counterclockwise rotation of the text block in the coordinate system of the shape it is associated with.
TxtWidth and TxtHeight	Specifies the width and height of the text block.
LeftMargin, RightMargin, TopMargin, and BottomMargin	Specifies the positioning of the text runs against the borders of the text block.
TextDirection	Specifies whether the text runs are rendered in an upright alignment with the top border of the text block or in an upright alignment with the right border of the text block within the text block coordinate system.
VerticalAlign	Specifies the vertical alignment of the text runs. If the value of the TextDirection Cell_Type element structure is equal to zero, text runs are rendered starting from the top border, middle, or bottom border of the text block within the text block coordinate system. If the value of the TextDirection Cell_Type element structure is equal to one, text runs are rendered starting from the right border, center, or left border of the text block with the text block coordinate system.
TextBkgnd	Specifies the solid fill color property of the background of the text block.
TextBkgndTrans	Specifies the transparency level of the solid fill color property of the background of the text block.

2.2.8.5.1 Text Block Coordinate System

A point on a text block is specified by coordinates on a two-dimensional Cartesian plane where the x-coordinate specifies the horizontal position and the y-coordinate specifies the vertical position. Every text block defines a coordinate system.

The TxtPinX and TxtPinY Cell_Type child elements of a ShapeSheet_Type element of a shape specify the pin of the text block in the coordinate system of the shape. The TxtLocPinX and TxtLocPinY Cell_Type child elements of a ShapeSheet_Type element of a shape specify the pin of the block in local coordinates.

A point on a text block specified in local coordinates can be converted into its associated shape coordinates by applying transformations in the following order:

1. Subtract the value of the TxtLocPinX Cell_Type element from the x-coordinate.
2. Subtract the value of the TxtLocPinY Cell_Type element from the y-coordinate.
3. Mirror the point about the y-axis if the value of the FlipX Cell_Type element is equal to one.
4. Mirror the point about the x-axis if the value of the FlipY Cell_Type element is equal to one.
5. Rotate the point counterclockwise around the origin by the value of the TxtAngle Cell_Type element.
6. Add the value of the TxtPinX Cell_Type element to the x-coordinate.
7. Add the value of the TxtPinY Cell_Type element to the y-coordinate.

2.2.9 Comments

Comments are plain text annotations in a web drawing. Each comment has an associated author and drawing page. It can have an associated shape on the drawing page. A collection of comments in a web drawing is specified by a Comments XML part.

A Comments_Type element in a Comments XML part contains the AuthorList_Type element and the CommentList_Type elements, which specifies the comment authors and comments respectively.

Each AuthorEntry_Type child of an AuthorList_Type parent element contains information for a single author. An author can be associated with one or more comments. The author is uniquely identified by the **ID** and **ResolutionID** attributes. Additional author information is provided by the **Name** and **Initials** attributes.

Each CommentEntry_Type child of a CommentList_Type parent element represents a single comment. The text runs associated with a comment are specified by the contents of a CommentEntry_Type element. The following attributes specify additional properties of the comment:

- The **AuthorID** attribute specifies the author of a comment. This attribute is equal to the **ID** attribute of the AuthorEntry_Type element that corresponds to the author.
- The **PageID** attribute specifies the page a comment refers to. This attribute is equal to the **ID** attribute of a Page_Type element of the drawing page.
- The **ShapeID** attribute can specify a shape on the drawing page that the comment refers to. When the **ShapeID** attribute exists, it is equal to the **ID** attribute of the ShapeSheet_Type element of the shape.

2.2.10 Data Connectivity and Refresh

This section describes how data sources can be referenced, queried and connected to from within a web drawing.

2.2.10.1 Data Connections

A web drawing can be linked to data in databases and other data sources which can affect various attributes of the web drawing, including its visual appearance. A relationship to such data sources is called a data connection.

A data connection contains properties that specify how the application connects to and queries the data source, including the type of **data provider** (for example, **OLE DB** or **ODBC**) required to access

a data source, the name of the server on which the data source is hosted, security information to access the data source, and a **query** to execute on the server.

The data connections in a web drawing are specified by the Connections XML part. Each data connection is specified by a DataConnection_Type child element of the DataConnections element in a part.

Data connection information can be specified solely by the DataConnection_Type element or partially by information in an external file. If the **FileName** attribute of the DataConnection_Type element is empty, a data connection is solely specified by the DataConnection_Type element. If the **FileName** attribute is not empty, a data connection is specified by the DataConnection_Type element and the information contained in the file found at the path described by the value of the **FileName** attribute.

The following elements in parts of a web drawing specify supplementary information about the data connection.

- A DataRecordSet_Type element in the Recordsets XML Part contains a **ConnectionID** attribute that is equal to the **ID** attribute of the DataConnection_Type element for the data connection and specifies a recordset that uses this data connection to connect to and query a data source.

Data connections can be established for the types of data sources listed in the **ConnectionString** attribute of the DataConnection_Type element.

2.2.10.2 Recordset

A recordset is the data that is returned from a data source, organized into sets of rows and fields. The recordset is related to a specific data source using a data connection. The operation of replacing the contents of a recordset with data queried from a data source, using the associated data connection, is called refreshing the recordset.

The rows of a recordset can be linked to shapes in drawing pages in a web drawing through data binding. This allows additional properties of the web drawing to be updated when the recordset is refreshed.

The recordsets in a web drawing are specified by the Recordsets XML part. Each recordset is specified by a DataRecordSet_Type child element of the DataRecordSets element in a part. The fields of the recordset are specified by the DataColumns_Type child element of the DataRecordSet_Type element.

2.2.10.2.1 Data Binding

Data binding is the association between a row of a recordset and a shape in a drawing page. A row can be bound to zero or more shapes. A shape can have zero or one recordset rows bound to it.

The rows of a recordset that are bound to shapes are specified by the RowMap_Type child elements of the DataRecordSet_Type element for the recordset. In each RowMap_Type element, the row is identified by a **RowID** attribute, the shape is identified by a **ShapeID** attribute, and the drawing page containing the shape is identified by a **PageID** attribute.

The fields of a recordset are mapped to shape data items in the bound shapes. A field can be mapped to zero or one shape data item in each shape that is bound to a row in the recordset. A shape data item can have zero or one fields bound to it.

The mapping between each field of the recordset and shape data item of the bound shape is specified by the DataColumn_Type element for the field and a Row_Type child element of the Property Section_Type element for the shape. A mapping exists if there is a Row_Type element with an **N** attribute that is equal to the **Name** attribute of the DataColumn_Type element.

2.2.10.3 Recordset Refresh

A recordset refresh is the operation of replacing the contents of a recordset with data queried from a data source. Refresh information is specified by the recordset and its associated data connection.

The refresh query is specified by the **Command** attribute of the DataRecordSet_Type element for the recordset. If the **Command** attribute of the DataRecordSet_Type element is empty, the refresh query is specified by the **Command** attribute of the DataConnection_Type element for the data connection.

Only recordsets that are enabled for data refresh participate in refresh operations. A recordset is enabled for data refresh when both of the following conditions are true:

- A PublishSettings_Type child element of the VisioDocument_Type element for the web drawing is missing, or the PublishSettings_Type element contains a RefreshableData_Type child element with an **ID** attribute equal to the **ID** attribute of the DataRecordSet_Type element for the recordset.
- The DataRecordSet_Type element for the recordset contains an **Options** attribute with a value that is not a bitwise OR combination of the value one.

When the data in the rows of a recordset change, shape data in shapes with data bindings to the recordset are also updated. The **RefreshOverwriteAll** attribute of the DataRecordSet_Type element for the recordset determines which shape data items are updated. Individual shape data items are then updated in the following manner:

- If the cell associated with the shape data item contains a formula expression containing a Guard function token, the shape data item is not updated.
- If the cell associated with the shape data item contains a formula expression containing a SetAtRef function token, the value of the cell referenced by the first argument of the function is updated with the value of the recordset for the mapped field and row.
- Otherwise, the value of the shape data item is updated with the value of the recordset for the mapped field and row. This could involve a data type conversion from the data type of the field, as specified by the **DataType** attribute of its corresponding DataColumn_Type element, to the data type of the shape data item.

All formulas in cells that have been updated are recalculated as part of a diagram update.

2.2.10.4 Recordset Row Addressing

Specific recordset rows are tracked across a recordset refresh operation using a **primary key**. A recordset can explicitly specify a primary key, or it can specify that the current ordering of the rows be used as a primary key.

If the **RowOrder** attribute of the DataRecordSet_Type element for the recordset is zero, the primary key is specified by a collection of PrimaryKey_Type child elements of the DataRecordSet_Type element. If the **RowOrder** attribute is one, the primary key is specified by the position of each row in the recordset regardless of its contents.

2.2.11 Diagram Update

This section describes how the properties of a web drawing are changed from their current state to an updated state by a diagram update operation. A diagram update is initiated following a recordset refresh or through update triggers. These actions each specify a set of properties to change.

Additional properties of the web drawing can have formulas that are dependent on the initial set of updated properties. Expressions in the formulas are evaluated to calculate new property values.

2.2.11.1 Update Triggers

An update trigger is a structure in a web drawing that signifies a diagram update is needed. The trigger is specified by the presence of a special function token in the formula expression of a property of the web drawing.

The Category, Creator, Description, Directory, DocLastEdit, DocLastSave, Keywords, Now, Subject, and Title function tokens specify update triggers.

The Trigger_Type element specifies one or more drawing pages that contain a specific update trigger. The **N** attribute of the Trigger_Type element determines the update trigger and the possible values for the **N** attribute are defined in the Triggers section of this specification.

2.2.11.2 Formulas

The properties that are specified in cells can have formulas. Formulas specify how the properties of a web drawing are modified during a diagram update operation.

A formula is specified by the **F** attribute of a Cell_Type child element in a Section_Type, Row_Type, ShapeSheet_Type, PageSheet_Type, StyleSheet_Type, or DocumentSheet_Type element.

The following sections describe the concepts and elements of a formula.

2.2.11.2.1 Formula Expression

A formula expression is a sequence of functions, values, and references that make up a formula and that produce a value when evaluated.

A formula expression contains a sequence of parse tokens. The Formula ABNF and Full Grammar Definition section in this specification defines the valid formula expressions in a web drawing.

2.2.11.2.2 Parse Tokens

A parse token is a string of characters that specifies a **token** in a formula expression. A parse token in a web drawing is a function, an operand, or a reference token.

2.2.11.2.2.1 Function Tokens

A function token represents a function in a formula expression. The Formula ABNF and Full Grammar Definition section in this specification defines the valid function tokens in a formula expression. The syntax for each function token is described in the Function Token Definitions section.

A function can specify a set of arguments used in the evaluation of the function token. The arguments of a function are additional parse tokens in the formula expression. The value returned by an evaluated function is an operand token.

2.2.11.2.2.2 Operand Tokens

An operand token represents a value in a formula expression. This token can be either the solitary value in a formula, an argument of a function, the evaluation result of a function, or the evaluation result of a cell reference.

The Formula ABNF and Full Grammar Definition section in this specification defines the valid operand tokens in a formula expression. The syntax for each operand token is described in the Parse Token Definitions section.

In addition to its use in a formula expression, an operand token also specifies a single value that can be persisted in the file and represents one of the tokens specified in the token group vAny.

An operand token can have **Value**, **Unit**, **Dimension**, **Currency**, and **Error State** properties.

The **Value** of an operand token is the value of the structure. When stored in a Cell_Type element, the **Value** of an operand token is stored in the **V** attribute with the following exceptions:

- For a Boolean value, the **Value** is "FALSE" or "TRUE" but is stored as zero or one, respectively, in the Cell_Type element **V** attribute.
- For a currency value, both the **Value** and **Currency** are stored in the **V** attribute.
- For a multi-dimensional value, the **Value**, **Unit**, and **Dimension** are stored in the **V** attribute.
- For a two-dimensional point, the **Value** and **Unit** are stored in the **V** attribute.
- For an error code, the operand token has no **Value**.

The **Unit**, **Dimension**, and **Currency** of an operand token give additional meaning to the token's **Value**. Not all operand tokens have a **Unit**, **Dimension**, or **Currency**. When stored in a Cell_Type element, the **Unit** of an operand token is stored in the **U** attribute.

A **Dimension** is not persisted unless the token is a PtgNumMultiDim. For a PtgNumMultiDim, the **Value** and the **Dimension** are stored in the **V** attribute as specified by the PtgNumMultiDim format.

Currency values are the only operand tokens to have a **Currency**. For a currency value, the **Value** is concatenated with the **Currency** and stored in the **V** attribute as specified by the PtgCy parse token format.

The **Error State** of an operand token represents an error obtained during formula evaluation. Depending on the function, the **Error State** of an operand token can either be used or ignored during formula evaluation. When stored in a Cell_Type element, the **Error State** of an operand token is stored in the **E** attribute.

An operand token represents, and can be converted into, one of the following types of values.

- A string value
- A numeric value
- A Boolean value
- A currency value
- A color value
- A date value
- A geometry function value
- An error value

These conversions translate many different source operand tokens into tokens representing different classes of inputs that are required by functions. Functions can operate on the converted tokens but can also refer to elements of the source token. See the Custom Input Types section for details on common token conversions used by functions.

2.2.11.2.2.2.1 String Values

A string value represents textual information and is specified as a PtgString parse token. For a string operand token, the **Value** property is the string and the **Unit** property is equal to "STR". The token does not have a **Dimension** or **Currency** property.

Other tokens can specify a string value according to the conversion specified in the vString custom input type.

2.2.11.2.2.2.2 Numeric Values

A numeric value represents a number with or without units. Boolean values, currency values, color values, and date values are classified separately.

A numeric value is specified as one of the tokens in the custom token group vNum (except PtgDate). Other token types can also represent numeric values as specified in the vDouble, vFloat, vSignedInt, vSignedLong, vUnsignedInt, and vUnsignedLong custom input types.

Numeric values that represent length, angle, duration, and typographic units, as well as higher dimensional forms of these units, are described in the Unit Number section. These numeric values have the special property so that their **Value** property is specified as a Custom Internal Unit Type. When found in a formula expression, the **Value** is converted to the **Unit** and **Dimension** properties of the operand token; this is called the display value. During formula evaluation, the operand token **Value** (not the display value) from the formula expression is used.

A numeric value that represents a percentage value is specified as a PtgNumPct parse token. The **Value** is a number as a fraction of 100, the **Unit** is equal to "PER", and the **Dimension** is zero.

If the numeric value has no units, it represents a number and is persisted in the file as a PtgNum parse token or equivalent member of the vScalar custom token group. The **Value** is equal to the numeric value, and the **Dimension** is zero. It does not have a **Unit**.

2.2.11.2.2.2.3 Boolean Values

A value that represents a Boolean value is specified as a PtgBool parse token. The **Value** property is either "FALSE" or "TRUE", the **Unit** is equal to "BOOL" or does not exist, and the **Dimension** property is zero. It does not have a **Currency** property.

When stored in a Cell_Type element, the **Value** of an operand token is converted to zero or one, where zero represents "FALSE" and one represents "TRUE", and is stored in the **V** attribute.

Other tokens can also represent a Boolean value according to the conversion specified in the vBoolean custom input type.

2.2.11.2.2.2.4 Currency Values

A value that represents a currency is specified as a PtgCy parse token. No other token type can represent a currency value. The only custom input type that preserves both a currency value and its associated currency is vDoubleEx.

The **Value** is the numeric value of the currency, the **Currency** is the associated currency string as specified in vCurrency custom structure, the **Unit** is equal to "CY" and the **Dimension** is zero.

2.2.11.2.2.2.5 Color Values

A value that represents a red-green-blue (RGB) color value is specified as a PtgColorRGB parse token. The **Value** represents the hexadecimal value of the color or the index in the color table, the **Unit** property is equal to "COLOR" or does not exist, and the **Dimension** property is zero. It does not have a **Currency** property.

Other tokens can also represent a color value according to the conversion specified in the vColor custom input type.

2.2.11.2.2.2.6 Date Values

A value that represents a date is specified as a PtgDate parse token. The **Value** property is a date and time of day, the **Unit** property is equal to "DATE", and the **Dimension** property is one. It does not have a **Currency** property.

Other token types can represent a date according to the conversion specified in the DateTime function.

2.2.11.2.2.2.7 Geometry Function Values

A geometry function value represents geometry path data that is specified by a PtgPnt, PtgNURBS, or PtgPolyline parse token. The **Value** property is a set of numeric values that specify individual properties of the geometry path. These numeric values are arranged in a syntax that matches the Pnt, NURBS, and Polyline function token definitions.

The **Unit** property of the token is equal to "PNT" for a PtgPnt or "POLYLINE" for a PtgPolyline; the PtgNURBS does not have a **Unit**. The **Dimension** property is zero, and it does not have a **Currency** property. No other token types can represent a geometry function value.

2.2.11.2.2.2.8 Error Values

An error code that is returned as a result of a formula evaluation is specified as a PtgErr parse token. When a function encounters a PtgErr as one of its arguments, it returns the same error value. The exceptions are the functions IsErr, IsErrNA, IsError, and IsErrValue, which are specifically designed to detect specific error values.

2.2.11.2.2.3 Reference Tokens

A reference token represents a cell, other than the cell containing the formula expression, whose value is used in the evaluation of a formula expression. A reference token allows a formula expression to depend on the values of other properties in the web drawing.

The Formula ABNF and Full Grammar Definition section defines the valid reference tokens in a formula expression. The syntax for each reference token is described in the Reference Token Definitions section.

The result of a reference token that is evaluated is an operand token.

2.2.11.2.3 Formula Evaluation

Formula evaluation is the process of taking a complex formula expression and computing a single resulting parse token.

The parse tokens that make up the formula expression are evaluated in sequence as specified by the Order of Operations. Each function token and reference token in the formula expression is evaluated to produce an operand token.

The logic for evaluating a particular function token is specified by the Function Token Definitions. A reference token is evaluated by returning the value of the cell specified by the reference token. Functions and references are evaluated within a reference context, which is the specification of the sheet containing the properties to be used in the evaluation.

When the formula expression of a cell is evaluated, the formula expressions of other cells that contain reference tokens that reference the cell are also evaluated.

2.2.11.2.4 Reference Context

A reference context is the sheet containing the properties to be used in the evaluation of a function token or reference token.

The reference context can vary for each function token or reference token in a formula expression. The current reference context is the context used for the current token being evaluated.

The CrossPageRef, DocSheetRef, MasterSheetRef, PageSheetRef, ShapeSheetRef, and StyleSheetRef reference tokens specify the reference context of the function token or reference token that immediately follows them in the formula expression. If one of these reference tokens does not precede the token to be evaluated, the default reference context is the sheet containing the formula expression that contains the function token or reference token.

2.2.11.3 Unit Number

A unit number is a numeric value with a unit of measure. Unit numbers represent length, angle, duration and typographic units, higher dimensional forms of these units, and dates.

All unit numbers have a **Dimension** property. One-dimensional unit numbers are used to represent length, angle, duration, and typographic measurements. Two-dimensional units are used to represent area measurements, and three-dimensional units are used to represent volume measurements. A numeric value that has a **Dimension** greater than one is called a multidimensional unit number.

The **Value** property of a one-dimensional unit number is specified as a Custom Internal Unit Type or a date and time as specified by a PtgDate parse token. The **Value** of a multidimensional unit number is specified as a Custom Internal Unit Type for the PtgAcre and PtgHectare operand tokens or as a value as specified by the PtgNumMultiDim parse token.

For numeric values where the **Value** is expressed as a Custom Internal Unit Types, the **Unit** property determines how the numeric value is formatted and displayed in a formula expression, or the user interface. When found in a formula expression or the user interface, the **Value** is converted to the **Unit** and **Dimension** of the operand token; this is called the display value. During formula evaluation, the operand token **Value**, not the display value from the formula expression, is used.

2.2.11.3.1 One-dimensional Unit Number

If the numeric value represents a length or distance measurement, the **Value** property is expressed as a lengthInternalUnitNumber custom internal unit type. The operand tokens that represent length or distance measurements are PtgNumCM, PtgNumDft, PtgNumF, PtgNumFI, PtgNumI, PtgNumKM, PtgNumM, PtgNumMI, PtgNumMM, PtgNumNM, and PtgNumYards.

If the numeric value represents an angle measurement, the **Value** is expressed as an angleInternalUnitNumber custom internal unit type. The operand tokens that represent angles are specified in the vAngle custom token grouping.

If the numeric value represents a duration measurement, the **Value** is expressed as a durationInternalUnitNumber custom internal unit type. The operand tokens that represent durations are PtgEDay, PtgEHour, PtgEMin, PtgESec, PtgEWeek, and PtgTDurDft.

If the numeric value represents a length measurement used in typography, the **Value** is expressed as a typographicInternalUnitNumber custom internal unit type. The operand tokens that represent typographic measurements are PtgTypCD, PtgTypCi, PtgTypDft, PtgTypDi, PtgTypPi, PtgTypPP, and PtgTypPt.

A numeric value with units specified as a PtgPageDft parse token indicates that the internal units are determined by the default values of the drawing page, as specified by PtgPageDft. The **Value** is a number expressed as a Custom Internal Unit Types. For this operand token, the **Unit** of the numeric value is not specified in the PtgPageDft token itself. It is computed as specified by PtgPageDft, and is determined by the default values of the drawing page.

If a numeric value represents a date value, the **Value** is expressed as a date and time of day, in complete extended format, as specified in [ISO-8601] section 4.3.2. The operand token that represents dates is a PtgDate parse token.

The **Value** and **Unit** properties for each unit number are described in the Parse Token Definitions section. The **Dimension** property of one-dimensional unit numbers is equal to one. They do not have a **Currency** property.

2.2.11.3.2 Multidimensional Unit Number

If the numeric value represents an acre or hectare, the **Value** property is expressed as the square of a lengthInternalUnitNumber custom internal unit type. The operand tokens that represent these measurements are PtgAcre and PtgHectare, respectively.

Higher dimensional forms of other unit numbers are specified as a PtgNumMultiDim operand token. The **Value**, **Unit**, and **Dimension** properties of the unit number are stored in the **V** attribute of the Cell_Type element containing the token as specified by the PtgNumMultiDim format.

Multidimensional unit numbers do not have a **Currency** property.

2.3 Parts

The Parts sections that follow specify the structure of the parts that are in the ZIP archive of a web drawing.

2.3.1 Part Enumeration

The web drawing contains the following ZIP package parts and relationships.

Part Name	Relationship between Source and Target Resource	Root Element
App	package	Specified outside this document
Comments	Document	Comments
Connections	Document	DataConnections
Content Type	package	Specified outside this document
Core	package	Specified outside this document
Custom	package	Specified outside this document
Document	package	VisioDocument
Extensions	Document	Extensions
Image	Image or Page	Specified outside this document
Master	Masters	MasterContents
Masters	Document	Masters
Page	Pages	PageContents
Pages	Document	Pages
Recordsets	Document	DataRecordSets
Rels	package	Specified outside this document
Theme	Document	Theme

All other parts are unused and MUST be ignored.

2.3.2 Shared XML Parts and Schema

The Shared XML Parts and Schema sections that follow list the parts in a web drawing that are specified outside this document in their entirety.

2.3.2.1 App XML Part

The App XML part is specified in [ISO/IEC29500-1:2011] section 15.2.12.3.

This is an optional part that specifies the **Extended Properties** of a web drawing, as specified by [ISO/IEC29500-1:2011] section 22.2.

The following properties in the App XML part are defined in a web drawing.

Property Name	Specified in
Application	[ISO/IEC29500-1:2011] section 22.2.2.1
AppVersion	[ISO/IEC29500-1:2011] section 22.2.2.2
Company	[ISO/IEC29500-1:2011] section 22.2.2.5
HeadingPairs	[ISO/IEC29500-1:2011] section 22.2.2.8
HyperlinkBase	[ISO/IEC29500-1:2011] section 22.2.2.11
HyperlinksChanged	[ISO/IEC29500-1:2011] section 22.2.2.12
LinksUpToDate	[ISO/IEC29500-1:2011] section 22.2.2.14
Manager	[ISO/IEC29500-1:2011] section 22.2.2.15
ScaleCrop	[ISO/IEC29500-1:2011] section 22.2.2.22
SharedDoc	[ISO/IEC29500-1:2011] section 22.2.2.23
Template	[ISO/IEC29500-1:2011] section 22.2.2.25
TitlesOfParts	[ISO/IEC29500-1:2011] section 22.2.2.26

2.3.2.2 ContentType XML Part

The ContentType XML part and its syntax are specified in [ISO/IEC29500-2:2011] section 10.1.2.

This part identifies the type of content for each package part.

2.3.2.3 Core XML Part

The Core XML part is specified in [ISO/IEC29500-1:2011] section 15.2.12.1.

This is an optional part that specifies the Core Properties of a web drawing, specified by [ISO/IEC29500-2:2011] section 11.

The following properties in the Core XML part are defined in a web drawing, specified by [ISO/IEC29500-2:2011] Table 11-1.

Property Name
category
created
creator
description
keywords
language
lastModifiedBy
lastPrinted
modified
subject
title

2.3.2.4 Custom XML Part

The Custom XML part is specified in [ISO/IEC29500-1:2011] section 15.2.12.2.

This is an optional part that specifies the Custom Properties of a web drawing, as specified by [ISO/IEC29500-1:2011] section 22.3. The syntax of the Custom Properties is specified by [ISO/IEC29500-1:2011] section 22.3.2.2.

The following properties in the Custom XML Part are defined in a web drawing.

Property Name	Data Type	Data Type Specified in
BuildNumberEdited	i4	[ISO/IEC29500-1:2011] section 22.4.2.14
IsMetric	bool	[ISO/IEC29500-1:2011] section 22.4.2.3

The lower 16 bits of the **BuildNumberEdited** property MUST be greater than 2714.

2.3.2.5 Rels XML Part

The Rels XMP part and its syntax are specified in [ISO/IEC29500-2:2011] section 9.3.

Each set of relationships sharing a common source is represented by XML stored in a Rels XML part.

2.3.3 Visio Parts

The following sections specify the Visio parts that are unique to web drawings and specified in this document.

2.3.3.1 Comments XML Part

An instance of a Comments XML part type that specifies comments in a web drawing. The following properties identify this part:

Content Type:	**application/vnd.ms-visio.comments+xml**
Root Namespace:	http://schemas.openxmlformats.org/officeDocument/2006/relationships
Source Relationship:	http://schemas.microsoft.com/visio/2010/relationships/comments

The Comments XML part MUST be a target of an explicit relationship from a Document XML part. Implicit or explicit relationships to any other parts are unused and MUST be ignored.

The root element for this part MUST be a Comments element.

2.3.3.2 Connections XML Part

An instance of a Connections XML part type that specifies the data connection information needed to query data sources and refresh the recordsets referenced by a web drawing. The following properties identify this part:

Content Type:	**application/vnd.ms-visio.connections+xml**
Root Namespace:	http://schemas.microsoft.com/office/visio/2011/1/core
Source Relationship:	http://schemas.microsoft.com/visio/2010/relationships/connections

The Connections XML part MUST be a target of an explicit relationship from a Document XML Part. Implicit or explicit relationships to any other parts are unused and MUST be ignored.

The root element for this part MUST be a DataConnections element.

2.3.3.3 Document XML Part

An instance of a Document XML part type that contains properties of a web drawing. There MUST be exactly one Document XML part in the package. The following properties identify this part:

Content Types:	**application/vnd.ms-visio.drawing.main+xml** **application/vnd.ms-visio.drawing.macroEnabled.main+xml**
Root Namespace:	http://schemas.microsoft.com/office/visio/2011/1/core
Source Relationship:	http://schemas.microsoft.com/visio/2010/relationships/document

The Document XML part MUST be a target of an explicit relationship in the package-relationship item.

The Document XML part is permitted to have explicit relationships to the following parts:

- Connections XML Part
- Masters XML Part
- Pages XML Part
- Recordsets XML Part
- Theme XML Part
- Comments XML Part
- Extensions XML Part

Implicit or explicit relationships to any other parts are unused and MUST be ignored.

The root element for this part MUST be a VisioDocument element.

2.3.3.4 Extensions XML Part

An instance of an Extensions XML part type that specifies extensibility in a web drawing. The following properties identify this part:

Content Type:	**application/vnd.ms-visio.extensions+xml**
Root Namespace:	http://schemas.microsoft.com/office/visio/2011/1/core
Source Relationship:	http://schemas.microsoft.com/visio/2010/relationships/extensions

The Extensions XML part MUST be a target of an explicit relationship from a Document XML Part. Implicit or explicit relationships to any other parts are unused and MUST be ignored.

The root element for this part MUST be an Extensions element.

2.3.3.5 Image Part

An instance of an Image part type that specifies an image resource used in rendering a web drawing. The following properties identify this part:

Content Type:	**image/bmp** **image/x-emf** **image/gif** **image/jpeg** **image/png** **image/tiff**
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/image

Each part of this type is an image file that conforms to one of the following formats:

- The bitmap (BMP) format specified in [MSDN-BMPST].
- The enhanced metafile format (EMF) format specified in [MS-EMF].
- The Graphics Interchange Format (GIF) format specified in [GIF89a].
- The Joint Photographic Experts Group (JPEG) format specified in [JFIF].
- The Portable Network Graphics (PNG) format specified in [RFC2083].
- The TIFF format specified in [RFC3302].

An Image part MUST be a target of an explicit relationship from a Page XML Part except in the case of a fallback image. An Images part MUST NOT have implicit or explicit relationships to any other part specified in this specification.

2.3.3.6 Master XML Part

An instance of a Master XML part type that specifies contents of a master in a web drawing. The following properties identify this part:

Content Type:	**application/vnd.ms-visio.master+xml**
Root Namespace:	http://schemas.microsoft.com/office/visio/2011/1/core
Source Relationship:	http://schemas.microsoft.com/visio/2010/relationships/master

The Master XML part MUST be a target of an explicit relationship from a Masters part. The Master XML part is permitted to have explicit relationships to the following parts:

- Image Part

Implicit or explicit relationships to any other parts are unused and MUST be ignored except in the case of fallback images.

The root element for this part MUST be a MasterContents element.

2.3.3.7 Masters XML Part

An instance of a Masters XML part type that specifies a collection of masters in a web drawing. The following properties identify this part:

Content Type:	**application/vnd.ms-visio.masters+xml**
Root Namespace:	http://schemas.microsoft.com/office/visio/2011/1/core
Source Relationship:	http://schemas.microsoft.com/visio/2010/relationships/masters

The Masters part MUST be a target of an explicit relationship from a Document XML Part. The Masters part is permitted to have explicit relationships to the following parts:

- Master XML Part

Implicit or explicit relationships to any other parts are unused and MUST be ignored.

The root element for this part MUST be a Masters element.

2.3.3.8 Page XML Part

An instance of a Page XML part type specifies the contents of a drawing page in a web drawing. The following properties identify this part:

Content Type:	**application/vnd.ms-visio.page+xml**
Root Namespace:	http://schemas.microsoft.com/office/visio/2011/1/core
Source Relationship:	http://schemas.microsoft.com/visio/2010/relationships/page

The Page XML part MUST be a target of an explicit relationship from a Pages XML Part. The Page XML part is permitted to have explicit relationships to the following parts:

- Image Part

Implicit or explicit relationships to any other parts are unused and MUST be ignored except in the case of fallback images.

The root element for this part MUST be a PageContents element (section 2.3.4.3.5).

2.3.3.9 Pages XML Part

An instance of a Pages XML part type that specifies a collection of drawing pages in a web drawing. The following properties identify this part:

Content Type:	**application/vnd.ms-visio.pages+xml**
Root Namespace:	http://schemas.microsoft.com/office/visio/2011/1/core
Source Relationship:	http://schemas.microsoft.com/visio/2010/relationships/pages

There MUST be at most one Pages XML part in the package.

The Pages XML part MUST be a target of an explicit relationship from a Document XML Part. The Pages XML part is permitted to have explicit relationships to the following parts:

- Page XML Part

Implicit or explicit relationships to any other parts are unused and MUST be ignored.

The root element for this part MUST be a Pages element.

2.3.3.10 Recordsets XML Part

An instance of a Recordsets XML part type specifies the recordsets and data bindings in a web drawing. The following properties identify this part:

Content Type:	**application/vnd.ms-visio.recordsets+xml**
Root Namespace:	http://schemas.microsoft.com/office/visio/2011/1/core
Source Relationship:	http://schemas.microsoft.com/visio/2010/relationships/recordsets

The Recordsets XML part MUST be a target of an explicit relationship from a Document XML Part. Implicit or explicit relationships to any other parts are unused and MUST be ignored.

The root element for this part MUST be a DataRecordSets element.

2.3.3.11 Theme XML Part

An instance of a Theme XML part type specifies a dynamic theme in a web drawing. The following properties identify this part:

Content Type:	**application/vnd.openxmlformats-officedocument.theme+xml**
Root Namespace:	http://schemas.openxmlformats.org/drawingml/2006/main
Source Relationship:	http://schemas.openxmlformats.org/officeDocument/2006/relationships/theme

The Theme XML part MUST be a target of an explicit relationship from a Document XML Part. Implicit or explicit relationships to any other parts are unused and MUST be ignored.

The root element for this part MUST be a Theme element.

2.3.4 Visio XML Schema

The Visio XML Schema sections that follow specifies the XML simple types, complex types, elements and attributes contained in the parts of a web drawing.

2.3.4.1 Simple Types

This specification does not define any simple types.

2.3.4.2 Complex Types

The following Complex Type sections specify the XML complex types contained in the parts of a web drawing.

2.3.4.2.1 AttachedToolbars_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="AttachedToolbars_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:base64Binary"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.2 AuthorEntry_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: AuthorList_Type

A complex type that specifies properties used to identify an author in a Web drawing.

Attributes:

Name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the author.

Initials: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the initials of the author.

ResolutionID: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that identifies the author within the Web drawing. It MUST be equal to or greater than one. It MUST be unique amongst all the **ID** attributes of the AuthorEntry_Type child elements of the containing AuthorList_Type element.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="AuthorEntry_Type">
  <xsd:attribute name="Name" type="xsd:string"/>
  <xsd:attribute name="Initials" type="xsd:string"/>
  <xsd:attribute name="ResolutionID" type="xsd:string"/>
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.3 AuthorList_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Comments_Type

A complex type that specifies the authors in a web drawing.

Child Elements:

AuthorEntry: An AuthorEntry_Type element that specifies properties used to identify an author in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="AuthorList_Type">
  <xsd:sequence>
    <xsd:element name="AuthorEntry" type="AuthorEntry_Type" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.4 AutoLinkComparison_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataRecordSet_Type

A complex type that is unused and MUST be ignored.

Attributes:

ColumnName: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

ContextType: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

ContextTypeLabel: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="AutoLinkComparison_Type">
  <xsd:attribute name="ColumnName" type="xsd:string" use="required"/>
  <xsd:attribute name="ContextType" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="ContextTypeLabel" type="xsd:string"/>
</xsd:complexType>
```

2.3.4.2.5 Cell_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: StyleSheet_Type, Section_Type, Row_Type, Sheet_Type, DocumentSheet_Type, PageSheet_Type, ShapeSheet_Type

A complex type that specifies a single property, which can also be used to represent an operand token.

Child Elements:

RefBy: A complex type that is unused and MUST be ignored.

Attributes:

N: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of the property. It MUST be unique amongst all of the Cell_Type elements of the containing Row_Type element, and MUST be equal to a value specified in the Cells (section 2.4.4) section of this specification.

U: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies how this property is formatted and displayed in a user interface, and how it is used in a formula expression. If present, it MUST be equal to a value from the following table.

Value	Meaning
AC	Acres
DEG	Degrees
DA	Radians
AD	Degrees-minutes-seconds
RAD	Radians
BOOL	Boolean
COLOR	RGB color value
CY	Currency
DATE	Days
ED	Days
EH	Hours
EM	Minutes
ES	Seconds
EW	Weeks
HA	Hectare
CM	Centimeters
DL	Inches
FT	Feet
F_I	Feet and inches
IN	Inches
IN_F	Inches
KM	Kilometers
M	Meters
MI	Miles
MI_F	Miles
MM	Millimeters
NM	Nautical miles
PER	Percentage
YD	Yards
DP	Inches
PNT	Coordinates of a two-dimensional point
STR	String

Value	Meaning
DE	Days
C_D	Ciceros and didots
C	Ciceros
D	Didots
DT	Points
P	Picas
P_PT	Picas and points
PT	Points

E: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the error state of the property, obtained during a formula evaluation. If present, it MUST be equal to a value from the following table.

Value	Meaning
#DIM!	An error value that specifies that a dimensional value exceeds the dimension range.
#DIV/0!	An error value that specifies division by zero.
#VALUE!	An error value that specifies that an operand token is of the wrong type.
#REF!	An error value that specifies that a reference to a cell does not exist.
#NUM!	An error value that specifies an invalid number.
#N/A	An error value that specifies that a value is not available.

F: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the formula expression of the property. It MUST be either a formula expression that satisfies the Formula ABNF and Full Grammar Definition in this specification or equal to a value in the following table.

Value	Meaning
No Formula	Specifies that no formula exists.
Inh	Specifies a formula that is inherited.

V: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the value of the property. It MUST be equal to "1.#INF" if it specifies a **floating-point number** that is larger than 1.7976e308. If the value of the **V** attribute is equal to "themed", the value of the property is specified by theme inheritance.

When the **F** attribute is present, the value of the **V** attribute MUST be used until a formula evaluation is triggered on the **F** attribute that does not result in an error value. After formula evaluation is triggered on the **F** attribute, the value of the property is specified by the most recent result of the formula evaluation that does not produce an error value.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Cell_Type" mixed="true">
  <xsd:sequence>
    <xsd:element name="RefBy" type="RefBy Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="N" type="xsd:string" use="required"/>
  <xsd:attribute name="U" type="xsd:string"/>
  <xsd:attribute name="E" type="xsd:string"/>
  <xsd:attribute name="F" type="xsd:string"/>
  <xsd:attribute name="V" type="xsd:string"/>
</xsd:complexType>
```

2.3.4.2.6 CellDef_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Extensions_Type, SectionDef_Type, RowDef_Type

A complex type that specifies the definition of a cell that is not specified in this specification.

Attributes:

N: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the cell. It MUST be unique amongst all the FunctionDef_Type, CellDef_Type, and SectionDef_Type elements in the Web drawing. It MUST NOT be equal to the name of a function token listed in the Function Token Definitions section of this specification. It MUST NOT be equal to the name of a section listed in the Sections section of this specification. It MUST NOT be equal to the name of a cell listed in the Cells section of this specification.

T: An xsd:token ([XMLSCHEMA2] section 3.3.2) attribute that specifies the operand token used to specify the **Value** of the cell. It MUST be equal to a value from the following table.

Value	Operand Token
BYTE	PtgByte
BOOL	PtgBool
WORD	PtgUnsShort
SHORT	PtgShort
LONG	PtgInt
DOUBLE	PtgNum
PERCENT	PtgNum
MULTIDIM	PtgNumMultiDim
CAL	vCalendar

F: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the default formula expression of the cell.

IX: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that specifies the zero-based identifier of a collection of cells. It MUST be unique amongst all of the CellDef_Type elements of the

containing RowDef_Type element. It MUST be greater than the **IX** attribute of any preceding CellDef_Type element of the containing Extensions_Type, SectionDef_Type or RowDef_Type element. If the containing element is a RowDef_Type element and if the RowDef_Type element's containing element is a SectionDef_Type element with **T** attribute equal to "Indexed" or **N** attribute equal to "Character", "Field", "FillGradient", "Geometry", "Layer", "LineGradient", "Paragraph", "Reviewer", "Scratch", or "Tabs", **IX** MUST exist.

S: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CellDef_Type">
  <xsd:attribute name="N" type="xsd:string" use="required"/>
  <xsd:attribute name="T" type="xsd:token" use="required"/>
  <xsd:attribute name="F" type="xsd:string"/>
  <xsd:attribute name="IX" type="xsd:unsignedByte"/>
  <xsd:attribute name="S" type="xsd:unsignedByte"/>
</xsd:complexType>
```

2.3.4.2.7 ColorEntry_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Colors_Type

A complex type that specifies a color available in a color table.

Attributes:

IX: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the **zero-based index** of the element. It MUST be less than or equal to 253. It MUST be unique amongst all of the ColorEntry_Type elements of the containing Colors_Type.

RGB: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the hexadecimal value of a color in the color table.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="ColorEntry_Type">
  <xsd:attribute name="IX" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="RGB" type="xsd:string" use="required"/>
</xsd:complexType>
```

2.3.4.2.8 Colors_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: VisioDocument_Type

A complex type that specifies the color table of a web drawing.

Child Elements:

ColorEntry: A ColorEntry_Type element that specifies the colors available in a color table.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Colors_Type">
  <xsd:sequence>
    <xsd:element name="ColorEntry" type="ColorEntry Type" minOccurs="1"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.9 CommentEntry_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: CommentList_Type

A complex type that specifies properties used to identify a comment in a web drawing.

Attributes:

AuthorID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is a value that identifies the author. It MUST be equal to or greater than one.

PageID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is a value that identifies the drawing page the comment is on. The comment MUST be contained in the drawing page specified by **PageID.**

ShapeID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is a value that identifies the shape the comment is on. If no **ShapeID** is specified, the comment refers to the drawing page.

Date: An xsd:dateTime ([XMLSCHEMA2] section 3.2.7) attribute that specifies when a comment was created.

EditDate: An xsd:dateTime ([XMLSCHEMA2] section 3.2.7) attribute that specifies when a comment was last changed. The **EditDate** MUST be greater than or equal to the value of **Date**.

Done: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that specifies the current state of the comment. It MUST be equal to zero or one.

CommentID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is a unique value that identifies the comment in a drawing page. It MUST be unique amongst all the **CommentEntry_Type** child elements of the containing CommentList_Type.

AutoCommentType: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CommentEntry Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="AuthorID" type="xsd:unsignedInt" use="required"/>
      <xsd:attribute name="PageID" type="xsd:unsignedInt" use="required"/>
      <xsd:attribute name="ShapeID" type="xsd:unsignedInt"/>
      <xsd:attribute name="Date" type="xsd:dateTime" use="required"/>
      <xsd:attribute name="EditDate" type="xsd:dateTime"/>
      <xsd:attribute name="Done" type="xsd:boolean"/>
      <xsd:attribute name="CommentID" type="xsd:unsignedInt" use="required"/>
      <xsd:attribute name="AutoCommentType" type="xsd:unsignedInt"/>
    </xsd:extension>
```

```
    </xsd:simpleContent>
  </xsd:complexType>
```

2.3.4.2.10 CommentList_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Comments_Type

A complex type that specifies the comments in a web drawing.

Child Elements:

CommentEntry: A CommentEntry_Type element that specifies properties used to identify a comment in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CommentList_Type">
  <xsd:sequence>
    <xsd:element name="CommentEntry" type="CommentEntry_Type" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.11 Comments_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Comments

A complex type that specifies properties used to identify the authors and comments in a web drawing.

Child Elements:

AuthorList: An AuthorList_Type element that specifies the authors in a web drawing.

CommentList: A CommentList_Type element that specifies the comments in a web drawing.

Attributes:

ShowCommentTags: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Comments_Type">
  <xsd:sequence>
    <xsd:element name="AuthorList" type="AuthorList_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="CommentList" type="CommentList_Type" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="ShowCommentTags" type="xsd:boolean"/>
</xsd:complexType>
```

2.3.4.2.12 Connect_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Connects_Type

A complex type that is unused and MUST be ignored.

Attributes:

FromSheet: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

FromCell: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

FromPart: An xsd:int ([XMLSCHEMA2] section 3.3.17) attribute that is unused and MUST be ignored.

ToSheet: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

ToCell: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

ToPart: An xsd:int ([XMLSCHEMA2] section 3.3.17) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Connect_Type">
  <xsd:attribute name="FromSheet" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="FromCell" type="xsd:string"/>
  <xsd:attribute name="FromPart" type="xsd:int"/>
  <xsd:attribute name="ToSheet" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="ToCell" type="xsd:string"/>
  <xsd:attribute name="ToPart" type="xsd:int"/>
</xsd:complexType>
```

2.3.4.2.13 Connects_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: PageContents_Type

A complex type that is unused and MUST be ignored.

Child Elements:

Connect: A Connect_Type element that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Connects_Type">
  <xsd:sequence>
    <xsd:element name="Connect" type="Connect_Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.14 cp_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Text_Type

A complex type that specifies the beginning of a text run, and specifies an index designating the set of character properties to use.

Attributes:

IX: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the character properties used in the text run. It MUST be the **IX** attribute of a Row_Type that has a Character Section_Type parent element.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="cp_Type">
  <xsd:attribute name="IX" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.15 CT_FmtSchemeEx

Target namespace: http://visThemeSchemaUri

Referenced by: **Ext** element as specified by the **CT_OfficeArtExtension** type specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14.

Child Elements:

schemeID: A CT_SchemeID element that specifies the index of an effect scheme dynamic theme component or a connector scheme dynamic theme component.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_FmtSchemeEx" oxsd:cname="StyleMatrixEx" oxsd:cwrap="noTemplate">
  <xsd:sequence>
    <xsd:element name="schemeID" type="CT_SchemeID" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.16 CT_FontProps

Target namespace: http://visThemeSchemaUri

Referenced by: CT_FontStyles

Specifies properties used to format a text run.

Attributes:

style: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies properties used to format a text run. The value of the structure MUST be a bitwise OR combination of one or more of the values from the table in the Style Cell_Type element.

Child Elements:

color: A **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 that specifies color properties used to format a text run.

extLst: An a:CT_OfficeArtExtensionList ([ISO/IEC29500-1:2011] section 20.1.2.2.15) type which is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_FontProps" oxsd:cname="FontProps" oxsd:cwrap="noTemplate">
  <xsd:sequence>
    <xsd:element name="color" type="a:CT_Color" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="extLst" oxsd:cname="ext" type="a:CT_OfficeArtExtensionList"
minOccurs="0" maxOccurs="1" oxsd:dataStructure="optional"/>
  </xsd:sequence>
  <xsd:attribute name="style" type="xsd:unsignedInt" use="required" oxsd:cname="style"/>
</xsd:complexType>
```

2.3.4.2.17 CT_FontStyles

Target namespace: http://visThemeSchemaUri

Referenced by: CT_FontStylesGroup

Specifies a set of properties used to format a text run.

Child Elements:

fontProps: A CT_FontProps element that specifies properties used to format a text run.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_FontStyles" oxsd:cname="FontStyles" oxsd:cwrap="noTemplate">
    <xsd:sequence>
      <xsd:element name="fontProps" oxsd:cname="fontProps" type="CT_FontProps" minOccurs="3"
maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.18 CT_FontStylesGroup

Target namespace: http://visThemeSchemaUri

Referenced by: **Ext** element as specified by the **CT_OfficeArtExtension** type specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14.

Specifies the properties used to format a text run in shapes.

Child Elements:

connectorFontStyles: A CT_FontStyles element that specifies the properties used to format a text run in a connector shape.

fontStyles: A CT_FontStyles element that specifies the properties used to format a text run in a non-connector shape.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_FontStylesGroup" oxsd:cname="FontStylesGroup"
oxsd:cwrap="noTemplate">


    <xsd:sequence>
      <xsd:element name="connectorFontStyles" oxsd:cname="connectorFontStyles"
type="CT_FontStyles" minOccurs="1" maxOccurs="1" />
```

```
      <xsd:element name="fontStyles" oxsd:cname="fontStyles" type="CT_FontStyles"
minOccurs="1" maxOccurs="1" />
    </xsd:sequence>

</xsd:complexType>
```

2.3.4.2.19 CT_LineEx

Target namespace: http://visThemeSchemaUri

Referenced by: CT_LineStyle

A complex type that specifies line properties information of an effect scheme or a connector scheme dynamic theme component in a dynamic theme.

Attributes:

rndg: An a:ST_PositiveCoordinate ([ISO/IEC29500-1:2011] section 20.1.10.42) attribute that specifies the rounding radius of the outline of a shape. The value of the structure MUST be greater than or equal to zero inches. The value of zero specifies that there is no rounding. A value greater than zero specifies that any corner between two line segments, a line segment and an elliptical arc, or two elliptical arcs within the outline is rounded with a radius equal to the value.

start: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that specifies an arrowhead at the first vertex of a one-dimensional shape.

The value of the structure MUST be specified by the table in the BeginArrow Cell_Type element, and it MUST NOT be 254.

startSize: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that specifies the size of the arrowhead at the first vertex of a shape.

The value of the structure MUST be specified by the table in the BeginArrowSize Cell_Type element.

end: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that specifies an arrowhead at the last vertex of a one-dimensional shape.

The value of the structure MUST be specified by the table in the BeginArrow Cell_Type element, and it MUST NOT be 254.

endSize: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that specifies the size of the arrowhead at the last vertex of a shape.

The value of the structure MUST be specified by the table in the BeginArrowSize Cell_Type element.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_LineEx" oxsd:cname="LineEx" oxsd:cwrap="noTemplate"
oxsd:cexport="true">
  <xsd:attribute name="rndg" type="a:ST_PositiveCoordinate" use="optional"
oxsd:cname="rounding"/>
  <xsd:attribute name="start" type="xsd:unsignedByte" use="optional"
oxsd:cname="startSymbol"/>
  <xsd:attribute name="startSize" type="xsd:unsignedByte" use="optional"
oxsd:cname="startSymbolSize"/>
  <xsd:attribute name="end" type="xsd:unsignedByte" use="optional" oxsd:cname="endSymbol"/>
  <xsd:attribute name="endSize" type="xsd:unsignedByte" use="optional"
oxsd:cname="endSymbolSize"/>
</xsd:complexType>
```

2.3.4.2.20 CT_LineStyle

Target namespace: http://visThemeSchemaUri

Referenced by: CT_SchemeLineStyles

Specifies line properties and sketch effect set information of an effect scheme dynamic theme component or a connector scheme dynamic theme component in a dynamic theme.

Child Elements:

lineEx: A CT_LineEx element that specifies the line properties information.

sketch: A CT_Sketch element that specifies sketch effect set information.

extLst: An a:CT_OfficeArtExtensionList ([ISO/IEC29500-1:2011] section 20.1.2.2.15) type which is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_LineStyle" oxsd:cname="LineStyle" oxsd:cwrap="noTemplate">
  <xsd:sequence>
    <xsd:element name="lineEx" oxsd:cname="lineEx" type="CT_LineEx" minOccurs="1"
maxOccurs="1"/>
    <xsd:element name="sketch" oxsd:cname="sketch" type="CT_Sketch" minOccurs="0"
maxOccurs="1" oxsd:dataStructure="optional"/>
    <xsd:element name="extLst" oxsd:cname="ext" type="a:CT_OfficeArtExtensionList"
minOccurs="0" maxOccurs="1" oxsd:dataStructure="optional"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.21 CT_LineStyles

Target namespace: http://visThemeSchemaUri

Referenced by: **Ext** element as specified by the **CT_OfficeArtExtension** type specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14.

Child Elements:

fmtConnectorSchemeLineStyles: A CT_SchemeLineStyles element that specifies line properties and sketch effect set information of a connector scheme dynamic theme component in a dynamic theme.

fmtSchemeLineStyles: A CT_SchemeLineStyles element that specifies line properties and sketch effect set information of an effect scheme dynamic theme component in a dynamic theme.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_LineStyles" oxsd:cname="LineStyles" oxsd:cwrap="noTemplate">
  <xsd:sequence>
    <xsd:element name="fmtConnectorSchemeLineStyles"
oxsd:cname="fmtConnectorSchemeLineStyles" type="CT_SchemeLineStyles" minOccurs="1"
maxOccurs="1"/>
    <xsd:element name="fmtSchemeLineStyles" oxsd:cname="fmtSchemeLineStyles"
type="CT_SchemeLineStyles" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.22 CT_OfficeStyleSheet

Target namespace: http://schemas.openxmlformats.org/drawingml/2006/main

Referenced by: Theme

A complex type specified in [ISO/IEC29500-1:2011] section 20.1.6.9 that specifies a dynamic theme.

The following descendant elements of a **CT_OfficeStyleSheet** element specified in [ISO/IEC29500-1:2011] section 20.1.6.9 are unused and MUST be ignored.

- **ObjectDefaults** element detailed by the **CT_ObjectStyleDefaults** type specified in [ISO/IEC29500-1:2011] section 20.1.6.7.
- **ExtraClrSchemeLst** element detailed by the **CT_ColorSchemeList** type specified in [ISO/IEC29500-1:2011] section 20.1.6.5.
- **Dk2** element detailed by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.10.
- **Lt2** element detailed by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.23.
- **Hlink** element detailed by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.19.
- **FolHlink** element detailed by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.15.
- **MajorFont** element detailed by the **CT_FontCollection** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.24.
- **BgFillStyleLst** element detailed by the **CT_BackgroundFillStyleList** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.7.
- **Camera** element detailed by the **CT_Camera** type specified in [ISO/IEC29500-1:2011] section 20.1.5.5.
- **HueMod** element detailed by the **CT_PositivePercentage** type specified in [ISO/IEC29500-1:2011] section 20.1.2.3.15.
- **CustClrLst** element detailed by the **CT_CustomColorList** type specified in [ISO/IEC29500-1:2011] section 20.1.6.3.
- **HeadEnd** element detailed by the **CT_LineEndProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.38.
- **TailEnd** element detailed by the **CT_LineEndProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.57.
- **Round** element detailed by the **CT_LineJoinRound** type specified in [ISO/IEC29500-1:2011] section 20.1.8.52.
- **PattFill** element detailed by the **CT_PatternFillProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.47.
- **NoFill** element detailed by the **CT_NoFillProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.44.
- **Miter** element detailed by the **CT_LineJoinMiterProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.43.

- **CustDash** element detailed by the **CT_DashStopList** type specified in [ISO/IEC29500-1:2011] section 20.1.8.21.
- **BlipFill** element detailed by the **CT_BlipFillProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.14.
- **GrpFill** element detailed by the **CT_GroupFillProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.35.
- **TileRect** element detailed by the **CT_RelativeRect** type specified in [ISO/IEC29500-1:2011] section 20.1.8.59.
- **EffectDag** element detailed by the **CT_EffectContainer** type specified in [ISO/IEC29500-1:2011] section 20.1.8.25.
- **Blur** element detailed by the **CT_BlurEffect** type specified in [ISO/IEC29500-1:2011] section 20.1.8.15.
- **FillOverlay** element detailed by the **CT_FillOverlayEffect** type specified in [ISO/IEC29500-1:2011] section 20.1.8.29.
- **PrstShdw** element detailed by the **CT_PresetShadowEffect** type specified in [ISO/IEC29500-1:2011] section 20.1.8.49.

The attributes of the descendant elements of a **CT_OfficeStyleSheet** element specified in [ISO/IEC29500-1:2011] section 20.1.6.9 listed in the following table are unused and MUST be ignored.

Element	Attributes
Lin as specified by the **CT_LinearShadeProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.41.	**Scaled**
Ln as specified by the **CT_LineProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.2.2.24.	**Algn**
LightRig as specified by the **CT_LightRig** type specified in [ISO/IEC29500-1:2011] section 20.1.5.9.	**Dir**
Rot as specified by the **CT_SphereCoords** type specified in [ISO/IEC29500-1:2011] section 20.1.5.11.	**Lat** and **long**
GradFill as specified by the **CT_GradientFillProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.33.	**Flip**
OuterShdw as specified by the **CT_OuterShadowEffect** type specified in [ISO/IEC29500-1:2011] section 20.1.8.45.	**Algn**, **kx**, **ky**, **sx**, and **sy**
Reflection as specified by the **CT_ReflectionEffect** type specified in [ISO/IEC29500-1:2011] section 20.1.8.50.	**Algn**, **dir**, **endA**, **fadeDir**, **kx**, **ky**, **rotWithShape**, **stPos**, **sx**, and **sy**

Child Elements:

themeElements element detailed by the **CT_BaseStyles** type specified in [ISO/IEC29500-1:2011] section §A.4.1. This element specifies the dynamic theme components of a dynamic theme.

objectDefaults element detailed by the **CT_ObjectStyleDefaults** type specified in [ISO/IEC29500-1:2011] section 20.1.6.7.

extraClrSchemeLst element detailed by the **CT_ColorSchemeList** type specified in[ISO/IEC29500-1:2011] section 20.1.6.5.

custClrLst element detailed by the **CT_CustomColorList** type specified in [ISO/IEC29500-1:2011] section 20.1.6.3.

extLst element detailed by the **CT_OfficeArtExtensionList** type specified in [ISO/IEC29500-1:2011] section 20.1.2.2.15.

Attributes:

name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of the dynamic theme.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_OfficeStyleSheet">
  <xsd:sequence>
    <xsd:element name="themeElements" type="CT_BaseStyles" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="objectDefaults" type="CT_ObjectStyleDefaults" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="extraClrSchemeLst" type="CT_ColorSchemeList" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="custClrLst" type="CT_CustomColorList" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="optional" default=""/>
</xsd:complexType>
```

2.3.4.2.23 CT_SchemeID

Target namespace: http://visThemeSchemaUri

Referenced by: CT_ThemeScheme, CT_FmtSchemeEx, and **ext** element detailed by the **CT_OfficeArtExtension** type specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14.

Specifies the index of a color scheme, font scheme, effect scheme, connector scheme, or primary scheme dynamic theme component in a dynamic theme, or the GUID of a custom dynamic theme color scheme.

Attributes:

schemeEnum: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the one-based index of a color scheme, font scheme, effect scheme, connector scheme, or primary scheme dynamic theme component. If the value of the structure is equal to 65535, the GUID of a custom dynamic theme color scheme is specified by the **schemeGUID** attribute.

schemeGUID: An a:ST_Guid ([ISO/IEC29500-1:2011] section 22.9.2.4) attribute that specifies the GUID of a custom dynamic theme color scheme. If the value of the **schemeEnum** attribute is not equal to 65535, this attribute is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_SchemeID" oxsd:cname="SchemeID" oxsd:cwrap="noTemplate">
  <xsd:attribute name="schemeEnum" type="xsd:unsignedInt" use="optional" oxsd:cname="enum"/>
  <xsd:attribute name="schemeGUID" type="a:ST_Guid" use="optional" oxsd:cname="guid"/>
```

```
</xsd:complexType>
```

2.3.4.2.24 CT_SchemeLineStyles

Target namespace: http://visThemeSchemaUri

Referenced by: CT_LineStyles

Specifies a set of line properties and sketch effect set information of an effect scheme or connector scheme dynamic theme component in a dynamic theme.

Child Elements:

lineStyle: A CT_LineStyle element that specifies line properties and sketch effect set information of an effect scheme or connector scheme dynamic theme component.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_SchemeLineStyles" oxsd:cname="SchemeLineStyles"
oxsd:cwrap="noTemplate">
  <xsd:sequence>
    <xsd:element name="lineStyle" oxsd:cname="lineStyle" type="CT_LineStyle" minOccurs="3"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.25 CT_Sketch

Target namespace: http://visThemeSchemaUri

Referenced by: CT_LineStyle

A complex type that specifies sketch effect set information of an effect scheme or connector scheme dynamic theme component in a dynamic theme.

Attributes:

lnAmp: An a:ST_PositiveFixedPercentage ([ISO/IEC29500-1:2011] section 22.9.2.10) attribute that specifies the amplitude of the path perturbations for a sketch effect set. The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies no perturbation to the path; a value of one specifies maximum perturbation.

fillAmp: An a:ST_PositiveFixedPercentage ([ISO/IEC29500-1:2011] section 22.9.2.10) attribute that specifies the amplitude of the fill perturbations for a sketch effect set. The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies no perturbation to the fill; a value of one specifies maximum perturbation.

lnWeight: An a:ST_PositiveCoordinate ([ISO/IEC29500-1:2011] section 22.1.10.42) attribute that specifies the amplitude of the path perturbations for a sketch effect set. The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies no perturbation to the path; a value of one specifies maximum perturbation.

numPts: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that specifies the number of points, distributed uniformly across each path segment of a shape, where perturbations are performed

for a sketch effect set. It MUST have a value greater than or equal to zero and less than or equal to 25, with a default value of five.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Sketch" oxsd:cname="Sketch" oxsd:cwrap="noTemplate"
oxsd:cexport="true">
  <xsd:attribute name="lnAmp" type="a:ST_PositiveFixedPercentage" use="optional"
oxsd:cname="lnAmp"/>
  <xsd:attribute name="fillAmp" type="a:ST_PositiveFixedPercentage" use="optional"
oxsd:cname="fillAmp"/>
  <xsd:attribute name="lnWeight" type="a:ST_PositiveCoordinate" use="optional"
oxsd:cname="lnWeight"/>
  <xsd:attribute name="numPts" type="xsd:unsignedByte" use="optional" oxsd:cname="numPts"/>
</xsd:complexType>
```

2.3.4.2.26 CT_ThemeScheme

Target namespace: http://visThemeSchemaUri

Referenced by: **Ext** element as detailed by the **CT_OfficeArtExtension** type specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14.

Specifies the primary scheme dynamic theme component in a dynamic theme.

Child Elements:

schemeID: A CT_SchemeID element that specifies the index of the primary scheme dynamic theme component.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_ThemeScheme" oxsd:cname="ThemeScheme" oxsd:cwrap="noTemplate">
  <xsd:sequence>
    <xsd:element name="schemeID" type="CT_SchemeID" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.27 CT_VarClrScheme

Target namespace: http://visThemeSchemaUri

Referenced by: CT_VariationClrSchemeLst

Specifies a color scheme list of a dynamic theme variant.

Attributes:

monotone: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that specifies multiformat information of a dynamic theme component in a dynamic theme.

Child Elements:

VarColor1: A **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 that specifies a color property.

VarColor2: A **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 that specifies a color property.

VarColor3: A **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 that specifies a color property.

VarColor4: A **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 that specifies a color property.

VarColor5: A **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 that specifies a color property.

VarColor6: A **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 that specifies a color property.

VarColor7: A **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 that specifies a color property.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_VarClrScheme" oxsd:cname="VariationColorScheme"
oxsd:cwrap="noTemplate">
    <xsd:sequence>
      <xsd:element name="varColor1" type="a:CT_Color" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="varColor2" type="a:CT_Color" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="varColor3" type="a:CT_Color" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="varColor4" type="a:CT_Color" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="varColor5" type="a:CT_Color" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="varColor6" type="a:CT_Color" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="varColor7" type="a:CT_Color" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="extLst" oxsd:cname="ext" type="a:CT_OfficeArtExtensionList"
minOccurs="0" maxOccurs="1" oxsd:dataStructure="optional"/>
    </xsd:sequence>
    <xsd:attribute name="monotone" type="xsd:boolean" use="optional" default="false"/>
  </xsd:complexType>
```

2.3.4.2.28 CT_VariationClrSchemeLst

Target namespace: http://visThemeSchemaUri

Referenced by: **Ext** element as detailed by the **CT_OfficeArtExtension** type specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14.

Specifies four distinct color scheme lists of four distinct dynamic theme variants in a dynamic theme.

Child Elements:

VariationClrScheme: A CT_VarClrScheme type that specifies a color scheme list of a dynamic theme variant.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_VariationClrSchemeLst" oxsd:cname="VariationColorSchemeList"
oxsd:cwrap="noTemplate">
  <xsd:sequence>
    <xsd:element name="variationClrScheme" oxsd:cname="variationClrScheme"
type="CT_VarClrScheme" minOccurs="4" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.29 CT_VariationStyle

Target namespace: http://visThemeSchemaUri

Referenced by: CT_VariationStyleScheme

Specifies a style property of a style scheme list of a dynamic theme variant.

Attributes:

fillIdx: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that indirectly specifies the value of the properties of the QuickStyleFillMatrix Cell_Type element.

lineIdx: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that indirectly specifies the value of the properties of the QuickStyleLineMatrix Cell_Type element.

effectIdx: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that indirectly specifies the value of the properties of the QuickStyleEffectsMatrix Cell_Type element.

fontIdx: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute indirectly specifies the value of the properties of the QuickStyleFontMatrix Cell_Type element.

Child Elements:

extLst: An **a:CT_OfficeArtExtensionList** ([ISO/IEC29500-1:2011] section 20.1.2.2.15) type which is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_VariationStyle" oxsd:cname="VariationStyle"
oxsd:cwrap="noTemplate">
    <xsd:sequence>
      <xsd:element name="extLst" oxsd:cname="ext" type="a:CT OfficeArtExtensionList"
minOccurs="0" maxOccurs="1" oxsd:dataStructure="optional"/>
    </xsd:sequence>
    <xsd:attribute name="fillIdx" type="xsd:unsignedInt" use="required"/>
    <xsd:attribute name="lineIdx" type="xsd:unsignedInt" use="required"/>
    <xsd:attribute name="effectIdx" type="xsd:unsignedInt" use="required"/>
    <xsd:attribute name="fontIdx" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.30 CT_VariationStyleScheme

Target namespace: http://visThemeSchemaUri

Referenced by: CT_VariationStyleSchemeLst

Specifies a style scheme list of a dynamic theme variant.

Attributes:

embellishment: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies embellishment information of a dynamic theme variant in a dynamic theme.

Child Elements:

VarStyle: A CT_VariationStyle type that specifies a style property of a style scheme list of a dynamic theme variant.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_VariationStyleScheme" oxsd:cname="VariationStyleScheme"
oxsd:cwrap="noTemplate">
    <xsd:sequence>
      <xsd:element name="varStyle" oxsd:cname="varStyle"  type="CT_VariationStyle"
minOccurs="4" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="embellishment" type="xsd:unsignedInt"/>
  </xsd:complexType>
```

2.3.4.2.31 CT_VariationStyleSchemeLst

Target namespace: http://visThemeSchemaUri

Referenced by: **Ext** element as detailed by the **CT_OfficeArtExtension** type specified in [ISO/IEC29500-1:2011] section 20.1.2.2.14.

Specifies four distinct style scheme lists of four distinct dynamic theme variants in a dynamic theme.

Child Elements:

VariationStyleScheme: A CT_VariationStyleScheme type that specifies a style scheme list of a dynamic theme variant.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_VariationStyleSchemeLst" oxsd:cname="VariationStyleSchemeList"
oxsd:cwrap="noTemplate">
  <xsd:sequence>
    <xsd:element name="variationStyleScheme" oxsd:cname="variationStyleScheme"
type="CT_VariationStyleScheme" minOccurs="4" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.32 CustomMenusFile_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CustomMenusFile_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.33 CustomToolbarsFile_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CustomToolbarsFile_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.34 Data_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: ShapeSheet_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Data_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.35 DataColumn_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataColumns_Type

A complex type that specifies a field in a recordset.

Attributes:

ColumnNameID: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the identifier of the field. It MUST be unique amongst all the **DataColumn_Type** child elements of the containing DataColumns_Type.

Name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the shape data items mapped to this field in each shape that is bound to a row in the recordset as described in data binding. It MUST be unique amongst all the **DataColumn_Type** child elements of the containing DataColumns_Type.

Label: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

OrigLabel: An xsd:string ([XMLSCHEMA2] section 3.2.1) that is unused and MUST be ignored.

LangID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

Calendar: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

DataType: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that specifies the type of the data of this field. This value MUST be specified by vDataType.

UnitType: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

Currency: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

Degree: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

DisplayWidth: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

DisplayOrder: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

Mapped: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

Hyperlink: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="DataColumn_Type">
  <xsd:attribute name="ColumnNameID" type="xsd:string" use="required"/>
  <xsd:attribute name="Name" type="xsd:string" use="required"/>
  <xsd:attribute name="Label" type="xsd:string" use="required"/>
  <xsd:attribute name="OrigLabel" type="xsd:string"/>
  <xsd:attribute name="LangID" type="xsd:unsignedInt"/>
  <xsd:attribute name="Calendar" type="xsd:unsignedShort"/>
  <xsd:attribute name="DataType" type="xsd:unsignedShort"/>
  <xsd:attribute name="UnitType" type="xsd:string"/>
  <xsd:attribute name="Currency" type="xsd:unsignedShort"/>
  <xsd:attribute name="Degree" type="xsd:unsignedInt"/>
  <xsd:attribute name="DisplayWidth" type="xsd:unsignedInt"/>
  <xsd:attribute name="DisplayOrder" type="xsd:unsignedInt"/>
  <xsd:attribute name="Mapped" type="xsd:boolean"/>
  <xsd:attribute name="Hyperlink" type="xsd:boolean"/>
</xsd:complexType>
```

2.3.4.2.36 DataColumns_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataRecordSet_Type

A complex type that specifies a collection of fields in a recordset.

Child Elements:

DataColumn: A DataColumn_Type element that specifies a field in a recordset.

Attributes:

SortColumn: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

SortAsc: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="DataColumns_Type">
  <xsd:sequence>
    <xsd:element name="DataColumn" type="DataColumn Type" minOccurs="1"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="SortColumn" type="xsd:string"/>
  <xsd:attribute name="SortAsc" type="xsd:boolean"/>
</xsd:complexType>
```

2.3.4.2.37 DataConnection_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataConnections_Type

A complex type that specifies a data connection.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of a data connection. It MUST be unique amongst all the DataConnection_Type child elements of the containing DataConnections_Type.

FileName: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the full path of an **Office data connection (ODC) file** or data source file. If the type of data source specified in the **ConnectionString** attribute corresponds to a **list (2)** or Custom **add-in**, the value of this attribute MUST be an empty string.

ConnectionString: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the **connection string** to a data source. It MUST be a valid connection string as specified in [MS-ODBCSTR]. If the type of the data source specified in the connection string is listed in the following table, the syntax of the connection string MUST match the syntax specified in the following table. Otherwise, the **DataConnection_Type** MUST be ignored.

Type of data source	Syntax
OLE DB	MUST be a valid connection string as specified in [MS-ODBCSTR]
ODBC	MUST be a valid connection string as specified in [MS-ODBCSTR]
List	MUST be in the following format: "PROVIDER=WSS;DATABASE=*list URL*;LIST={*list GUID*};*viewparam*" ▪ *list URL* is the **URL** of a list (2). ▪ *list GUID* is the GUID of the list (2). ▪ *viewparam* MUST be in the format "VIEW={*view GUID*};", if the data source is a **view** of a list (2), where *view GUID* is the GUID of the view; otherwise, *viewparam* MUST be an empty string.
Workbook	MUST be a valid connection string as specified in [MS-ODBCSTR] and MUST contain the following

	key-value pairs: "DataModule=Microsoft.Office.Visio.Server.EcsDataHandler,Microsoft.Office.Visio.Server;Data Source=*workbook URL*;Extended Properties='HDR=*hdrvalue*;';" ▪ *workbook URL* is the URL of a workbook. ▪ *hdrvalue* equals YES, if the first row of data is the **header row**;otherwise, it equals NO. Other key-value pairs are unused and MUST be ignored.
Custom add-in	MUST be a valid connection string as specified in [MS-ODBCSTR] and MUST be in the following format: "DataModule=*class name,assembly name*;Add-in key=*value pairs*;" ▪ *class name* is a **class name**. ▪ *assembly name* is an **assembly name**. ▪ *value pairs* is a semicolon-separated set of key-value pairs specific to the add-in.

Command: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the query in a data connection.

FriendlyName: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

Timeout: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

AlwaysUseConnectionFile: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="DataConnection_Type">
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="FileName" type="xsd:string" use="required"/>
  <xsd:attribute name="ConnectionString" type="xsd:string"/>
  <xsd:attribute name="Command" type="xsd:string"/>
  <xsd:attribute name="FriendlyName" type="xsd:string"/>
  <xsd:attribute name="Timeout" type="xsd:unsignedInt"/>
  <xsd:attribute name="AlwaysUseConnectionFile" type="xsd:boolean"/>
</xsd:complexType>
```

2.3.4.2.38 DataConnections_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataConnections

A complex type that specifies a collection of data connections in a web drawing.

Child Elements:

DataConnection: A DataConnection_Type element that specifies a data connection.

Attributes:

NextID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the next sequential identifier of a DataConnection_Type. It MUST be a value that is one greater than the largest value found amongst the **ID** attributes of all DataConnection_Type elements in this collection.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="DataConnections_Type">
  <xsd:sequence>
    <xsd:element name="DataConnection" type="DataConnection_Type" minOccurs="1"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="NextID" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.39 DataRecordSet_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataRecordSets_Type

A complex type that specifies a recordset and the data binding between that recordset and shapes in drawing pages.

Child Elements:

Rel: A Rel_Type element that is unused and MUST be ignored.

DataColumns: A DataColumns_Type element that specifies the fields of the recordset.

PrimaryKey: A PrimaryKey_Type element that specifies a component of the primary key of the recordset. If the **RowOrder** attribute does not exist or equals zero, there MUST be at least one occurrence of this element. If the **RowOrder** attribute exists and is equal to one, this element MUST NOT exist.

RowMap: A RowMap_Type element that specifies the data binding between a row in the recordset and a shape.

RefreshConflict: A RefreshConflict_Type element that is unused and MUST be ignored.

AutoLinkComparison: An AutoLinkComparison_Type element that is unused and MUST be ignored.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the recordset. It MUST be unique amongst all the DataRecordSet_Type child elements of the containing DataRecordSets_Type.

ConnectionID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the data connection that corresponds to the recordset. It MUST be the value of the **ID** attribute of the DataConnection_Type element associated with the data connection.

Command: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the query for the data connection that corresponds to the recordset.

Options: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies additional information about the recordset. The value MUST be zero or a bitwise OR combination of one or more values in the following table.

Value	Description
1	Specifies that the recordset does not participate in refresh operations.
2	Unused, and MUST be ignored.
4	Unused, and MUST be ignored.
8	Unused, and MUST be ignored.
16	Unused, and MUST be ignored.

TimeRefreshed: An xsd:dateTime ([XMLSCHEMA2] section 3.2.7) attribute that is unused and MUST be ignored.

NextRowID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

Name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the display name of the recordset.

RowOrder: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that specifies whether the recordset uses the row number as the primary key to bind rows of data in the recordset to shapes. It MUST be zero or one. A value of one specifies that the row number is used. If one or more PrimaryKey_Type child elements exist, this attribute MUST NOT exist or MUST be zero. If no PrimaryKey_Type child elements exist, this attribute MUST exist and MUST be one.

RefreshOverwriteAll: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that specifies the conditions where a refresh operation on the recordset will update shape data values in shapes bound to the rows of the recordset. It MUST be equal to zero or one. A value of one specifies that data in shapes bound to the rows of the recordset will be updated with new data from the refreshed rows. A value of zero specifies that only the shape data values where there is a corresponding Cell_Type element with an **N** attribute equal to "DataLinked" and a **V** attribute equal to "1" will be updated with new data from the refreshed row.

RefreshNoReconciliationUI: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

RefreshInterval: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

ReplaceLinks: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

Checksum: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="DataRecordSet_Type">
  <xsd:sequence>
    <xsd:element name="Rel" type="Rel_Type" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="DataColumns" type="DataColumns Type" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="PrimaryKey" type="PrimaryKey_Type" minOccurs="0"
maxOccurs="unbounded"/>

    <xsd:element name="RowMap" type="RowMap_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="RefreshConflict" type="RefreshConflict_Type" minOccurs="0"
maxOccurs="unbounded"/>
```

```
    <xsd:element name="AutoLinkComparison" type="AutoLinkComparison_Type" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="ConnectionID" type="xsd:unsignedInt"/>
  <xsd:attribute name="Command" type="xsd:string"/>
  <xsd:attribute name="Options" type="xsd:unsignedInt"/>
  <xsd:attribute name="TimeRefreshed" type="xsd:dateTime"/>
  <xsd:attribute name="NextRowID" type="xsd:unsignedInt"/>
  <xsd:attribute name="Name" type="xsd:string"/>
  <xsd:attribute name="RowOrder" type="xsd:boolean"/>
  <xsd:attribute name="RefreshOverwriteAll" type="xsd:boolean"/>
  <xsd:attribute name="RefreshNoReconciliationUI" type="xsd:boolean"/>
  <xsd:attribute name="RefreshInterval" type="xsd:unsignedInt"/>
  <xsd:attribute name="ReplaceLinks" type="xsd:unsignedInt"/>
  <xsd:attribute name="Checksum" type="xsd:unsignedInt"/>
</xsd:complexType>
```

2.3.4.2.40 DataRecordSets_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataRecordSets

A complex type that specifies a collection of recordsets and the data binding between those recordsets and shapes in drawing pages.

Child Elements:

DataRecordSet: A DataRecordSet_Type element that specifies a recordset and the data binding between that recordset and shapes in drawing pages.

Attributes:

NextID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

ActiveRecordsetID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

DataWindowOrder: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="DataRecordSets_Type">
  <xsd:sequence>
    <xsd:element name="DataRecordSet" type="DataRecordSet_Type" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="NextID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="ActiveRecordsetID" type="xsd:unsignedInt"/>
  <xsd:attribute name="DataWindowOrder" type="xsd:string"/>
</xsd:complexType>
```

2.3.4.2.41 DocumentSettings_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: VisioDocument_Type

A complex type that is unused and MUST be ignored.

Child Elements:

GlueSettings: A GlueSettings_Type element that is unused and MUST be ignored.

SnapSettings: A SnapSettings_Type element that is unused and MUST be ignored.

SnapExtensions: A SnapExtensions_Type element that is unused and MUST be ignored.

SnapAngles: A SnapAngles_Type element that is unused and MUST be ignored.

DynamicGridEnabled: A DynamicGridEnabled_Type element that is unused and MUST be ignored.

ProtectStyles: A ProtectStyles_Type element that is unused and MUST be ignored.

ProtectShapes: A ProtectShapes_Type element specifies whether a shape is selectable.

ProtectMasters: A ProtectMasters_Type element that is unused and MUST be ignored.

ProtectBkgnds: A ProtectBkgnds_Type element that is unused and MUST be ignored.

CustomMenusFile: A CustomMenusFile_Type element that is unused and MUST be ignored.

CustomToolbarsFile: A CustomToolbarsFile_Type element that is unused and MUST be ignored.

AttachedToolbars: An AttachedToolbars_Type element that is unused and MUST be ignored.

Attributes:

TopPage: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

DefaultTextStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

DefaultLineStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

DefaultFillStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

DefaultGuideStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="DocumentSettings_Type">
  <xsd:all>
    <xsd:element name="GlueSettings" type="GlueSettings_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="SnapSettings" type="SnapSettings_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="SnapExtensions" type="SnapExtensions_Type" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="SnapAngles" type="SnapAngles Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="DynamicGridEnabled" type="DynamicGridEnabled_Type" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="ProtectStyles" type="ProtectStyles_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ProtectShapes" type="ProtectShapes Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ProtectMasters" type="ProtectMasters_Type" minOccurs="0"
maxOccurs="1"/>

    <xsd:element name="ProtectBkgnds" type="ProtectBkgnds_Type" minOccurs="0" maxOccurs="1"/>
```

```
    <xsd:element name="CustomMenusFile" type="CustomMenusFile_Type" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="CustomToolbarsFile" type="CustomToolbarsFile_Type" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="AttachedToolbars" type="AttachedToolbars_Type" minOccurs="0"
maxOccurs="1"/>
  </xsd:all>
  <xsd:attribute name="TopPage" type="xsd:unsignedInt"/>
  <xsd:attribute name="DefaultTextStyle" type="xsd:unsignedInt"/>
  <xsd:attribute name="DefaultLineStyle" type="xsd:unsignedInt"/>
  <xsd:attribute name="DefaultFillStyle" type="xsd:unsignedInt"/>
  <xsd:attribute name="DefaultGuideStyle" type="xsd:unsignedInt"/>
</xsd:complexType>
```

2.3.4.2.42 DocumentSheet_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: VisioDocument_Type

A complex type that specifies properties of a web drawing.

Child Elements:

Cell: A Cell_Type element that specifies a single property.

Trigger: A Trigger_Type element that specifies the existence of an update trigger associated with the sheet.

Section: A Section_Type element that specifies a collection of related properties.

Attributes:

Name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-dependent name of the DocumentSheet_Type.

NameU: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of the DocumentSheet_Type.

IsCustomName: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

IsCustomNameU: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

UniqueID: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

LineStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit line formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

FillStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit fill formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

TextStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit text formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="DocumentSheet_Type">
  <xsd:complexContent>
    <xsd:extension base="Sheet Type">
      <xsd:attribute name="Name" type="xsd:string"/>
      <xsd:attribute name="NameU" type="xsd:string"/>
      <xsd:attribute name="IsCustomName" type="xsd:boolean"/>
      <xsd:attribute name="IsCustomNameU" type="xsd:boolean"/>
      <xsd:attribute name="UniqueID" type="xsd:string"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

2.3.4.2.43 DynamicGridEnabled_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="DynamicGridEnabled_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:boolean"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.44 Extensions_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Extensions

A complex type that specifies sections, rows, and cells that are not specified in this specification.

Child Elements:

CellDef: A CellDef_Type element that specifies the definition of a cell that is not specified in this specification.

FunctionDef: A FunctionDef_Type element that specifies the definition of a function token that is not specified in this specification.

SectionDef: A SectionDef_Type element that specifies the definition of a section that is not specified in this specification.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Extensions_Type">
  <xsd:sequence>
    <xsd:element name="CellDef" type="CellDef_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="FunctionDef" type="FunctionDef_Type" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="SectionDef" type="SectionDef Type" minOccurs="0"
maxOccurs="unbounded"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
```

2.3.4.2.45 FaceName_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: FaceNames_Type

A complex type that specifies a font from the font table of the web drawing.

Attributes:

NameU: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of the font as a **UTF-16 Unicode** string.

UnicodeRanges: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the Unicode range of the font.

CharSets: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the set of supported **character sets** for the font.

Panos: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the PANOSE signature for the font. The structure of the string is specified in vPanose structure. If the **Panose** attribute exists, then this attribute is unused and MUST be ignored.

Panose: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the PANOSE signature for the font. The structure of the string is specified in the vPanose structure.

Flags: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="FaceName_Type">
  <xsd:attribute name="NameU" type="xsd:string" use="required"/>
  <xsd:attribute name="UnicodeRanges" type="xsd:string"/>
  <xsd:attribute name="CharSets" type="xsd:string"/>
  <xsd:attribute name="Panos" type="xsd:string"/>
  <xsd:attribute name="Panose" type="xsd:string"/>
  <xsd:attribute name="Flags" type="xsd:unsignedInt"/>
</xsd:complexType>
```

2.3.4.2.46 FaceNames_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: VisioDocument_Type

A complex type that specifies the font table of the web drawing.

Child Elements:

FaceName: A FaceName_Type element that specifies a font from the font table of the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="FaceNames_Type">
  <xsd:sequence>
    <xsd:element name="FaceName" type="FaceName_Type" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.47 fld_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Text_Type

A complex type that specifies a text field in a text run.

Attributes:

IX: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the text field to use. It MUST be the **IX** attribute of a Row_Type that has a Field Section_Type parent element.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="fld_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="IX" type="xsd:unsignedInt" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.48 FooterCenter_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: HeaderFooter_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="FooterCenter_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.49 FooterLeft_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: HeaderFooter_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="FooterLeft_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.50 FooterMargin_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: HeaderFooter_Type

A complex type that is unused and MUST be ignored.

Attributes:

Unit: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="FooterMargin_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:double">
      <xsd:attribute name="Unit" type="xsd:string"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.51 FooterRight_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: HeaderFooter_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="FooterRight_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.52 ForeignData_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: ShapeSheet_Type

A complex type that specifies the image of a shape.

Child Elements:

Rel: A Rel_Type element that specifies a relationship to a part containing the image data.

Attributes:

ForeignType: An xsd:token ([XMLSCHEMA2] section 3.3.2) attribute that specifies the type of the image. It MUST be equal to a value from the following table.

Value	Meaning
Bitmap	The format of the image specified by the **Rel** child element is bitmap (BMP), Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), Portable Network Graphics (PNG) or TIFF.
EnhMetaFile	The format of the image specified by the **Rel** child element is enhanced metafile format (EMF).
Ink	The image specified by the **Rel** child element is unused and the containing shape is invisible.
Object	The format of the image specified by the **Rel** child element is unsupported. If the unsupported part itself has a relationship to a fallback image part and the format of the fallback image is bitmap (BMP), enhanced metafile format (EMF), Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), Portable Network Graphics (PNG) or TIFF, that fallback image will be used as the image of the shape; otherwise, the unsupported part MUST be ignored and the containing shape is invisible.

ObjectType: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

ShowAsIcon: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

ObjectWidth: An xsd:double ([XMLSCHEMA2] section 3.2.5) attribute that is unused and MUST be ignored.

ObjectHeight: An xsd:double ([XMLSCHEMA2] section 3.2.5) attribute that is unused and MUST be ignored.

MappingMode: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

ExtentX: An xsd:double ([XMLSCHEMA2] section 3.2.5) attribute that is unused and MUST be ignored.

ExtentY: An xsd:double ([XMLSCHEMA2] section 3.2.5) attribute that is unused and MUST be ignored.

CompressionType: An xsd:token ([XMLSCHEMA2] section 3.3.2) attribute that is unused and MUST be ignored.

CompressionLevel: An xsd:double ([XMLSCHEMA2] section 3.2.5) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="ForeignData Type" mixed="true">
  <xsd:sequence>
    <xsd:element name="Rel" type="Rel_Type" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="ForeignType" type="xsd:token" use="required"/>
  <xsd:attribute name="ObjectType" type="xsd:unsignedInt"/>
  <xsd:attribute name="ShowAsIcon" type="xsd:boolean"/>
  <xsd:attribute name="ObjectWidth" type="xsd:double"/>
  <xsd:attribute name="ObjectHeight" type="xsd:double"/>
  <xsd:attribute name="MappingMode" type="xsd:unsignedShort"/>
```

```
    <xsd:attribute name="ExtentX" type="xsd:double"/>
    <xsd:attribute name="ExtentY" type="xsd:double"/>
    <xsd:attribute name="CompressionType" type="xsd:token"/>
    <xsd:attribute name="CompressionLevel" type="xsd:double"/>
  </xsd:complexType>
```

2.3.4.2.53 FunctionDef_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Extensions_Type

A complex type that specifies a function that is not specified in this specification.

Attributes:

N: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of the function. It MUST be unique amongst all the FunctionDef_Type, CellDef_Type, and SectionDef_Type elements in the Web drawing. It MUST NOT be equal to the name of a function token listed in the Function Token Definitions section of this specification. It MUST NOT be equal to the name of a section listed in the Sections section of this specification. It MUST NOT be equal to the name of a cell listed in the Cells section of this specification.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="FunctionDef_Type">
  <xsd:attribute name="N" type="xsd:string" use="required"/>
</xsd:complexType>
```

2.3.4.2.54 GlueSettings_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="GlueSettings_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:int"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.55 HeaderCenter_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: HeaderFooter_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="HeaderCenter_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.56 HeaderFooter_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: VisioDocument_Type

A complex type that is unused and MUST be ignored.

Child Elements:

HeaderMargin: A HeaderMargin_Type element that is unused and MUST be ignored.

FooterMargin: A FooterMargin_Type element that is unused and MUST be ignored.

HeaderLeft: A HeaderLeft_Type element that is unused and MUST be ignored.

HeaderCenter: A HeaderCenter_Type element that is unused and MUST be ignored.

HeaderRight: A HeaderRight_Type element that is unused and MUST be ignored.

FooterLeft: A FooterLeft_Type element that is unused and MUST be ignored.

FooterCenter: A FooterCenter_Type element that is unused and MUST be ignored.

FooterRight: A FooterRight_Type element that is unused and MUST be ignored.

HeaderFooterFont: A HeaderFooterFont_Type element that is unused and MUST be ignored.

Attributes:

HeaderFooterColor: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="HeaderFooter_Type">
  <xsd:all>
    <xsd:element name="HeaderMargin" type="HeaderMargin_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FooterMargin" type="FooterMargin_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="HeaderLeft" type="HeaderLeft_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="HeaderCenter" type="HeaderCenter_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="HeaderRight" type="HeaderRight_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FooterLeft" type="FooterLeft_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FooterCenter" type="FooterCenter_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FooterRight" type="FooterRight_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="HeaderFooterFont" type="HeaderFooterFont_Type" minOccurs="0"
maxOccurs="1"/>
  </xsd:all>
  <xsd:attribute name="HeaderFooterColor" type="xsd:string"/>
</xsd:complexType>
```

2.3.4.2.57 HeaderFooterFont_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: HeaderFooter_Type

A complex type that is unused and MUST be ignored.

Attributes:

Height: An xsd:int ([XMLSCHEMA2] section 3.3.17) attribute that is unused and MUST be ignored.

Width: An xsd:int ([XMLSCHEMA2] section 3.3.17) attribute that is unused and MUST be ignored.

Escapement: An xsd:int ([XMLSCHEMA2] section 3.3.17) attribute that is unused and MUST be ignored.

Orientation: An xsd:int ([XMLSCHEMA2] section 3.3.17) attribute that is unused and MUST be ignored.

Weight: An xsd:int ([XMLSCHEMA2] section 3.3.17) attribute that is unused and MUST be ignored.

Italic: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

Underline: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

StrikeOut: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

CharSet: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

OutPrecision: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

ClipPrecision: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

Quality: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

PitchAndFamily: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

FaceName: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="HeaderFooterFont_Type">
  <xsd:attribute name="Height" type="xsd:int"/>
  <xsd:attribute name="Width" type="xsd:int"/>
  <xsd:attribute name="Escapement" type="xsd:int"/>
  <xsd:attribute name="Orientation" type="xsd:int"/>
  <xsd:attribute name="Weight" type="xsd:int"/>
  <xsd:attribute name="Italic" type="xsd:unsignedByte"/>
  <xsd:attribute name="Underline" type="xsd:unsignedByte"/>
  <xsd:attribute name="StrikeOut" type="xsd:unsignedByte"/>
  <xsd:attribute name="CharSet" type="xsd:unsignedByte"/>
  <xsd:attribute name="OutPrecision" type="xsd:unsignedByte"/>
  <xsd:attribute name="ClipPrecision" type="xsd:unsignedByte"/>
  <xsd:attribute name="Quality" type="xsd:unsignedByte"/>
  <xsd:attribute name="PitchAndFamily" type="xsd:unsignedByte"/>
  <xsd:attribute name="FaceName" type="xsd:string"/>
```

```
</xsd:complexType>
```

2.3.4.2.58 HeaderLeft_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: HeaderFooter_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="HeaderLeft_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.59 HeaderMargin_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: HeaderFooter_Type

A complex type that is unused and MUST be ignored.

Attributes:

Unit: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="HeaderMargin_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:double">
      <xsd:attribute name="Unit" type="xsd:string"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.60 HeaderRight_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: HeaderFooter_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="HeaderRight_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.61 Icon_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Master_Type, MasterShortcut_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Icon_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:base64Binary"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.62 Master_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Masters_Type

A complex type that specifies a master in a web drawing.

Child Elements:

PageSheet: A PageSheet_Type element that specifies the properties of the drawing page associated with the master.

Rel: A Rel_Type element that specifies a relationship to the corresponding Master XML Part.

Icon: An Icon_Type element that is unused and MUST be ignored.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the unique one-based index of the master. It MUST be equal to or greater than one. It MUST be unique amongst all the Master_Type and MasterShortcut_Type child elements of the containing Masters_Type.

BaseID: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies a GUID which identifies the master across web drawings.

UniqueID: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies a GUID which identifies the master within a web drawing. It MUST be different than any other **UniqueID** attribute of a Master_Type in the same web drawing.

MatchByName: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

Name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-dependent name of the master.

NameU: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of a master.

IsCustomName: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused an MUST be ignored.

IsCustomNameU: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused an MUST be ignored.

IconSize: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

PatternFlags: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that specifies whether and how a master behaves as a custom pattern. If present, it MUST be equal to a value from the following table.

Value	Description
1	Specifies a line pattern where multiple copies of the pattern are bent to follow the line.
2	Specifies a line end pattern where the line end is positioned in the direction of the line.
65	Specifies a line pattern where multiple copies of the pattern are bent to follow the line and the pattern is scaled based on the scale of the drawing page.
1026	Specifies a line end pattern where the line end is in the direction of the line and the pattern is scaled based on the scale of the drawing page.
4100	Specifies a fill pattern where a single copy of the pattern is placed at the center of the shape.
20484	Specifies a fill pattern where a single copy of the pattern is placed at the center of the shape and the pattern is scaled based on the scale of the drawing page.
Any other value	Specifies that the master does not behave as a custom pattern.

Prompt: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

Hidden: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

IconUpdate: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute is unused and that MUST be ignored.

AlignName: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

MasterType: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Master_Type">
  <xsd:all>
    <xsd:element name="PageSheet" type="PageSheet Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Rel" type="Rel_Type" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="Icon" type="Icon_Type" minOccurs="0" maxOccurs="1"/>
  </xsd:all>
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="BaseID" type="xsd:string"/>
  <xsd:attribute name="UniqueID" type="xsd:string"/>
  <xsd:attribute name="MatchByName" type="xsd:boolean"/>
  <xsd:attribute name="Name" type="xsd:string"/>
  <xsd:attribute name="NameU" type="xsd:string"/>
  <xsd:attribute name="IsCustomName" type="xsd:boolean"/>
  <xsd:attribute name="IsCustomNameU" type="xsd:boolean"/>
  <xsd:attribute name="IconSize" type="xsd:unsignedShort"/>
  <xsd:attribute name="PatternFlags" type="xsd:unsignedShort"/>
```

```
    <xsd:attribute name="Prompt" type="xsd:string"/>
    <xsd:attribute name="Hidden" type="xsd:boolean"/>
    <xsd:attribute name="IconUpdate" type="xsd:boolean"/>
    <xsd:attribute name="AlignName" type="xsd:unsignedShort"/>
    <xsd:attribute name="MasterType" type="xsd:unsignedShort"/>
  </xsd:complexType>
```

2.3.4.2.63 Masters_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Masters

A complex type that specifies a collection of masters in a web drawing.

Child Elements:

Master: A Master_Type element that specifies a master.

MasterShortcut: A MasterShortcut_Type element that specifies an unused master format.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Masters_Type">
  <xsd:sequence>
    <xsd:element name="Master" type="Master_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="MasterShortcut" type="MasterShortcut_Type" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.64 MasterShortcut_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Masters_Type

A complex type that specifies an unused master format.

Child Elements:

Icon: An Icon_Type element that is unused and MUST be ignored.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the unique one-based index of the master. It MUST be equal to or greater than one. It MUST be unique amongst all the Master_Type and MasterShortcut_Type child elements of the containing Masters_Type.

Name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

NameU: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

IsCustomName: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

IsCustomNameU: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

IconSize: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

PatternFlags: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

Prompt: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

ShortcutURL: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

ShortcutHelp: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

AlignName: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

MasterType: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that specifies that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="MasterShortcut_Type">
  <xsd:all>
    <xsd:element name="Icon" type="Icon_Type" minOccurs="0" maxOccurs="1"/>
  </xsd:all>
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="Name" type="xsd:string"/>
  <xsd:attribute name="NameU" type="xsd:string"/>
  <xsd:attribute name="IsCustomName" type="xsd:boolean"/>
  <xsd:attribute name="IsCustomNameU" type="xsd:boolean"/>
  <xsd:attribute name="IconSize" type="xsd:unsignedShort"/>
  <xsd:attribute name="PatternFlags" type="xsd:unsignedShort"/>
  <xsd:attribute name="Prompt" type="xsd:string"/>
  <xsd:attribute name="ShortcutURL" type="xsd:string"/>
  <xsd:attribute name="ShortcutHelp" type="xsd:string"/>
  <xsd:attribute name="AlignName" type="xsd:unsignedShort"/>
  <xsd:attribute name="MasterType" type="xsd:unsignedShort"/>
</xsd:complexType>
```

2.3.4.2.65 Page_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Pages_Type

A complex type that specifies a drawing page.

Child Elements:

PageSheet: A PageSheet_Type element that specifies the properties of a drawing page. There MUST be exactly one occurrence of this child element.

Rel: A Rel_Type element that specifies a relationship to the corresponding Page XML part.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the unique one-based index of the drawing page. It MUST be unique amongst all the Page Type child elements of the containing Pages Type.

Name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-dependent name of a drawing page.

NameU: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of a drawing page.

IsCustomName: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

IsCustomNameU: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

Background: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that specifies whether the current page can be used as a background. It MUST be equal to zero or one.

BackPage: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the background page of the current drawing page. It MUST be equal to the **ID** attribute of a Page_Type element that has a **Background** attribute value of **TRUE**.

ViewScale: An xsd:double ([XMLSCHEMA2] section 3.2.5) attribute that is unused and MUST be ignored.

ViewCenterX: An xsd:double ([XMLSCHEMA2] section 3.2.5) attribute that is unused and MUST be ignored.

ViewCenterY: An xsd:double ([XMLSCHEMA2] section 3.2.5) attribute that is unused and MUST be ignored.

ReviewerID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies that the PageSheet values and any associated PageContents are unused and MUST be ignored.

AssociatedPage: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Page_Type">
  <xsd:all>
    <xsd:element name="PageSheet" type="PageSheet_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Rel" type="Rel_Type" minOccurs="1" maxOccurs="1"/>
  </xsd:all>
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="Name" type="xsd:string"/>
  <xsd:attribute name="NameU" type="xsd:string"/>
  <xsd:attribute name="IsCustomName" type="xsd:boolean"/>
  <xsd:attribute name="IsCustomNameU" type="xsd:boolean"/>
  <xsd:attribute name="Background" type="xsd:boolean"/>
  <xsd:attribute name="BackPage" type="xsd:unsignedInt"/>
  <xsd:attribute name="ViewScale" type="xsd:double"/>
  <xsd:attribute name="ViewCenterX" type="xsd:double"/>
  <xsd:attribute name="ViewCenterY" type="xsd:double"/>
  <xsd:attribute name="ReviewerID" type="xsd:unsignedInt"/>
  <xsd:attribute name="AssociatedPage" type="xsd:unsignedInt"/>
</xsd:complexType>
```

2.3.4.2.66 PageContents_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: MasterContents, PageContents

A complex type that specifies the information about the shapes in a master or drawing page of a web drawing.

Child Elements:

Shapes: A Shapes_Type element that specifies a collection of shapes contained in a master or drawing page.

Connects: A Connects_Type element that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="PageContents_Type">
  <xsd:sequence>
    <xsd:element name="Shapes" type="Shapes_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Connects" type="Connects_Type" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.67 Pages_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Pages

A complex type that specifies a collection of drawing pages.

Child Elements:

Page: A Page_Type element that specifies a drawing page in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Pages_Type">
  <xsd:sequence>
    <xsd:element name="Page" type="Page_Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.68 PageSheet_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Master_Type, Page_Type

A complex type that specifies the properties of a page in a web drawing.

Child Elements:

Cell: A Cell_Type element that specifies a single property.

Trigger: A Trigger_Type element that specifies the existence of an update trigger associated with the sheet.

Section: A Section_Type element that specifies a collection of related properties.

Attributes:

UniqueID: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

LineStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit line formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

FillStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit fill formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

TextStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit text formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="PageSheet_Type">
  <xsd:complexContent>
    <xsd:extension base="Sheet_Type">
      <xsd:attribute name="UniqueID" type="xsd:string"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

2.3.4.2.69 pp_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Text_Type

A complex type that specifies the beginning of a text run and an index designating the set of paragraph properties to use.

Attributes:

IX: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the paragraph properties used in the text run. It MUST be the **IX** attribute of a Row_Type that has a Paragraph Section_Type parent element.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="pp_Type">
  <xsd:attribute name="IX" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.70 PrimaryKey_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataRecordSet_Type

A complex type that specifies a component of the primary key of a recordset.

Child Elements:

RowKeyValue: A RowKeyValue_Type element that specifies the value of this component of the primary key for an individual row of a recordset. There MUST be at least one occurrence of this child element.

Attributes:

ColumnNameID: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the name of a field that is a component of the primary key. It MUST be the value of the **ColumnNameID** attribute of a DataColumn_Type descendant element of the DataRecordSet_Type whose primary key is being specified.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="PrimaryKey_Type">
  <xsd:sequence>
    <xsd:element name="RowKeyValue" type="RowKeyValue_Type" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="ColumnNameID" type="xsd:string" use="required"/>
</xsd:complexType>
```

2.3.4.2.71 ProtectBkgnds_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="ProtectBkgnds_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:boolean"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.72 ProtectMasters_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="ProtectMasters_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:boolean"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.73 ProtectShapes_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that specifies whether a shape can be selectable. If this value is one and the value of the LockSelect Cell_Type element of the shape is one, the shape is not selectable; otherwise the shape can be selectable.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="ProtectShapes_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:boolean"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.74 ProtectStyles_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="ProtectStyles_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:boolean"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.75 PublishedPage_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: PublishSettings_Type

A complex type that specifies that a drawing page is viewable in a web drawing.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of a drawing page. It MUST be the value of the **ID** attribute of the Page_Type element that corresponds to a drawing page in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="PublishedPage_Type">
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.76 PublishSettings_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: VisioDocument_Type

A complex type that specifies the set of drawing pages that are viewable and set of recordsets that can be refreshed in a web drawing.

Child Elements:

PublishedPage: A PublishedPage_Type element that specifies that a drawing page is viewable in the web drawing.

RefreshableData: A RefreshableData_Type element that specifies that a recordset can be refreshed in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="PublishSettings_Type">
  <xsd:sequence>
    <xsd:element name="PublishedPage" type="PublishedPage_Type" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="RefreshableData" type="RefreshableData_Type" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.77 RefBy_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Cell_Type, Trigger_Type

A complex type that specifies a drawing page.

Attributes:

T: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the type of the reference. The value MUST be equal to "Page".

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the **ID** attribute of a Page_Type element in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="RefBy_Type">
  <xsd:attribute name="T" type="xsd:string" use="required"/>
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.78 RefreshableData_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: PublishSettings_Type

A complex type that specifies that a recordset is refreshable in a web drawing.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of a recordset. It MUST be the value of the **ID** attribute of the DataRecordSet_Type element that corresponds to a recordset in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="RefreshableData_Type">
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.79 RefreshConflict_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataRecordSet_Type

A complex type that is unused and MUST be ignored.

Attributes:

RowID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

ShapeID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

PageID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="RefreshConflict_Type">
  <xsd:attribute name="RowID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="ShapeID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="PageID" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.80 Rel_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: ForeignData_Type, Master_Type, Page_Type, DataRecordSet_Type

A complex type that specifies a relationship to a part as specified in [ISO/IEC29500-1:2011] section 8.3.

If the parent element is a DataRecordSet_Type element, this element is unused and MUST be ignored.

Attributes:

r:id: A **ST_RelationshipId** as specified in [ISO/IEC29500-1:2011] section 22.8.2.1) attribute that specifies a relationship to a part. The namespace of the attribute is http://schemas.openxmlformats.org/officeDocument/2006/relationships.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Rel_Type">
  <xsd:attribute ref="r:id" use="required"/>
</xsd:complexType>
```

2.3.4.2.81 Row_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Section_Type

A complex type that specifies a collection of related properties.

Child Elements:

Cell: A Cell_Type element that specifies a single property.

Trigger: A Trigger_Type element that specifies the existence of an update trigger associated with the row.

Attributes:

N: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of a collection of properties. If the **N** attribute of the containing Section_Type element is equal to "User", "Property", "Actions", "Control", "Hyperlink", or "ActionTag", or if the containing Section_Type element is specified by a SectionDef_Type with a **T** attribute equal to "Named", the contained Row_Type element MUST specify an **N** attribute. If a Row_Type element specifies an **N** attribute, then it MUST NOT specify an **IX** attribute. This attribute MUST be unique amongst all of the Row_Type elements of the containing Section_Type element.

LocalName: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-dependent name of a collection of properties. It MUST be unique amongst all of the Row_Type elements of the containing Section_Type element.

IX: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the one-based identifier of a collection of properties. If the **N** attribute of the containing Section_Type element is equal to "Character", "Field", "FillGradient", "Geometry", "Layer", "LineGradient", "Paragraph", "Reviewer", "Scratch", or "Tabs", or if the containing Section_Type element is specified by a SectionDef_Type with a **T** attribute equal to "Indexed", the contained Row_Type element MUST specify an **IX** attribute or the index of the contained Row_Type element MUST be calculated implicitly by counting the number of preceding Row_Type elements with the same **N** attribute in the containing Section_Type element. If a Row_Type element specifies an **IX** attribute, then it MUST NOT specify an **N** attribute. The **IX** attribute of a Row_Type element MUST be unique amongst all of the Row_Type elements of the containing Section_Type element. It MUST be greater than the **IX** attribute of any preceding Row_Type element of the containing Section_Type.

T: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the type of the geometric path represented by a collection of properties and used in geometry visualization. It MUST be equal to a value as specified in GeometryRowTypes. It is unused and MUST be ignored unless the Row_Type element has a Geometry Section_Type parent element.

Del: A xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that specifies whether a collection that would otherwise be inherited has been deleted. It MUST be equal to zero or one. A value of one specifies that a collection has been deleted and that the values of the properties in the collection are not inherited. A value of zero specifies that a collection of properties is valid for the shape. If the **Del** attribute is not present, the value is zero.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Row_Type">
  <xsd:sequence>
    <xsd:element name="Cell" type="Cell_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Trigger" type="Trigger_Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="N" type="xsd:string"/>
```

```
    <xsd:attribute name="LocalName" type="xsd:string"/>
    <xsd:attribute name="IX" type="xsd:unsignedInt"/>
    <xsd:attribute name="T" type="xsd:string"/>
    <xsd:attribute name="Del" type="xsd:boolean"/>
  </xsd:complexType>
```

2.3.4.2.82 RowDef_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: SectionDef_Type

A complex type that specifies the definition of a row that is not specified in this specification.

Child Elements:

CellDef: A CellDef_Type element that specifies the definition of a cell that is not specified in this specification.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="RowDef_Type">
  <xsd:sequence>
    <xsd:element name="CellDef" type="CellDef_Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.83 RowKeyValue_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: PrimaryKey_Type

A complex type that specifies the value of a component of the primary key for an individual row of a recordset.

Attributes:

RowID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of a row of a recordset. It MUST be equal to the **RowID** attribute of a RowMap_Type child element of the DataRecordSet_Type element that contains this RowKeyValue_Type element. It MUST be unique amongst all the RowKeyValue_Type child elements of the containing PrimaryKey_Type.

Value: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the value of the field specified by the containing PrimaryKey_Type for the row specified by the **RowID** attribute.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="RowKeyValue_Type">
  <xsd:attribute name="RowID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="Value" type="xsd:string" use="required"/>
</xsd:complexType>
```

2.3.4.2.84 RowMap_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DataRecordSet_Type

A complex type that specifies the data binding between a row of a recordset and a shape.

Attributes:

RowID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of a row of the recordset. It MUST be unique amongst all the RowMap_Type child elements of the containing DataRecordSet_Type.

PageID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the drawing page that contains the shape. It MUST be equal to the **ID** attribute of the Page_Type element associated with the drawing page.

ShapeID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the shape. It MUST be equal to the **ID** attribute of the ShapeSheet_Type element associated with the shape.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="RowMap_Type">
  <xsd:attribute name="RowID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="PageID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="ShapeID" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.85 Section_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: StyleSheet_Type, Sheet_Type, DocumentSheet_Type, PageSheet_Type, ShapeSheet_Type

A complex type that specifies a collection of related properties.

Child Elements:

Cell: A Cell_Type element that specifies a single property.

Trigger: A Trigger_Type element that specifies the existence of an update trigger associated with the section.

Row: A Row_Type element that specifies a collection of Cell_Type elements.

Attributes:

N: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of a collection of properties. It MUST be unique amongst all of the Section_Type elements of the containing Sheet_Type element unless it is equal to "Geometry". It MUST be equal to a value as specified in Sections.

Del: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that specifies whether a collection that would otherwise be inherited has been deleted. It MUST be equal to zero or one. A value of one specifies that a collection has been deleted and that the values of the properties in the collection are not inherited. A value of zero specifies that a collection of properties is valid for the shape. If the **Del** attribute is not present, the value is zero.

IX: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the zero-based index of the element. It MUST be unique amongst all of the Section_Type elements with the same **N**

attribute of the containing Sheet_Type. It MUST be greater than the **IX** attribute of any preceding Section_Type element with the same **N** attribute of the containing Sheet_Type.

When the **IX** attribute is not present, the index of the element is calculated implicitly by counting the number of preceding Section_Type elements with the same **N** attribute in the containing Sheet_Type.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Section_Type">
  <xsd:sequence>
    <xsd:element name="Cell" type="Cell_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Trigger" type="Trigger_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Row" type="Row_Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="N" type="xsd:string" use="required"/>
  <xsd:attribute name="Del" type="xsd:boolean"/>
  <xsd:attribute name="IX" type="xsd:unsignedInt"/>
</xsd:complexType>
```

2.3.4.2.86 SectionDef_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Extensions_Type

A complex type that specifies the definition of a section that is not specified in this specification.

Child Elements:

CellDef: A CellDef_Type element that specifies the definition of a cell that is not specified in this specification.

RowDef: A RowDef_Type element that specifies the definition of a row that is not specified in this specification.

Attributes:

N: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of a collection of properties. It MUST be unique amongst all the FunctionDef_Type, CellDef_Type, and SectionDef_Type elements in the Web drawing. It MUST NOT be equal to the name of a function token listed in the Function Token Definitions section of this specification. It MUST NOT be equal to the name of a section listed in the Sections section of this specification. It MUST NOT be equal to the name of a cell listed in the Cells section of this specification.

T: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the type of rows contained by the SectionDef_Type element. It MUST be equal to a value from following table.

Value	Description
Indexed	Specifies the Row_Type elements in the SectionDef_Type element MUST have an **IX** attribute and no **N** attribute.
Named	Specifies the Row_Type elements in the SectionDef_Type element MUST have an **N** attribute and no **IX** attribute.

S: An xsd:unsignedByte ([XMLSCHEMA2] section 3.3.24) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="SectionDef_Type">
  <xsd:sequence>
    <xsd:element name="CellDef" type="CellDef Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="RowDef" type="RowDef_Type" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="N" type="xsd:string" use="required"/>
  <xsd:attribute name="T" type="xsd:string"/>
  <xsd:attribute name="S" type="xsd:unsignedByte"/>
</xsd:complexType>
```

2.3.4.2.87 Shapes_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: ShapeSheet_Type, PageContents_Type

A complex type that specifies a collection of shapes.

Child Elements:

Shape: A ShapeSheet_Type element that specifies a shape in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Shapes_Type">
  <xsd:sequence>
    <xsd:element name="Shape" type="ShapeSheet Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.88 ShapeSheet_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Shapes_Type

A complex type that specifies a collection of properties associated with a shape.

Child Elements:

Cell: A Cell_Type element that specifies a single property.

Trigger: A Trigger_Type element that specifies the existence of an update trigger associated with the sheet.

Section: A Section_Type element that specifies a collection of related properties.

Text: A Text_Type element that specifies the text of a shape. It is unused and MUST be ignored when contained in a StyleSheet_Type, DocumentSheet_Type, or PageSheet_Type.

Data1: A Data_Type element that is unused and MUST be ignored.

Data2: A Data_Type element that is unused and MUST be ignored.

Data3: A Data_Type element that is unused and MUST be ignored.

ForeignData: A ForeignData_Type element that specifies picture data, such as a **metafile**, bitmap (BMP), or fallback image. It is unused and MUST be ignored when the **Type** attribute does not equal "Foreign".

Shapes: A Shapes_Type element that specifies a collection of subshapes of the shape.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the unique one-based index of the shape. It MUST be greater than or equal to 4. It MUST be unique amongst all the ShapeSheet_Type child elements of the containing Shapes_Type.

OriginalID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

Del: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that specifies whether a shape which is an instance of a master is deleted locally. It MUST be equal to zero or one.

MasterShape: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies a relationship between a subshape on a drawing page and a subshape of a master shape. It MUST be equal to the **ID** attribute of a ShapeSheet_Type element of a master shape. If the **Master** attribute is present, **MasterShape** is unused and MUST be ignored.

UniqueID: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

Name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-dependent name of a shape.

NameU: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of a shape.

IsCustomName: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

IsCustomNameU: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

Master: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the master of the shape. It MUST be equal to the **ID** attribute of a Master_Type element.

Type: An xsd:token ([XMLSCHEMA2] section 3.3.2) attribute that specifies the type of the shape. It MUST be equal to a value from the following table.

Value	Description
Group	This MUST be the value of the attribute if the shape specified by this element has subshapes.
Guide	Specifies that the shape is not shown.
Foreign	Specifies that the data specified in this element's **ForeignData** element is shown.
Shape	Specifies that the shape does not meet any of the previous conditions.

LineStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit line formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

FillStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit fill formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

TextStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit text formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="ShapeSheet_Type">
  <xsd:complexContent>
    <xsd:extension base="Sheet Type">
      <xsd:sequence>
        <xsd:element name="Text" type="Text_Type" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Data1" type="Data_Type" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Data2" type="Data_Type" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Data3" type="Data_Type" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="ForeignData" type="ForeignData Type" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Shapes" type="Shapes_Type" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
      <xsd:attribute name="OriginalID" type="xsd:unsignedInt"/>
      <xsd:attribute name="Del" type="xsd:boolean"/>
      <xsd:attribute name="MasterShape" type="xsd:unsignedInt"/>
      <xsd:attribute name="UniqueID" type="xsd:string"/>
      <xsd:attribute name="Name" type="xsd:string"/>
      <xsd:attribute name="NameU" type="xsd:string"/>
      <xsd:attribute name="IsCustomName" type="xsd:boolean"/>
      <xsd:attribute name="IsCustomNameU" type="xsd:boolean"/>
      <xsd:attribute name="Master" type="xsd:unsignedInt"/>
      <xsd:attribute name="Type" type="xsd:token"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

2.3.4.2.89 Sheet_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: StyleSheet_Type, DocumentSheet_Type, PageSheet_Type, ShapeSheet_Type

A complex type that specifies a collection of properties associated with a style, web drawing, drawing page, or shape.

Child Elements:

Cell: A Cell_Type element that specifies a single property.

Trigger: A Trigger_Type element that specifies the existence of an update trigger associated with the sheet.

Section: A Section_Type element that specifies a collection of related properties.

Attributes:

LineStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit line formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

FillStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit fill formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

TextStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit text formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Sheet_Type" abstract="true">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Cell" type="Cell_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Trigger" type="Trigger_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Section" type="Section_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:any minOccurs="0" maxOccurs="unbounded" namespace="##other" processContents="lax"/>
  </xsd:sequence>
  <xsd:attribute name="LineStyle" type="xsd:unsignedInt"/>
  <xsd:attribute name="FillStyle" type="xsd:unsignedInt"/>
  <xsd:attribute name="TextStyle" type="xsd:unsignedInt"/>
  <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
```

2.3.4.2.90 SnapAngle_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: SnapAngles_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="SnapAngle_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:double"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.91 SnapAngles_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

Child Elements:

SnapAngle: A SnapAngle_Type element that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="SnapAngles_Type">
  <xsd:sequence>
    <xsd:element name="SnapAngle" type="SnapAngle_Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
```

```
</xsd:complexType>
```

2.3.4.2.92 SnapExtensions_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="SnapExtensions_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:int"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.93 SnapSettings_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: DocumentSettings_Type

A complex type that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="SnapSettings_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:int"/>
  </xsd:simpleContent>
</xsd:complexType>
```

2.3.4.2.94 StyleSheet_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: StyleSheets_Type

A complex type that specifies a style sheet.

Child Elements:

Cell: A Cell_Type element that specifies a single property.

Trigger: A Trigger_Type element that specifies the existence of an update trigger associated with the sheet.

Section: A Section_Type element that specifies a collection of related properties.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the index of the style sheet. It MUST be unique amongst all the StyleSheet_Type child elements of the containing StyleSheets_Type.

Name: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-dependent name of a style sheet.

NameU: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of a style sheet.

IsCustomName: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

IsCustomNameU: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

LineStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit line formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

FillStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit fill formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

TextStyle: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the identifier of the style sheet from which to inherit text formatting. It MUST be the value of the **ID** attribute associated with a StyleSheet_Type in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="StyleSheet_Type">
  <xsd:complexContent>
    <xsd:extension base="Sheet_Type">
      <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
      <xsd:attribute name="Name" type="xsd:string"/>
      <xsd:attribute name="NameU" type="xsd:string"/>
      <xsd:attribute name="IsCustomName" type="xsd:boolean"/>
      <xsd:attribute name="IsCustomNameU" type="xsd:boolean"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

2.3.4.2.95 StyleSheets_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: VisioDocument_Type

A complex type that specifies a collection of style sheets in the web drawing.

Child Elements:

StyleSheet: A StyleSheet_Type element that specifies a style sheet.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="StyleSheets_Type">
  <xsd:sequence>
    <xsd:element name="StyleSheet" type="StyleSheet_Type" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

2.3.4.2.96 Text_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: ShapeSheet_Type

A complex type that specifies the text and text properties of a shape. The text is specified by the contents of this element and is separated into text runs by its child elements.

Child Elements:

cp: A cp_Type element that specifies the beginning of a text run and the set of character properties to use.

pp: A pp_Type element that specifies the beginning of a text run and the set of paragraph properties to use.

tp: A tp_Type element that specifies the beginning of a text run and the set of tabs properties to use.

fld: A fld_Type element that specifies a text field in a text run.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Text_Type" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="cp" type="cp_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="pp" type="pp_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="tp" type="tp_Type" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="fld" type="fld_Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:choice>
</xsd:complexType>
```

2.3.4.2.97 tp_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: Text_Type

A complex type that specifies the beginning of a text run and specifies an index designating the set of tabs properties to use.

Attributes:

IX: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that specifies the tabs properties used in the text run. It MUST be the **IX** attribute of a Row_Type that has a Tabs Section_Type parent element.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="tp_Type">
  <xsd:attribute name="IX" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

2.3.4.2.98 VisioDocument_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: VisioDocument

A complex type that specifies properties of a web drawing.

Child Elements:

DocumentSettings: A DocumentSettings_Type element that is unused and MUST be ignored.

Colors: A Colors_Type element that specifies the color table for the web drawing.

FaceNames: A FaceNames_Type element that specifies the font table for the web drawing.

StyleSheets: A StyleSheets_Type element that specifies a collection of style sheets in the web drawing.

DocumentSheet: A DocumentSheet_Type element that specifies properties of the web drawing.

EventList: An EventList_Type element that that is unused and MUST be ignored.

HeaderFooter: A HeaderFooter_Type element that is unused and MUST be ignored.

PublishSettings: A PublishSettings_Type element that specifies the set of drawing pages that are viewable and set of recordsets that can be refreshed in the web drawing. If this element is missing, all drawing pages are viewable and all recordsets can be refreshed in the web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="VisioDocument_Type">
  <xsd:sequence>
    <xsd:element name="DocumentSettings" type="DocumentSettings_Type" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="Colors" type="Colors_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="FaceNames" type="FaceNames_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="StyleSheets" type="StyleSheets_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="DocumentSheet" type="DocumentSheet_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="EventList" type="EventList_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="HeaderFooter" type="HeaderFooter_Type" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="PublishSettings" type="PublishSettings_Type" minOccurs="0"
maxOccurs="1"/>
    <xsd:any minOccurs="0" maxOccurs="unbounded" namespace="##other" processContents="lax"/>
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
```

2.3.4.2.99 EventList_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: VisioDocument_Type

A complex type that is unused and MUST be ignored.

Child Elements:

EventItem: An EventItem_Type element that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="EventList_Type">
  <xsd:sequence>
    <xsd:element name="EventItem" type="EventItem_Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
```

```
</xsd:complexType>
```

2.3.4.2.100 EventItem_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: EventList_Type

A complex type that is unused and MUST be ignored.

Attributes:

ID: An xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) attribute that is unused and MUST be ignored.

Action: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

EventCode: An xsd:unsignedShort ([XMLSCHEMA2] section 3.3.23) attribute that is unused and MUST be ignored.

Enabled: An xsd:boolean ([XMLSCHEMA2] section 3.2.2) attribute that is unused and MUST be ignored.

Target: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

TargetArgs: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that is unused and MUST be ignored.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="EventItem_Type">
  <xsd:attribute name="ID" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="Action" type="xsd:unsignedShort" use="required"/>
  <xsd:attribute name="EventCode" type="xsd:unsignedShort" use="required"/>
  <xsd:attribute name="Enabled" type="xsd:boolean"/>
  <xsd:attribute name="Target" type="xsd:string" use="required"/>
  <xsd:attribute name="TargetArgs" type="xsd:string" use="required"/>
</xsd:complexType>
```

2.3.4.2.101 Trigger_Type

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

Referenced by: StyleSheet_Type, Section_Type, Row_Type, Sheet_Type, DocumentSheet_Type, PageSheet_Type, ShapeSheet_Type

A complex type that specifies the existence of an update trigger.

Child Elements:

RefBy: A RefBy_Type element that specifies a drawing page.

Attributes:

N: An xsd:string ([XMLSCHEMA2] section 3.2.1) attribute that specifies the language-independent name of the property. It MUST be unique amongst all of the Trigger_Type elements of the parent element, and MUST be equal to a value specified in the Triggers (section 2.4.5) section of this specification.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="Trigger_Type" mixed="true">
  <xsd:sequence>
    <xsd:element name="RefBy" type="RefBy Type" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="N" type="xsd:string" use="required"/>
</xsd:complexType>
```

2.3.4.3 Elements

This section specifies the XML root elements contained in the parts of a web drawing.

2.3.4.3.1 VisioDocument

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

A VisioDocument_Type element that specifies properties of a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="VisioDocument" type="VisioDocument_Type"/>
```

2.3.4.3.2 Masters

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

A Masters_Type element that specifies the masters in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Masters" type="Masters_Type"/>
```

2.3.4.3.3 MasterContents

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

A PageContents_Type element that specifies information about the shapes in a master in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="MasterContents" type="PageContents_Type"/>
```

2.3.4.3.4 Pages

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

A Pages_Type element that specifies the drawing pages in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Pages" type="Pages_Type"/>
```

2.3.4.3.5 PageContents

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

A PageContents_Type element that specifies information about the contents of a drawing page in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="PageContents" type="PageContents_Type"/>
```

2.3.4.3.6 DataConnections

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

A DataConnections_Type element that specifies the data connection information needed to query data sources and refresh the recordsets referenced by a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="DataConnections" type="DataConnections_Type"/>
```

2.3.4.3.7 DataRecordSets

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

A DataRecordSets_Type element that specifies the recordsets and data bindings in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="DataRecordSets" type="DataRecordSets_Type"/>
```

2.3.4.3.8 Comments

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

A Comments_Type element that specifies the properties of comments in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Comments" type="Comments_Type"/>
```

2.3.4.3.9 Theme

Target namespace: http://visThemeSchemaUri

A CT_OfficeStyleSheet element specified in [ISO/IEC29500-1:2011] section 20.1.6.9 that specifies the properties of a dynamic theme in a web drawing.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="theme" type="CT_OfficeStyleSheet"/>
```

2.3.4.3.10 Extensions

Target namespace: http://schemas.microsoft.com/office/visio/2011/1/core

An Extensions_Type element that specifies the definitions of sections, rows, and cells not specified in the specification.

The following W3C XML Schema ([XMLSCHEMA1] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Extensions" type="Extensions_Type"/>
```

2.3.4.4 Attributes

This specification does not define any attributes.

2.3.5 Markup Compatibility Schema

Target namespace: http://schemas.openxmlformats.org/markupcompatibility/2006

The Markup Compatibility Schema section specifies the markup compatibility attributes and elements, specified in [ISO/IEC29500-3:2011] section 10, supported in a web drawing.

2.3.5.1 Compatibility-Rule Attributes

Compatibility-rule attributes, specified in [ISO/IEC29500-3:2011] section 10.1, are XML attributes which express rules governing markup consumers' behavior when encountering XML elements and attributes from non-understood namespaces.

The following compatibility-rule attributes are supported.

Attribute Name	Specified in
Ignorable	[ISO/IEC29500-3:2011] section 10.1.1
MustUnderstand	[ISO/IEC29500-3:2011] section 10.1.4

2.3.5.2 Alternate-Content Elements

Alternate-content elements, specified in [ISO/IEC29500-3:2011] section 10.2, are a set of alternatives of XML markup and character data, of which no more than one can be processed by a markup consumer. A markup consumer chooses from among the alternatives based upon its set of understood namespaces.

The following alternate-content elements are supported.

Element Name	Specified in
AlternateContent	[ISO/IEC29500-3:2011] section 10.2.1
Choice	[ISO/IEC29500-3:2011] section 10.2.2
Fallback	[ISO/IEC29500-3:2011] section 10.2.3

2.4 ShapeSheet Properties

The ShapeSheet properties sections that follow specify the different types of Section_Type, Row_Type, and Cell_Type elements in a sheet.

2.4.1 Sections

The following sections specify the pre-defined sections that can exist for a sheet. Each section is specified by a Section_Type child element of a ShapeSheet_Type, PageSheet_Type, StyleSheet_Type, or DocumentSheet_Type element. Other sections can be defined through sheet extensibility.

The **N** attribute of a Section_Type element MUST be equal to one of the values defined in the following sections, unless it is defined through sheet extensibility. The meaning of the attribute is specified in the corresponding section.

2.4.1.1 Actions

Actions is a collection of properties that are used for formula evaluation. The collection MUST have a ShapeSheet_Type or PageSheet_Type parent element.

2.4.1.2 ActionTag

ActionTag is a collection of properties that are used for formula evaluation only. The collection MUST have a ShapeSheet_Type or PageSheet_Type parent element.

2.4.1.3 Character

Character is a collection of related properties that specify the character properties of the text of a shape. The collection MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.1.4 Connection

Connection is a collection of properties that are used for formula evaluation only. The collection MUST have a ShapeSheet_Type parent element.

2.4.1.5 Control

Control is a collection of properties that are used for formula evaluation only. The collection MUST have a ShapeSheet_Type parent element.

2.4.1.6 Field

Field is a collection of related properties that specify the text fields of a shape. The collection MUST have a ShapeSheet_Type parent element.

2.4.1.7 FillGradient

FillGradient is a collection of properties that specify the fill color gradient of a shape. The collection MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.1.8 Geometry

Geometry is a collection of related properties that specify the geometry visualization. The collection MUST have a ShapeSheet_Type parent element.

The first Row_Type child element of this element MUST be of the type MoveTo, RelMoveTo, Ellipse, or InfiniteLine.

2.4.1.9 Hyperlink

Hyperlink is a collection of related properties that specify the shape hyperlinks. The collection MUST have a ShapeSheet_Type parent element.

2.4.1.10 Layer

Layer is a collection of properties that specify all layers defined on a drawing page. The collection MUST be the child of a PageSheet_Type element.

2.4.1.11 LineGradient

LineGradient is a collection of related properties that specify the line color gradient of a shape. The collection MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.1.12 Paragraph

Paragraph is a collection of related properties that specify the paragraph properties of the text of a shape. The collection MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.1.13 Property

Property is a collection of related properties that specify the shape data. The collection MUST have a ShapeSheet_Type parent element.

2.4.1.14 Reviewer

Reviewer is a collection of properties that are used for formula evaluation. The collection MUST have a DocumentSheet_Type parent element.

2.4.1.15 Scratch

Scratch is a collection of properties that are used for formula evaluation. The collection MUST have a DocumentSheet_Type, PageSheet_Type, or ShapeSheet_Type parent element.

2.4.1.16 Tabs

Tabs is a collection of related properties that specify the tabs properties of the text of a shape. The collection MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.1.17 User

User is a collection of properties that are used for formula evaluation. The collection MUST have a DocumentSheet_Type, PageSheet_Type, or ShapeSheet_Type parent element.

If the **N** attribute of a Row_Type child element of a User Section_Type element is in a UserRowNames, that element specifies additional properties.

2.4.2 GeometryRowTypes

The following **GeometryRowTypes** sections specify the Row_Type elements that make up geometric paths. They specify the allowable XML structures that can exist under a Geometry Section_Type element. The heading of each section specifies the **T** attribute required for that structure.

The **T** attribute of a Row_Type element of a Geometry Section_Type element MUST be equal to one of the structures defined in the following sections. The meaning of the attribute is specified in the corresponding section.

2.4.2.1 ArcTo

The **ArcTo** structure is a Row_Type element that specifies a circular arc in a geometric path.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the ending vertex of the arc.
Y	Specifies the y-coordinate of the ending vertex of the arc.
A	Specifies the distance from the arc's midpoint to the midpoint of its chord.

2.4.2.2 Ellipse

The **Ellipse** structure is a Row_Type element that specifies the geometric path of an ellipse. An ellipse is specified by its center point and two points on the ellipse.

A Geometry Section_Type element that contains this Row_Type MUST NOT contain any other Row_Types.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the center point.
Y	Specifies the y-coordinate of the center point.
A	Specifies the x-coordinate of the first point on the ellipse.
B	Specified the y-coordinate of the first point on the ellipse.
C	Specified the x-coordinate of the second point on the ellipse.
D	Specified the y-coordinate of the second point on the ellipse.

2.4.2.3 EllipticalArcTo

The **EllipticalArcTo** structure is a Row_Type element that specifies an elliptical arc in a geometric path.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the endpoint.
Y	Specifies the y-coordinate of the endpoint.
A	Specifies the x-coordinate of a point on the arc.
B	Specifies the y-coordinate of a point on the arc.
C	Specifies the angle of an arc's major axis relative to the x-axis of its parent shape.
D	Specifies the ratio of an arc's major axis to its minor axis.

2.4.2.4 InfiniteLine

The **InfiniteLine** structure is a Row_Type element that specifies the geometric path of an infinite line. The infinite line is specified by two points.

A Geometry Section_Type element that contains this Row_Type element MUST NOT contain any other Row_Type elements.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the first point.
Y	Specifies the y-coordinate of the first point.
A	Specifies the x-coordinate of the second point.
B	Specifies the y-coordinate of the second point.

2.4.2.5 LineTo

The **LineTo** structure is a Row_Type element that specifies a line segment in a geometric path.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the ending vertex of the line segment.
Y	Specifies the y-coordinate of the ending vertex of the line segment.

2.4.2.6 MoveTo

The **MoveTo** structure is a Row_Type element that specifies either the starting vertex of a geometric path or the first vertex after a break in a path.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the vertex.
Y	Specifies the y-coordinate of the vertex.

2.4.2.7 NURBSTo

The **NURBSTo** structure is a Row_Type element that specifies a Non-Uniform Rational B-Spline (NURBS) in a geometric path.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the endpoint of a NURBS.
Y	Specifies the y-coordinate of the endpoint of a NURBS.
A	Specifies the second to the last knot of the NURBS.
B	Specifies the last weight of the NURBS.
C	Specifies the first knot of the NURBS.
D	Specifies the first weight of the NURBS.
E	Specifies a formula that MUST contain a NURBS function.

2.4.2.8 PolylineTo

The **PolylineTo** structure is a Row_Type element that specifies a polyline segment in a geometric path.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the ending vertex of the last line segment.
Y	Specifies the y-coordinate of the ending vertex of the last line segment.
A	Specifies a formula that MUST contain a polyline function.

2.4.2.9 RelCubBezTo

The **RelCubBezTo** structure is a Row_Type element that specifies a cubic Bezier in a geometric path. Coordinates are specified as relative coordinates. A cubic Bezier is specified by its endpoint and two control points, one at the beginning of the curve and the other at the end of the curve.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the endpoint in relative coordinates.
Y	Specifies the y-coordinate of the endpoint in relative coordinates.
A	Specifies the x-coordinate of the control point at the beginning of the curve in relative coordinates.
B	Specifies the y-coordinate of the control point at the beginning of the curve in relative coordinates.
C	Specifies the x-coordinate of the control point at the end of the curve in relative coordinates.
D	Specifies the y-coordinate of the control point at the end of the curve in relative coordinates.

2.4.2.10 RelEllipticalArcTo

The **RelEllipticalArcTo** structure is a Row_Type element that specifies an elliptical arc in a geometric path. Coordinates are specified as relative coordinates.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the endpoint in relative coordinates.
Y	Specifies the y-coordinate of the endpoint in relative coordinates.
A	Specifies the x-coordinate of the arc's control point in relative coordinates.
B	Specifies the y-coordinate of an arc's control point in relative coordinates.
C	Specifies the angle of an arc's major axis relative to the x-axis of its parent.
D	Specifies the ratio of an arc's major axis to its minor axis.

2.4.2.11 RelLineTo

The **RelLineTo** structure is a Row_Type element that specifies a line segment in a geometric path. Coordinates are specified as relative coordinates.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the ending vertex of a line segment in relative coordinates.
Y	Specifies the y-coordinate of the ending vertex of a line segment in relative coordinates.

2.4.2.12 RelMoveTo

The **RelMoveTo** structure is a Row_Type element that specifies either the starting vertex of a geometric path or the first vertex after a break in a path. Coordinates are specified as relative coordinates.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the vertex in relative coordinates.
Y	Specifies the y-coordinate of the vertex in relative coordinates.

2.4.2.13 RelQuadBezTo

The **RelQuadBezTo** structure is a Row_Type element that specifies a quadratic Bezier in a geometric path. Coordinates are specified as relative coordinates. A quadratic Bezier is specified by its endpoint and one control point.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of the endpoint in relative coordinates.
Y	Specifies the y-coordinate of the endpoint in relative coordinates.
A	Specifies the x-coordinate of the control point in relative coordinates.
B	Specifies the y-coordinate of the control point in relative coordinates.

2.4.2.14 SplineKnot

The **SplineKnot** structure is a Row_Type element that specifies a knot of a spline. A spline knot is specified by an endpoint and one control point.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of a control point.
Y	Specifies the y-coordinate of a control point.

Cell_Type element	Meaning
A	Specifies a knot of the spline.

2.4.2.15 SplineStart

The **SplineStart** is a Row_Type element that specifies properties of a spline.

The Cell_Type child elements of this structure MUST be a value from the following table.

Cell_Type element	Meaning
X	Specifies the x-coordinate of a spline's second control point.
Y	Specifies the y-coordinate of a spline's second control point.
A	Specifies the second knot of the spline.
B	Specifies the first knot of a spline.
C	Specifies the last knot of a spline.
D	Specifies the degree of a spline. The value of the structure MUST be greater than or equal to zero, and less than or equal to 25.

2.4.3 UserRowNames

The following sections specify the pre-defined rows that can exist within a User section for a sheet. Each row is specified by a Row_Type child element of a User Section_Type element. Other rows can be defined through sheet extensibility.

If the **N** attribute of a Row_Type child element of a User Section_Type element is equal to one of the values defined in the following sections. The meaning of the Row_Type element is specified in the corresponding section.

2.4.3.1 msvShapeCategories

The value of the **msvShapeCategories** user row is a PtgString structure representing a category list. The category list is a semicolon-delimited collection of category name strings as per the following:

ABNF:

```
category-list = [category *( ";" category )]
category = *(%x20-3A/%3C-7E); same as string-value minus the ";"
```

2.4.3.2 msvThemeAccentColor

The **msvThemeAccentColor** element is a Row_Type element that specifies a fixed theme or dynamic theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **AccentColor** property of a custom fixed color scheme as specified in vThemeString or the **AccentColor** property of a custom dynamic theme color scheme as specified in vDynamicThemeString.

2.4.3.3 msvThemeDarkColor

The **msvThemeDarkColor** element is a Row_Type element that specifies a dynamic theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **DarkColor** property of a custom dynamic theme color scheme as specified in vDynamicThemeString.

2.4.3.4 msvThemeLightColor

The **msvThemeLightColor** element is a Row_Type element that specifies a dynamic theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **LightColor** property of a custom dynamic theme color scheme as specified in vDynamicThemeString.

2.4.3.5 msvThemeAccentColor6

The **msvThemeAccentColor6** element is a Row_Type element that specifies a dynamic theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **AccentColor6** property of a custom dynamic theme color scheme as specified in vDynamicThemeString.

2.4.3.6 msvThemeAccentColor2

The **msvThemeAccentColor2** element is a Row_Type element that specifies a fixed theme or dynamic theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **AccentColor2** property of a custom fixed color scheme as specified in vThemeString or the **AccentColor2** property of a custom dynamic theme color scheme as specified in vDynamicThemeString.

2.4.3.7 msvThemeAccentColor3

The **msvThemeAccentColor3** element is a Row_Type element that specifies a fixed theme or dynamic theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **AccentColor3** property of a custom fixed color scheme as specified in vThemeString or the **AccentColor3** property of a custom dynamic theme color scheme as specified in vDynamicThemeString.

2.4.3.8 msvThemeAccentColor4

The **msvThemeAccentColor4** element is a Row_Type element that specifies a fixed theme or dynamic theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **AccentColor4** property of a custom fixed color scheme as specified in vThemeString or the **AccentColor4** property of a custom dynamic theme color scheme as specified in vDynamicThemeString.

2.4.3.9 msvThemeAccentColor5

The **msvThemeAccentColor5** element is a Row_Type element that specifies a fixed theme or dynamic theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **AccentColor5** property of a custom fixed color scheme as specified in vThemeString or the **AccentColor5** property of a custom dynamic theme color scheme as specified in vDynamicThemeString.

2.4.3.10 msvThemeAsianFont

The **msvThemeAsianFont** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vFont that specifies the **AsianFont** property of a custom fixed effect scheme as specified in vTheme.

2.4.3.11 msvThemeBackgroundColor

The **msvThemeBackgroundColor** element is a Row_Type element that specifies a fixed theme or dynamic theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **BackgroundColor** property of a custom fixed color scheme specified in vThemeString or the **AccentColor5** property of a custom dynamic theme color scheme as specified in vDynamicThemeString.

2.4.3.12 msvThemeColors

The **msvThemeColors** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

This Value child element is a vScalar that specifies the index of a fixed color scheme as specified in vThemeString. If the value of the structure of the Value Cell_Type child element is equal to 254, this Value specifies a custom fixed color scheme.

2.4.3.13 msvThemeComplexFont

The **msvThemeComplexFont** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vFont that specifies the **ComplexFont** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.14 msvThemeConnectorBegin

The **msvThemeConnectorBegin** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **ConnectorBegin** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.15 msvThemeConnectorBeginSize

The **msvThemeConnectorBeginSize** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **ConnectorBeginSize** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.16 msvThemeConnectorColor

The **msvThemeConnectorColor** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **ConnectorColor** property of a custom fixed color scheme as specified in vThemeString.

2.4.3.17 msvThemeConnectorEnd

The **msvThemeConnectorEnd** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **ConnectorEnd** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.18 msvThemeConnectorEnd2

The **msvThemeConnectorEnd2** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **ConnectorEnd2** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.19 msvThemeConnectorEndSize

The **msvThemeConnectorEndSize** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **ConnectorEndSize** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.20 msvThemeConnectorPattern

The **msvThemeConnectorPattern** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **ConnectorPattern** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.21 msvThemeConnectorRounding

The **msvThemeConnectorRounding** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vLength that specifies the **ConnectorRounding** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.22 msvThemeConnectorTransparency

The **msvThemeConnectorTransparency** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a PtgNumPct structure that specifies the **ConnectorTransparency** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.23 msvThemeConnectorWeight

The **msvThemeConnectorWeight** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vLength that specifies the **ConnectorWeight** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.24 msvThemeEffects

The **msvThemeEffects** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the index of a fixed effect scheme as specified in vThemeString. If the value of the structure of the Value Cell_Type child element is equal to 254, this Value specifies a custom fixed effect scheme.

2.4.3.25 msvThemeFillColor

The **msvThemeFillColor** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **FillColor** property of a custom fixed color scheme as specified in vThemeString.

2.4.3.26 msvThemeFillColor2

The **msvThemeFillColor2** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **FillColor2** property of a custom fixed color scheme as specified in vThemeString.

2.4.3.27 msvThemeFillPattern

The **msvThemeFillPattern** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **FillPattern** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.28 msvThemeFillTransparency

The **msvThemeFillTransparency** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a PtgNumPct structure that specifies the **FillTransparency** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.29 msvThemeLatinFont

The **msvThemeLatinFont** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vFont that specifies the **LatinFont** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.30 msvThemeLineColor

The **msvThemeLineColor** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **LineColor** property of a custom fixed color scheme as specified in vThemeString.

2.4.3.31 msvThemeLinePattern

The **msvThemeLinePattern** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **LinePattern** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.32 msvThemeLineRounding

The **msvThemeLineRounding** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vLength that specifies the **LineRounding** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.33 msvThemeLineTransparency

The **msvThemeLineTransparency** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a PtgNumPct structure that specifies the **LineTransparency** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.34 msvThemeLineWeight

The **msvThemeLineWeight** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vLength that specifies the **LineWeight** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.35 msvThemeShadowColor

The **msvThemeShadowColor** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **ShadowColor** property of a custom fixed color scheme as specified in vThemeString.

2.4.3.36 msvThemeShadowDirection

The **msvThemeShadowDirection** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vAngle custom token grouping that specifies the **ShadowDirection** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.37 msvThemeShadowMagnification

The **msvThemeShadowMagnification** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a PtgNumPct structure that specifies the **ShadowMagnification** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.38 msvThemeShadowPattern

The **msvThemeShadowPattern** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **ShadowPattern** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.39 msvThemeShadowStyle

The **msvThemeShadowStyle** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vScalar that specifies the **ShadowPattern** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.40 msvThemeShadowTransparency

The **msvThemeShadowTransparency** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a PtgNumPct structure that specifies the **ShadowTransparency** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.41 msvThemeShadowXOffset

The **msvThemeShadowXOffset** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vLength that specifies the **ShadowXOffset** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.42 msvThemeShadowYOffset

The **msvThemeShadowYOffset** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vLength that specifies the **ShadowYOffset** property of a custom fixed effect scheme as specified in vThemeString.

2.4.3.43 msvThemeTextColor

The **msvThemeTextColor** element is a Row_Type element that specifies a fixed theme property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a vColor that specifies the **TextColor** property of a custom fixed color scheme as specified in vThemeString.

2.4.3.44 visUSEType

The **visUSETTYpe** element is a Row_Type element that specifies a master property. It MUST have a Value child element of the type Cell_Type.

The Value child element is a PtgByte structure that specifies the type of a master when the master is not used in shape inheritance. If the master is used in shape inheritance this user row MUST NOT exist.

The value of the structure MUST equal a value from the following table.

Value	Meaning
0	This is used during formula evaluation only.
1	This is used during formula evaluation only.
2	Specifies that this is a custom fixed effect scheme master.
3	Specifies that this is a custom fixed color scheme master.

2.4.4 Cells

The following sections specify the pre-defined cells that can exist for a sheet. Each cell is specified by a Cell_Type child element of a Section_Type, Row_Type, ShapeSheet_Type, PageSheet_Type, StyleSheet_Type, or DocumentSheet_Type element. Other cells can be defined through sheet extensibility.

The **N** attribute of a Cell_Type element MUST be equal to one of the values defined in the following sections, unless it is defined through sheet extensibility. The meaning of the attribute is specified in the corresponding section.

2.4.4.1 A

The **A** cell is a formula expression or a vScalar, vAny or vLength custom token grouping that specifies a property of a shape according to the Row_Type of its parent element. It MUST have a Row_Type parent element that has a Geometry or Scratch Section_Type parent element.

If **A** has an ArcTo, Ellipse, EllipticalArcTo, or InfiniteLine Row_Type parent element, it is a vLength that specifies geometry path information according to the Row_Type.

If **A** has a NURBSTo, RelCubBezTo, RelEllipticalArcTo, RelQuadBezTo, SplineStart, or SplineKnot Row_Type parent element, it is a vScalar that specifies geometry path information according to the Row_Type.

If **A** has a PolylineTo parent element, it is a formula expression that MUST contain a polyline function.

If **A** is a descendant element of a Scratch Section_Type element, it is a vAny that is used during formula evaluation only.

2.4.4.2 Action

The **Action** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a Row_Type parent element that has an Actions Section_Type parent element.

2.4.4.3 Active

The **Active** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Layer Section_Type parent element.

2.4.4.4 AddMarkup

The **AddMarkup** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a DocumentSheet_Type parent element.

2.4.4.5 Address

The **Address** cell is a PtgString parse token that specifies a **Uniform Resource Identifier (URI)** hyperlink of a shape. It MUST have a Row_Type parent element that has a Hyperlink Section_Type parent element.

2.4.4.6 AlignBottom

The **AlignBottom** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element.

2.4.4.7 AlignCenter

The **AlignCenter** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element.

2.4.4.8 AlignLeft

The **AlignLeft** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element.

2.4.4.9 Alignment

The **Alignment** cell is a vScalar custom token grouping that specifies the tab alignment for a tabs property. It MUST have a Row_Type parent element that has a Tabs Section_Type parent element. It MUST have an **N** attribute with the following format.

Alignment#

is an unsigned long integer, and MUST be less than or equal to 59.

The value of the structure is specified by the following table.

Value	Meaning
0	Specifies a tab stop with left alignment. The text extends to the right from the tab stop position.
1	Specifies a tab stop with center alignment. The text is centered at the tab stop position.
2	Specifies a tab stop with right alignment. The text extends to the left from the tab stop position.
3	Specifies a decimal tab stop, where aligns a decimal point at the tab stop position. Text without a decimal point extends to the left of the tab stop position.
4 or greater	Specifies a decimal tab stop.

2.4.4.10 AlignMiddle

The **AlignMiddle** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element.

2.4.4.11 AlignRight

The **AlignRight** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element.

2.4.4.12 AlignTop

The **AlignTop** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element.

2.4.4.13 Angle

The **Angle** cell is a vAngle or vScalar custom token grouping that specifies the angle of rotation of a shape. The value is defined in relation to the coordinate system of the shape's parent. It MUST have a ShapeSheet_Type parent element.

If the structure is a vScalar, the value MUST be expressed as an angleInternalUnitNumber custom internal unit number. Increasing numbers indicate counterclockwise rotation.

2.4.4.14 AsianFont

The **AsianFont** cell is a vFont structure that specifies the font used for Asian characters in a text run. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

If the value of the structure is equal to zero or if the font specified does not include Asian characters, the font specified by the sibling Font Cell_Type element is used in its place.

2.4.4.15 AutoGen

The **AutoGen** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Connection Section_Type parent element.

2.4.4.16 AvenueSizeX

The **AvenueSizeX** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.17 AvenueSizeY

The **AvenueSizeY** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.18 AvoidPageBreaks

The **AvoidPageBreaks** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.19 B

The **B** cell is a vScalar, vAny or vLength custom token grouping that specifies a property of a shape according to the Row_Type of its parent element. It MUST have a Row_Type parent element that has a Geometry Section_Type or Scratch Section_Type parent element.

If the structure has an Ellipse, EllipticalArcTo, or InfiniteLine Row_Type parent element, it is a vLength that specifies the geometry path information according to the Row_Type.

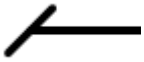
If the structure has a NURBSTo, RelCubBezTo, RelEllipticalArcTo, RelQuadBezTo, or SplineStart Row_Type parent element, it is a vScalar that specifies the geometry path information according to the Row_Type.

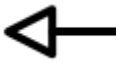
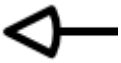
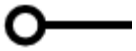
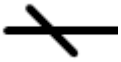
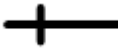
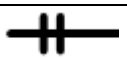
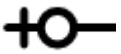
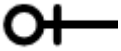
If the structure is a descendant element of a Scratch Section_Type element, it is a vAny and used during formula evaluation only.

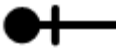
2.4.4.20 BeginArrow

The **BeginArrow** cell is a PtgByte parse token that specifies an arrowhead at the first vertex of a one-dimensional shape. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	No arrowhead.
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Value	Meaning
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	

Value	Meaning
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
254	Use the master that is specified by the formula associated with this cell for the Arrowhead.

2.4.4.21 BeginArrowSize

The **BeginArrowSize** cell is a PtgByte parse token that specifies the size of the arrowhead at the first vertex of a shape. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Very small
1	Small

Value	Meaning
2	Medium
3	Large
4	Extra Large
5	Jumbo
6	Colossal

2.4.4.22 BeginGroup

The **BeginGroup** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Actions Section_Type parent element.

2.4.4.23 BeginX

The **BeginX** cell is a vLength custom token grouping that specifies the x-coordinate of the beginning endpoint of a one-dimensional shape. The value is defined in relation to the coordinate system of the shape's parent. It MUST have a ShapeSheet_Type parent element.

If the one-dimensional shape has at least one subshape, the structure is used during formula evaluation only.

2.4.4.24 BeginY

The **BeginY** cell is a vLength custom token grouping that specifies the y-coordinate of the beginning endpoint of a one-dimensional shape. The value is defined in relation to the coordinate system of the shape's parent. It MUST have a ShapeSheet_Type parent element.

If the one-dimensional shape has at least one subshape, the structure is used during formula evaluation only.

2.4.4.25 BegTrigger

The **BegTrigger** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.26 BevelBottomHeight

The **BevelBottomHeight** cell is a vLength custom token grouping that specifies the height of the bottom bevel on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If the value of the sibling BevelBottomType Cell_Type element is zero, this value MUST be ignored.

The value of the structure MUST be greater than or equal to zero, and less than or equal to 1584 points.

2.4.4.27 BevelBottomType

The **BevelBottomType** cell is a PtgByte parse token that specifies the preset bevel type for the bottom bevel on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the table specified in BevelTopType cell.

2.4.4.28 BevelBottomWidth

The **BevelBottomWidth** cell is a vLength custom token grouping that specifies the width of the bottom bevel on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If the value of the sibling BevelBottomType Cell_Type element is zero, this value MUST be ignored.

The value of the structure MUST be greater than or equal to zero, and less than or equal to 1584 points.

2.4.4.29 BevelContourColor

The **BevelContourColor** cell is a PtgColorRGB parse token that specifies the color of the outline on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If both values of the sibling BevelTopType Cell_Type element and BevelBottomType Cell_Type element are zero, this value MUST be ignored.

2.4.4.30 BevelContourSize

The **BevelContourSize** cell is a vLength custom token grouping that specifies the thickness of the outline on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If both values of the sibling BevelTopType Cell_Type element and BevelBottomType Cell_Type element are zero, this value MUST be ignored.

The value of the structure MUST be greater than or equal to zero, and less than or equal to 1584 points.

2.4.4.31 BevelDepthColor

The **BevelDepthColor** cell is a PtgColorRGB parse token that specifies the extrusion color on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If both values of the sibling BevelTopType Cell_Type element and BevelBottomType Cell_Type element are zero, this value MUST be ignored.

2.4.4.32 BevelDepthSize

The **BevelDepthSize** cell is a vLength custom token grouping that specifies the extrusion depth on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If both values of the sibling BevelTopType Cell_Type element and BevelBottomType Cell_Type element are zero, this value MUST be ignored.

The value of the structure MUST be greater than or equal to zero, and less than or equal to 1584 points.

2.4.4.33 BevelLightingAngle

The **BevelLightingAngle** cell is a vScalar custom token grouping that specifies the direction for lighting on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If both values of the sibling BevelTopType Cell_Type element and BevelBottomType Cell_Type element are zero, this value MUST be ignored.

The unit of the value is degrees. The value MUST be greater than or equal to zero, and less than 360. See [ISO/IEC29500-1:2011] section 20.1.10.30 for details regarding the meaning of this value.

2.4.4.34 BevelLightingType

The **BevelLightingType** cell is a PtgByte parse token that specifies the preset type of lighting on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If both values of the sibling BevelTopType Cell_Type element and BevelBottomType Cell_Type element are zero, this value MUST be ignored.

The value of the structure MUST be from the following table. See [ISO/IEC29500-1:2011] section 20.1.10.30 for details regarding the meaning of the listed values.

Value	Meaning
0	None
1	Three Point
2	Balance
3	Soft
4	Harsh
5	Flood
6	Contrasting
7	Morning
8	Sunrise
9	Sunset
10	Chilly
11	Freezing
12	Flat
13	Two Point
14	Glow
15	Bright Room

2.4.4.35 BevelMaterialType

The **BevelMaterialType** cell is a PtgByte parse token that specifies the preset surface appearance on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If both values of the sibling BevelTopType Cell_Type element and BevelBottomType Cell_Type element are zero, this value MUST be ignored.

The value of the structure MUST be from the following table. See [ISO/IEC29500-1:2011] section 20.1.10.50 for details regarding the meaning of the listed values.

Value	Meaning
0	None
1	Matte
2	Warm Matte
3	Plastic
4	Metal
5	Dark Edge
6	Soft Edge
7	Flat
8	Wireframe
9	Powder
10	Translucent Powder
11	Clear

2.4.4.36 BevelTopHeight

The **BevelTopHeight** cell is a vLength custom token grouping that specifies the height of the top bevel on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If the value of the sibling BevelTopType Cell_Type element is zero, this value MUST be ignored.

The value of the structure MUST be greater than or equal to zero, and less than or equal to 1584 points.

2.4.4.37 BevelTopType

The **BevelTopType** cell is a PtgByte parse token that specifies the preset bevel type for the top bevel on a 3D shape. The bevel type specifies the appearance of a bevel. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	None
1	Circle
2	Relaxed Inset
3	Cross
4	Cool Slant
5	Angle
6	Soft Round

Value	Meaning
7	Convex
8	Slope
9	Divot
10	Riblet
11	Hard Edge
12	Art Decoration

2.4.4.38 BevelTopWidth

The **BevelTopWidth** cell is a vLength custom token grouping that specifies the width of the top bevel on a 3D shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If the value of the sibling BevelTopType Cell_Type element is zero, this value MUST be ignored.

The value of the structure MUST be greater than or equal to zero, and less than or equal to 1584 points.

2.4.4.39 BlockSizeX

The **BlockSizeX** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.40 BlockSizeY

The **BlockSizeY** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.41 Blur

The **Blur** cell is a vScalar custom token grouping that specifies the degree of blurring of an image. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.
The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies there is no blurring; a value of one specifies maximum blurring.

2.4.4.42 BottomMargin

The **BottomMargin** cell is a vLength custom token grouping that specifies the margin between the bottom border of a text block and the last line of text it contains. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

2.4.4.43 Brightness

The **Brightness** cell is a vScalar custom token grouping that specifies the brightness of an image. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100

percent. A value of 0.5 is the default brightness. A value of less than 0.5 decreases the brightness of the image; a value of greater than 0.5 increases the brightness of the image.

2.4.4.44 Bullet

The **Bullet** cell is a PtgByte parse token that specifies the bullet for a text paragraph. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element. The bullet is specified by a character in a font.

A value of zero for the structure specifies no bullet is used. If there is a bullet, any non-empty character value of the sibling BulletStr Cell_Type element is used in its place; any font value not equal to zero of the sibling BulletFont Cell_Type element is used in its place.

The value of the structure MUST be from the following table.

Value	Character	Default Font	Appearance
0	None	None	
1	0xB7	Symbol	•
2	0x6F	Courier New	◆
3	0xA7	Wingdings	▪
4	0x71	Wingdings	□
5	0x76	Wingdings	❖
6	0xD8	Wingdings	➢
7	0xFC	Wingdings	✓

2.4.4.45 BulletFont

The **BulletFont** cell is a vFont structure that specifies the font to use for the bullet of a text paragraph. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

2.4.4.46 BulletFontSize

The **BulletFontSize** cell is a vLength token grouping that specifies the font size for the bullet of a text paragraph. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

If the value of the structure is greater than zero, the bullet font size is equal to the value of the structure. If the value of the structure is equal to zero, the font size is equal to the value of the font size of the first character following the bullet. If the value of the structure is less than zero, the font size is equal to the absolute value of the structure multiplied by the value of the font size of the first character following the bullet divided by 100.

2.4.4.47 BulletStr

The **BulletStr** cell is a PtgString parse token that specifies the character or characters to use for the bullet of a text paragraph. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

2.4.4.48 ButtonFace

The **ButtonFace** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has an Actions Section_Type parent element or a Row_Type parent element that has an ActionTag Section_Type parent element.

2.4.4.49 C

The **C** cell is a vScalar, vAngle, vAny or vLength custom token grouping that specifies a property of a shape according to the Row_Type of its parent element. It MUST have a Row_Type parent element that has a Geometry Section_Type or Scratch Section_Type parent element.

If the structure has an Ellipse Row_Type parent element, it is a vLength that specifies the geometry path information according to the Row_Type.

If the structure has an EllipticalArcTo or RelEllipticalArcTo, it is a vAngle that specifies the geometry path information according to the Row_Type.

If the structure has a NURBSTo, RelCubBezTo, or SplineStart Row_Type parent element, it is a vScalar that specifies geometry path information according to the Row_Type.

If **C** is a descendant element of a Scratch Section_Type element, the structure is a vAny that is used during formula evaluation only.

2.4.4.50 Calendar

The **Calendar** cell is a vScalar custom token grouping that specifies the calendar system to use when formatting dates and times. The value of the structure MUST be specified by the vCalendar structure. It MUST have either a ShapeSheet_Type parent element or Row_Type parent element that has a Field Section_Type or Property Section_Type parent element.

If **Calendar** is a descendant of a Field Section_Type element, it specifies the calendar system used when formatting the text field specified by the element's parent Row_Type element. It is unused and MUST be ignored if it does not have a sibling Cell_Type element with an **N** attribute equal to "Type" and structure value equal to five.

If **Calendar** is a descendant of a Property Section_Type element, it specifies the calendar system to use when formatting the data specified by the element's parent Row_Type element. It is unused and MUST be ignored if it does not have a sibling Cell_Type element with an **N** attribute equal to "Type" and structure value equal to five.

If **Calendar** is a child of a ShapeSheet_Type element, it is used during formula evaluation only.

2.4.4.51 CanGlue

The **CanGlue** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Control Section_Type parent element.

2.4.4.52 Case

The **Case** cell is a PtgByte parse token that specifies the displayed case of the text of a shape. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that all characters are displayed with no changes to the case.
1	Specifies that all characters are capitalized.
2	Specifies that initial characters of each word are capitalized.

2.4.4.53 CenterX

The **CenterX** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.54 CenterY

The **CenterY** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.55 Checked

The **Checked** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has an Actions Section_Type parent element.

2.4.4.56 ClippingPath

The **ClippingPath** cell is a PtgString parse token that specifies a geometry path used to clip an image. It MUST have a ShapeSheet_Type parent element. If the value of the structure is a CellRef associated with a geometry path, the image is clipped according to the geometry path; otherwise, the image is not clipped.

The Geometry Section_Type element that specifies the associated geometry path MUST have the same ShapeSheet_Type or StyleSheet_Type parent element as the **ClippingPath** cell.

2.4.4.57 Color

The **Color** cell is a PtgColorRGB parse token that specifies either the color used for characters in a text run, the color used for stops in a fill gradient fill property, the color used for stops in a line gradient line property, or the color used for a layer. It MUST have a Row_Type parent element that has a Character Section_Type parent element, a FillGradient Section_Type parent element, a LineGradient Section_Type parent element, a Layer Section_Type parent element, or a Reviewer Section_Type parent element.

The use of the value of the structure is specified by the parent element Section_Type of the Row_Type parent element as specified in the following table.

Parent element Section_Type of the Row_Type parent element	Meaning
Character	Specifies the color used for characters in a text run.
FillGradient	Specifies the color used for a stop in a fill gradient fill property.
LineGradient	Specifies the color used for a stop in a line gradient line property.

Parent element Section_Type of the Row_Type parent element	Meaning
Layer	Specifies the color used for a layer.
Reviewer	Specifies that the value of the structure is used during formula evaluation only.

2.4.4.58 ColorSchemeIndex

The **ColorSchemeIndex** cell is a PtgByte parse token that specifies the index of the color scheme dynamic theme component. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

The value of the **V** attribute MUST be greater than or equal to 0 and less than or equal to 65535.

If the value of the **V** attribute is equal to 65535, the color scheme is specified by a custom dynamic theme color scheme specified by the Master_Type element associated with a master whose **UniqueID** attribute is equal to the argument of the USE function token of the **F** attribute of the **ColorSchemeIndex** Cell_Type element.

If the value of the structure is equal to 0, color is specified by the root style sheet.

If the value of the **V** attribute is equal to 65534, the index of the color scheme dynamic theme component of a shape is specified by the **V** attribute of the **ColorSchemeIndex** Cell_Type descendant element of the PageSheet_Type element containing the shape.

2.4.4.59 ColorTrans

The **ColorTrans** cell is a vScalar custom token grouping that specifies the color transparency used for characters in a text run, the color transparency used for stops in a fill gradient fill property, the color transparency used for stops in a line gradient line property, or the color transparency used for a layer. It MUST have a Row_Type parent element that has a Character Section_Type parent element, a FillGradient Section_Type parent element, a LineGradient Section_Type parent element, or a Layer Section_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies that the color is completely opaque; a value of one specifies that the color is completely transparent.

The use of the value of the structure is specified by the parent element Section_Type of the Row_Type parent element as specified in the following table.

Parent element Section_Type of the Row_Type parent element	Meaning
Character	Specifies the color transparency used for characters in a text run.
FillGradient	Specifies the color transparency used for a stop in a fill gradient fill property.
LineGradient	Specifies the color transparency used for a stop in a line gradient line property.
Layer	Specifies the color transparency used for a layer.

2.4.4.60 Comment

The **Comment** cell is a PtgString parse token containing plain text annotations with no specification. It is used during formula evaluation only. It MUST have either a ShapeSheet_Type parent element, or a StyleSheet_Type parent element.

2.4.4.61 ComplexScriptFont

The ComplexScriptFont cell is a vFont structure that specifies the font used for complex script characters in a text run. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

If the value of the structure is equal to zero or if the font specified does not include complex script characters, the font specified by the sibling Font Cell_Type element is used in its place.

2.4.4.62 ComplexScriptSize

The **ComplexScriptSize** cell is a vLength custom token grouping that specifies the font size used for complex script characters in a text run. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

If the value of the structure is greater than zero, the font size is equal to the value of the structure. If the value of the structure is equal to zero, the font size is equal to the value of the sibling Size Cell_Type element. If the value of the structure is less than zero, the font size is equal to the absolute value of the structure multiplied by the value of the sibling Size Cell_Type element divided by 100.

2.4.4.63 CompoundType

The **CompoundType** cell is a PtgByte parse token that specifies the compound line style for a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies a single line.
1	
2	
3	
4	

2.4.4.64 ConFixedCode

The **ConFixedCode** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.65 ConLineJumpCode

The **ConLineJumpCode** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.66 ConLineJumpDirX

The **ConLineJumpDirX** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.67 ConLineJumpDirY

The **ConLineJumpDirY** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.68 ConLineJumpStyle

The **ConLineJumpStyle** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.69 ConLineRouteExt

The **ConLineRouteExt** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.70 ConnectorSchemeIndex

The **ConnectorSchemeIndex** is a PtgByte parse token that specifies the index of the connector scheme dynamic theme component. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

The value of the **V** attribute MUST be greater than or equal to 0 and less than or equal to 65534.

If the value of the ColorSchemeIndex Cell_Type element is equal to 0, formats are specified by the root style sheet.

If the value of the **V** attribute is equal to 65534, the index of the connector scheme dynamic theme component of a shape is specified by the **V** attribute of the **ConnectorSchemeIndex** Cell_Type descendant element of the PageSheet_Type element containing the shape.

2.4.4.71 Contrast

The **Contrast** cell is a vScalar custom token grouping that specifies the contrast of an image. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of 0.5 is the default contrast. A value of less than 0.5 decreases the contrast of the image; a value of greater than 0.5 increases the contrast of the image.

2.4.4.72 Copyright

The **Copyright** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element with an **ID** attribute equal to zero.

2.4.4.73 CtrlAsInput

The **CtrlAsInput** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.74 CurrentIndex

The **CurrentIndex** cell is a PtgShort parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Reviewer Section_Type parent element.

2.4.4.75 D

The **D** cell is a vScalar or vAny or vLength custom token grouping that specifies a property of a shape according to the Row_Type of its parent element. It MUST have a Row_Type parent element that has a Geometry Section_Type or Scratch Section_Type parent element.

If **D** has an Ellipse Row_Type parent element, it is a vLength that specifies geometry path information according to the Row_Type.

If **D** has an EllipticalArcTo, RelCubBezTo, RelEllipticalArcTo, NURBSTo, or SplineStart Row_Type parent element, it is a vScalar that specifies geometry path information according to the Row_Type.

If **D** is a descendant element of a Scratch Section_Type element, it is a vAny that used during formula evaluation only.

2.4.4.76 DataLinked

The **DataLinked** cell is a PtgBool parse token that specifies that a shape data item of the data bound shape is mapped to a field of a recordset. It MUST have a Row_Type parent element that has a Property Section_Type parent element. If the shape is not a data bound shape, the token is used during formula evaluation only.

A value of one specifies that the shape data item is mapped to a field of recordset; a value of zero specifies that the shape data item is not mapped to a field of recordset.

2.4.4.77 DblUnderline

The **DblUnderline** cell is a PtgBool parse token that specifies whether a text run has a double underline character property. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

A value of one specifies that the text run has a double underline; a value of zero specifies that the text run does not have a double underline.

2.4.4.78 Default

The **Default** cell is a PtgBool parse token that specifies whether a hyperlink is the default hyperlink. It MUST have a Row_Type parent element that has a Hyperlink Section_Type parent element.

A value of one specifies that the hyperlink is the default hyperlink; a value of zero specifies that the hyperlink is not the default hyperlink.

2.4.4.79 DefaultTabStop

The **DefaultTabStop** cell is a vLength custom token grouping that specifies the default spacing between tabs in a text run. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

2.4.4.80 Denoise

The **Denoise** cell is a vScalar custom token grouping that specifies the amount of noise reduction applied to an image by modifying pixels that have randomly distributed color levels when compared to their surrounding pixels. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies no noise reduction. As the value of the structure increases, more noise reduction is applied.

2.4.4.81 Description

The **Description** cell is a PtgString parse token that specifies a description of a hyperlink. It MUST have a Row_Type parent element that has a Hyperlink Section_Type parent element.

2.4.4.82 DirX

The **DirX** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Connection Section_Type parent element.

2.4.4.83 DirY

The **DirY** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Connection Section_Type parent element.

2.4.4.84 Disabled

The **Disabled** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have either a Row_Type parent element that has an Actions Section_Type parent element or a Row_Type parent element that has an ActionTag Section_Type parent element.

2.4.4.85 DisplayLevel

The **DisplayLevel** cell is a PtgInt parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.86 DisplayMode

The **DisplayMode** cell is a PtgByte parse token that specifies the way shape information is displayed. It MUST have either a ShapeSheet_Type parent element, a StyleSheet_Type parent element, or a Row_Type parent element that has an ActionTag Section_Type parent element.

If **DisplayMode** has a Row_Type parent element, the token is used during formula evaluation only.

If **DisplayMode** has a ShapeSheet_Type parent element or a StyleSheet_Type parent element, the value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that the shape and its text is hidden.
1	Specifies that the shape is displayed behind its member shapes.
2	Specifies that the shape is displayed in front of its member shapes.

2.4.4.87 DistanceFromGround

The **DistanceFromGround** cell is a vLength custom token grouping that specifies the distance that a shape with 3D rotation properties (section 2.2.7.3.7) is raised from ground. The ground is the plane where the z-coordinate value is zero in the coordinate system of the shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to (-4000.0 / 72.0) inches, and less than or equal to (4000.0 / 72.0) inches. If the shape does not have 2.2.7.3D rotation properties, the token is used during formula evaluation only.

2.4.4.88 DocLangID

The **DocLangID** cell is a vLanguageString structure that specifies the locale settings of a web drawing as specified in [RFC4646] and [RFC4647]. It MUST have a DocumentSheet_Type parent element, and MUST be equal to the value of the Language property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML Part of a web drawing.

2.4.4.89 DocLockDuplicatePage

The **DocLockDuplicatePage** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a DocumentSheet_Type parent element.

2.4.4.90 DocLockReplace

The **DocLockReplaccell** is a PtgBool parse token that is used during formula evaluation only. It MUST have a DocumentSheet_Type parent element.

2.4.4.91 DontMoveChildren

The **DontMoveChildren** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.92 DoubleStrikethrough

The **DoubleStrikethrough** cell is a PtgBool parse token that specifies whether the text run has a double strikethrough character property. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

A value of one specifies that the text run has a double strikethrough; a value of zero specifies that the text run does not have a double strikethrough.

2.4.4.93 DrawingResizeType

The **DrawingResizeType** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.94 DrawingScale

The **DrawingScale** cell is a vLength custom token grouping that specifies a unit of distance used to define the scale of the web drawing. It MUST have a PageSheet_Type parent element.

The default value of the structure is one inch.

2.4.4.95 DrawingScaleType

The **DrawingScaleType** cell is a PtgByte parse token that specifies the type of scale of the web drawing. It MUST have a PageSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	No Scale
1	Architectural Scale
2	Civil Engineering Scale
3	Custom Scale
4	Metric Scale
5	Mechanical Engineering Scale

The default value is zero.

2.4.4.96 DrawingSizeType

The **DrawingSizeType** cell is a PtgByte parse token that specifies the type of size of the web drawing. It MUST have a PageSheet_Type parent element.

Value	Meaning
0	Same as Print Setup
1	Tightly enclose objects on Page
2	Standard Physical Page Size
3	Custom Physical Page Size
4	Logical Page Size
5	Metric Page Size
6	ANSI Engineering Page Size
7	ANSI Architectural Page Size

The default value is zero.

2.4.4.97 DropOnPageScale

The **DropOnPageScale** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.98 DynamicsOff

The **DynamicsOff** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.99 DynFeedback

The **DynFeedback** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.100 E

The **E** cell is a formula expression that specifies the NURBS formula of a geometric path of a non-uniform rational B-spline(NURBS). It MUST have a NURBSTo Row_Type parent element that has a Geometry Section_Type parent element.

2.4.4.101 EffectSchemeIndex

The **EffectSchemeIndex** cell is a PtgByte token that specifies the index of the effect scheme dynamic theme component. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

The value of the **V** attribute MUST be greater than or equal to 0 and less than or equal to 65534.

If the value of the ColorSchemeIndex Cell_Type element is equal to 0, formats are specified by the root style sheet.

If the value of the **V** attribute is equal to 65534, the index of the effect scheme dynamic theme component of a shape is specified by the **V** attribute of the **EffectSchemeIndex** Cell_Type descendant element of the PageSheet_Type element containing the shape.

2.4.4.102 EmbellishmentIndex

The **EmbellishmentIndex** cell is a PtgInt parse token that specifies embellishment information. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

The value of the **V** attribute MUST be greater than or equal to 0 and less than or equal to 65534.

If the value of the **V** attribute is equal to 65534, the embellishment information of a shape is specified by the **V** attribute of the **EmbellishmentIndex** Cell_Type descendant element of the PageSheet_Type element containing the shape.

If the value of the **V** attribute is equal to 0, the embellishment information of a shape is specified by the dynamic theme variant of a dynamic theme.

2.4.4.103 EnableFillProps

The **EnableFillProps** cell is a PtgBool parse token that specifies whether the fill properties of a style are included in inheritance. It MUST have a StyleSheet_Type parent element.

A value of one specifies that the fill properties are included in style inheritance; a value of zero specifies that the fill properties are not included in style inheritance.

2.4.4.104 EnableGrid

The **EnableGrid** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.105 EnableLineProps

The **EnableLineProps** cell is a PtgBool parse token that specifies whether the line properties of a style are included in inheritance. It MUST have a StyleSheet_Type parent element.

A value of one specifies that the line properties are included in style inheritance; a value of zero specifies that the line properties are not included in style inheritance.

2.4.4.106 EnableTextProps

The **EnableTextProps** cell is a PtgBool parse token that specifies the character properties, paragraph properties, tabs properties and text block format of a style are included in inheritance. It MUST have a StyleSheet_Type parent element.

A value of one specifies that the character properties, paragraph properties, tabs properties and text block format are included in style inheritance; a value of zero specifies that the character properties, paragraph properties, tabs properties and text block format are not included in style inheritance.

2.4.4.107 EndArrow

The **EndArrow** cell is a PtgByte parse token that specifies an arrowhead at the first vertex of a one-dimensional shape. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be from the table specified in the BeginArrow cell.

2.4.4.108 EndArrowSize

The **EndArrowSize** cell is a PtgByte parse token that specifies the size of the arrowhead at the last vertex of a shape. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Very small
1	Small
2	Medium
3	Large
4	Extra Large
5	Jumbo
6	Colossal

2.4.4.109 EndTrigger

The **EndTrigger** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.110 EndX

The **EndX** cell is a vLength custom token grouping that specifies the x-coordinate of the ending endpoint of a one-dimensional shape. The value is defined in relation to the coordinate system of the shape's parent. It MUST have a ShapeSheet_Type parent element.

If the one-dimensional shape has at least one subshape, the custom token grouping is used during formula evaluation only.

2.4.4.111 EndY

The **EndY** cell is a vLength custom token grouping that specifies the y-coordinate of the ending endpoint of a one-dimensional shape. The value is defined in relation to the coordinate system of the shape's parent. It MUST have a ShapeSheet_Type parent element.

If the one-dimensional shape has at least one subshape, the custom token grouping is used during formula evaluation only.

2.4.4.112 EventDblClick

The **EventDblClick** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.113 EventDrop

The **EventDrop** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.114 EventMultiDrop

The **EventMultiDrop** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.115 EventXFMod

The **EventXFMod** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.116 ExtraInfo

The ExtraInfo cell is a PtgString parse token that specifies a URI query string or zoom level string of the drawing page referenced by the hyperlink. It MUST have a Row_Type parent element that has a Hyperlink Section_Type parent element.

The zoom level percentage is normalized such that 1 corresponds to 100 percent. If the hyperlink refers to a drawing page, the value of the structure is a zoom level string that is defined by the following:

ABNF

```
zoom-level-string = ["zoom=" zoom-level]
zoom-level = 1*digit
```

2.4.4.117 FillBkgnd

The **FillBkgnd** cell is a vColor custom input type that specifies the color of the background fill property of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.118 FillBkgndTrans

The **FillBkgndTrans** cell is a vScalar custom token grouping that specifies the transparency level of the background fill property color of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies there is no transparency; a value of one specifies maximum transparency.

2.4.4.119 FillForegnd

The **FillForegnd** cell is a vColor custom input type that specifies the color of the foreground fill property of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.120 FillForegndTrans

The **FillForegndTrans** cell is a vScalar custom token grouping that specifies the transparency level of the foreground fill property color of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies there is no transparency; a value of one specifies maximum transparency.

2.4.4.121 FillGradientAngle

The **FillGradientAngle** cell is a vAngle custom token grouping that specifies the orientation of the fill color gradient of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

A value of zero corresponds to the direction of the positive x-axis. If the sibling FillGradientDir Cell_Type element is not equal to zero, this value MUST be ignored.

2.4.4.122 FillGradientDir

The **FillGradientDir** cell is a PtgByte parse token that specifies the type of the fill color gradient of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies a linear fill color gradient.
1	Specifies the fill color gradient of the shape is in radial mode from the bottom right corner of the

Value	Meaning
	bounding box of the shape.
2	Specifies the fill color gradient of the shape is in radial mode from the bottom left corner of the bounding box of the shape.
3	Specifies the fill color gradient of the shape is in radial mode from the center of the shape.
4	Specifies the fill color gradient of the shape is in radial mode from the center of the bottom edge of the shape.
5	Specifies the fill color gradient of the shape is in radial mode from the center of the top edge of the shape.
6	Specifies the fill color gradient of the shape is in radial mode from the top right corner of the bounding box of the shape.
7	Specifies the fill color gradient of the shape is in radial mode from the top left corner of the bounding box of the shape.
8	Specifies the fill color gradient of the shape is in rectangle mode from the bottom right corner of the bounding box of the shape.
9	Specifies the fill color gradient of the shape is in rectangle mode from the bottom left corner of the bounding box of the shape.
10	Specifies the fill color gradient of the shape is in rectangle mode from the center of the shape.
11	Specifies the fill color gradient of the shape is in rectangle mode from the top right corner of the bounding box of the shape.
12	Specifies the fill color gradient of the shape is in rectangle mode from the top left corner of the bounding box of the shape.

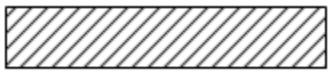
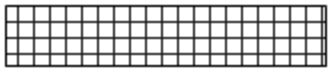
2.4.4.123 FillGradientEnabled

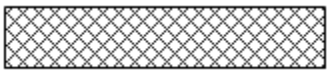
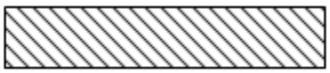
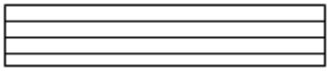
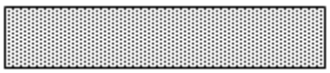
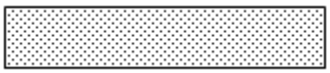
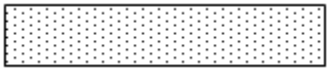
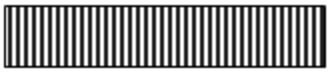
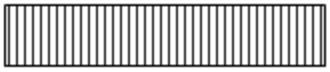
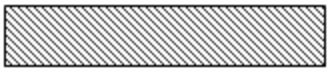
The **FillGradientEnabled** cell is a PtgBool parse token that specifies whether the fill color gradient is visible. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

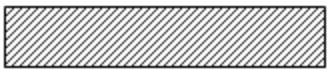
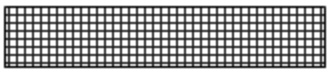
2.4.4.124 FillPattern

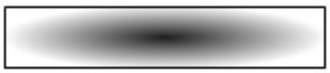
The **FillPattern** cell is a PtgByte parse token that specifies the fill pattern property of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

It MUST have a value from the following table.

Value	Meaning
0	Specifies a transparent fill.
1	Specifies a solid fill color.
2	
3	

Value	Meaning
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	

Value	Meaning
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	

Value	Meaning
40	
254	Use the master that is specified by the formula associated with this cell for the fill pattern.

2.4.4.125 Flags

The **Flags** cell is a PtgByte parse token that specifies the direction text property of a text paragraph. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

A value of zero specifies that the text direction is left to right; a value of one specifies that the text direction is right to left. Other values are used during formula evaluation only.

2.4.4.126 FlipX

The **FlipX** cell is a vScalar custom token grouping that specifies whether the coordinate system of a shape has an inverted x-axis. It MUST have a ShapeSheet_Type parent element.

The value of the structure MUST be equal to zero or one. A value of one specifies that the axis is inverted; a value of zero specifies that the axis is not inverted.

2.4.4.127 FlipY

The **FlipY** cell is a vScalar custom token grouping that specifies whether the coordinate system of a shape has an inverted y-axis. It MUST have a ShapeSheet_Type parent element.

The value of the structure MUST be equal to zero or one. A value of one specifies that the axis is inverted; a value of zero specifies that the axis is not inverted.

2.4.4.128 FlyoutChild

The **FlyoutChild** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Actions Section_Type parent element.

2.4.4.129 Font

The **Font** cell is a vFont structure that specifies the font used for the text in a text run. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

2.4.4.130 FontScale

The **FontScale** cell is a vScalar custom token grouping that specifies the font width of a text run. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

The value of the structure MUST be greater than or equal to zero and less than or equal to six. The value is normalized such that a value of 1 corresponds to 100 percent and the value of 6 corresponds to 600 percent.

2.4.4.131 FontSchemeIndex

The **FontSchemeIndex** cell is a PtgByte parse token that specifies the index of the font scheme dynamic theme component. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

The value of the **V** attribute MUST be greater than or equal to 0 and less than or equal to 65534.

If the value of the ColorSchemeIndex Cell_Type element is equal to 0, font is specified by the root style sheet.

If the value of the **V** attribute is equal to 65534, the index of the font scheme dynamic theme component of a shape is specified by the **V** attribute of the **FontSchemeIndex** Cell_Type descendant element of the PageSheet_Type element containing the shape.

2.4.4.132 Format

The **Format** cell is a vFormatString structure that specifies the data formatting to apply to a text field or shape data value. It MUST have a Row_Type parent element that has a Field or Property Section_Type parent element.

The formatting to apply is based on the type of the Row_Type element specified by a sibling Cell_Type element with an **N** attribute equal to "Type".

2.4.4.133 Frame

The **Frame** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Hyperlink Section_Type parent element.

2.4.4.134 Gamma

The **Gamma** cell is a vScalar custom token grouping that specifies the **gamma correction** applied to an image. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element. When the value of the structure is equal to one, no correction is applied.

2.4.4.135 GlowColor

The **GlowColor** cell is a vColor custom input type that specifies the color of a glow effect set of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.136 GlowColorTrans

The **GlowColorTrans** cell is a vScalar custom token grouping that specifies the transparency of the color of a glow effect set of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies there is no transparency; a value of one specifies maximum transparency.

2.4.4.137 GlowSize

The **GlowSize** cell is a vLength custom token grouping that specifies the thickness of a glow effect set of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value MUST be greater than or equal to zero points and less than or equal to 150 points.

2.4.4.138 Glue

The **Glue** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Layer Section_Type parent element.

2.4.4.139 GlueType

The **GlueType** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.140 GradientStopColor

The **GradientStopColor** cell is a vColor custom input type that specifies the color of a gradient stop of a shape. It MUST have a LineGradient Section_Type or FillGradient Section_Type parent element.

2.4.4.141 GradientStopColorTrans

The **GradientStopColorTrans** cell is a vScalar custom token grouping that specifies the transparency of the color of a gradient stop of a shape. It MUST have a LineGradient Section_Type or FillGradient Section_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies there is no transparency; a value of one specifies maximum transparency.

2.4.4.142 GradientStopPosition

The **GradientStopPosition** cell is a vScalar custom token grouping that specifies the position of a gradient stop of a shape. It MUST have a LineGradient Section_Type or FillGradient Section_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies a gradient stop at the starting point of the gradient; a value of one specifies a gradient stop at the ending point of the gradient.

The gradient starting and ending points are specified by FillGradientDir or LineGradientDir and FillGradientAngle or LineGradientAngle and UseGroupGradient Cell_Type elements. The value of the structure MUST be greater than or equal to the value of preceding GradientStopPosition structures in the same section.

2.4.4.143 Height

The **Height** cell is a vLength custom token grouping that specifies the height of a shape. It MUST have a ShapeSheet_Type parent element.

2.4.4.144 HelpTopic

The **HelpTopic** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element with an **ID** attribute equal to zero.

2.4.4.145 HideForApply

The **HideForApply** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a StyleSheet_Type parent element.

2.4.4.146 HideText

The **HideText** cell is a vScalar custom token grouping that specifies whether the text of the shape is displayed. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

The value of the structure MUST be equal to zero or one. A value of one specifies that the text is not displayed; a value of zero specifies that the text is displayed.

2.4.4.147 HorzAlign

The **HorzAlign** cell is a PtgByte parse token that specifies the horizontal alignment text properties of a text paragraph. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that the paragraph is left aligned.
1	Specifies that the paragraph is centered.
2	Specifies that the paragraph is right aligned.
3	Specifies that the paragraph is justified.
4	Specifies that the paragraph is distributed.

2.4.4.148 ImgHeight

The **ImgHeight** cell is a vLength custom token grouping that specifies the height of an image. It MUST have a ShapeSheet_Type parent element.

2.4.4.149 ImgOffsetX

The **ImgOffsetX** cell is a vLength custom token grouping that specifies the horizontal offset of an image. The value is defined in relation to the coordinate system of the shape. It MUST have a ShapeSheet_Type parent element.

2.4.4.150 ImgOffsetY

The **ImgOffsetY** cell is a vLength custom token grouping that specifies the vertical offset of an image. The value is defined in relation to the coordinate system of the shape. It MUST have a ShapeSheet_Type parent element.

2.4.4.151 ImgWidth

The **ImgWidth** cell is a vLength custom token grouping that specifies the width of an image. It MUST have a ShapeSheet_Type parent element.

2.4.4.152 IndFirst

The **IndFirst** cell is a vLength custom token grouping that specifies the distance that the first line of text in each paragraph of a shape's text block is indented from the left edge of the paragraph. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

If the indentation position is located outside the boundary of the shape's text block, the closest position on the boundary of the shape's text block MUST be used in its place.

2.4.4.153 IndLeft

The **IndLeft** cell is a vLength custom token grouping that specifies the distance that all lines of text in a paragraph are indented from the left margin of the text block of a shape. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

If the indentation position is located outside the boundary of the shape's text block, the closest position within the boundary of the shape's text block MUST be used in its place.

2.4.4.154 IndRight

The **IndRight** cell is a vLength custom token grouping that specifies the distance that all lines of text in a paragraph are indented from the right margin of the text block of a shape. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

If the indentation position is located outside the boundary of the shape's text block, the closest position within the boundary of the shape's text block MUST be used in its place.

If the indentation is greater than the width of the shape less the cell value of IndLeft, the text is positioned based on the IndLeft value.

2.4.4.155 InhibitSnap

The **InhibitSnap** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.156 Initials

The **Initials** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Reviewer Section_Type parent element.

2.4.4.157 Invisible

The **Invisible** cell is a PtgBool parse token that specifies whether a hyperlink or shape data item is invisible on a shape. It MUST have a Row_Type parent element that has a Hyperlink or Property Section_Type parent element.

A structure value of one specifies that the item is invisible; a structure value of zero specifies that the item is visible.

2.4.4.158 IsDropSource

The **IsDropSource** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.159 IsDropTarget

The **IsDropTarget** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.160 IsSnapTarget

The **IsSnapTarget** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.161 IsTextEditTarget

The **IsTextEditTarget** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.162 KeepTextFlat

The **KeepTextFlat** cell is a PtgBool parse token that specifies whether 3D rotation properties (section 2.2.7.3.7) apply to the text of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

A value of one specifies that 2.2.7.3D rotation properties do not apply to the shape text; a value of zero specifies that 3D rotation properties apply to the shape text.

2.4.4.163 Label

The **Label** cell is a PtgString parse token that specifies the label of a shape data. It MUST have a Row_Type parent element that has a Property Section_Type parent element.

2.4.4.164 LangID

The **LangID** cell is a vLanguageString structure that specifies the locale settings to use for character properties and paragraph properties, as specified in [RFC4646] and [RFC4647]. It MUST have either a ShapeSheet_Type parent element or a Row_Type parent element that has a Character or Property Section_Type parent element.

2.4.4.165 LayerMember

The **LayerMember** cell is a PtgString parse token that specifies a layer index list to which a shape is assigned. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element with an **ID** attribute equal to zero.

A layer index list is a semicolon-delimited collection of layer indices defined by the following.

ABNF

```
layer-index-list = [layer-index *(";" layer-index)]
layer-index = 1*DIGIT
```

2.4.4.166 LeftMargin

The **LeftMargin** cell is a vLength custom token grouping that specifies the margin between the left border of a text block and the text it contains. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

2.4.4.167 Letterspace

The **Letterspace** cell is a vLength custom token grouping that specifies the amount of space added or subtracted between characters in a text run. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

2.4.4.168 LineAdjustFrom

The **LineAdjustFrom** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.169 LineAdjustTo

The **LineAdjustTo** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.170 LineCap

The **LineCap** cell is a PtgByte parse token that specifies the line end style for a shape. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Rounded
1	Square
2	Extended

2.4.4.171 LineColor

The **LineColor** cell is a vColor custom input type that specifies the line color of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.172 LineColorTrans

The **LineColorTrans** cell is a vScalar custom token grouping that specifies the transparency of the line color of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies that the image is completely opaque; a value of one specifies that the image is completely transparent.

2.4.4.173 LineGradientAngle

The **LineGradientAngle** cell is a vAngle custom token grouping that specifies the orientation of the line color gradient of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

A value of zero corresponds to the direction of the positive x-axis. If the sibling FillGradientDir Cell_Type element is not equal to zero, this value MUST be ignored.

2.4.4.174 LineGradientDir

The **LineGradientDir** cell is a PtgByte parse token that specifies the type of the line color gradient of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies a linear line color gradient.
1	Specifies the line color gradient of the shape is in radial mode from the bottom right corner of the bounding box of the shape.
2	Specifies the line color gradient of the shape is in radial mode from bottom left corner of the bounding box of the shape.
3	Specifies the line color gradient of the shape is in radial mode from center of the shape.
4	Specifies the line color gradient of the shape is in radial mode from the center of the bottom edge of the shape.
5	Specifies the line color gradient of the shape is in radial mode from the center of the top edge of the shape.
6	Specifies the line color gradient of the shape is in radial mode from top right corner of the bounding box of the shape.
7	Specifies the line color gradient of the shape is in radial mode from top left corner of the bounding box of the shape.
8	Specifies the line color gradient of the shape is in rectangle mode from bottom right corner of the bounding box of the shape.
9	Specifies the line color gradient of the shape is in rectangle mode from bottom left corner of the bounding box of the shape.
10	Specifies the line color gradient of the shape is in rectangle mode from center of the shape.
11	Specifies the line color gradient of the shape is in rectangle mode from top right corner of the bounding box of the shape.
12	Specifies the line color gradient of the shape is in rectangle mode from top left corner of the bounding box of the shape.

2.4.4.175 LineGradientEnabled

The **LineGradientEnabled** cell is a PtgBool parse token that specifies whether the line gradient is visible. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.176 LineJumpCode

The **LineJumpCode** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.177 LineJumpFactorX

The **LineJumpFactorX** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.178 LineJumpFactorY

The **LineJumpFactorY** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.179 LineJumpStyle

The **LineJumpStyle** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.180 LinePattern

The **LinePattern** cell is a PtgByte parse token that specifies the line pattern of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies transparent line.
1	Specifies a solid line.
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

Value	Meaning
15	
16	
17	
18	
19	
20	
21	
22	
23	
254	Use the master that is specified by the formula associated with this cell for the line pattern.

2.4.4.181 LineRouteExt

The **LineRouteExt** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.182 LineToLineX

The **LineToLineX** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.183 LineToLineY

The **LineToLineY** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.184 LineToNodeX

The **LineToNodeX** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.185 LineToNodeY

The **LineToNodeY** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.186 LineWeight

The **LineWeight** cell is a vLength custom token grouping that specifies the line thickness of a shape. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element. It MUST be greater than zero.

2.4.4.187 LocalizeMerge

The **LocalizeMerge** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

2.4.4.188 Lock

The **Lock** cell is a PtgBool parse token that specifies whether the shapes in a layer are selectable. It MUST have a Row_Type parent element that has a Layer Section_Type parent element.

A value of one specifies that the shapes in this layer are not selectable; a value of zero specifies the shapes in this layer can be selected.

2.4.4.189 LockAspect

The **LockAspect** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.190 LockBegin

The **LockBegin** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.191 LockCalcWH

The **LockCalcWH** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.192 LockCrop

The **LockCrop** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.193 LockCustProp

The **LockCustProp** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.194 LockDelete

The **LockDelete** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.195 LockEnd

The **LockEnd** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.196 LockFormat

The **LockFormat** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.197 LockFromGroupFormat

The **LockFromGroupFormat** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.198 LockGroup

The **LockGroup** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.199 LockHeight

The **LockHeight** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.200 LockMoveX

The **LockMoveX** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.201 LockMoveY

The **LockMoveY** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.202 LockPreview

The **LockPreview** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a DocumentSheet_Type parent element.

2.4.4.203 LockReplace

The **LockReplace** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.204 LockRotate

The **LockRotate** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.205 LockSelect

The **LockSelect** cell is a PtgBool parse token that specifies whether a shape is selectable. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element. If the ProtectShapes_Type element in the web drawing has a value equal to zero, it is used during formula evaluation only.

A value of one specifies that the shape is not selectable; a value of zero specifies the shape is selectable.

2.4.4.206 LockTextEdit

The **LockTextEdit** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.207 LockThemeColors

The **LockThemeColors** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.208 LockThemeConnectors

The **LockThemeConnectors** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.209 LockThemeEffects

The **LockThemeEffects** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.210 LockThemeFonts

The **LockThemeFonts** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.211 LockThemeIndex

The **LockThemeIndex** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.212 LockVariation

The **LockVariation** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.213 LockVtxEdit

The **LockVtxEdit** is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.214 LockWidth

The **LockWidth** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.215 LocPinX

The **LocPinX** cell is a vLength custom token grouping that specifies the x-coordinate of the center of rotation of a shape. The value is defined in relation to the coordinate system of the shape. It MUST have a ShapeSheet_Type parent element.

2.4.4.216 LocPinY

The **LocPinY** cell is a vLength custom token grouping that specifies the y-coordinate of the center of rotation of a shape. The value is defined in relation to the coordinate system of the shape. It MUST have a ShapeSheet_Type parent element.

2.4.4.217 Menu

The **Menu** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has an Actions Section_Type parent element.

2.4.4.218 Name

The **Name** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Layer Section_Type parent element or a Row_Type parent element that has a Reviewer Section_Type parent element.

2.4.4.219 NameUniv

The **NameUniv** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Layer Section_Type parent element.

2.4.4.220 NewWindow

The **NewWindow** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Hyperlink Section_Type parent element.

2.4.4.221 NoAlignBox

The **NoAlignBox** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.222 NoCoauth

The **NoCoauth** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a DocumentSheet_Type parent element.

2.4.4.223 NoCtlHandles

The **NoCtlHandles** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.224 NoFill

The **NoFill** cell is a PtgBool parse token that specifies whether fill properties are applied to a geometric path. It MUST have a Geometry Section_Type parent element.

A value of one specifies that fill properties are not applied to the geometric path; a value of zero specifies that fill properties are applied to the geometric path.

2.4.4.225 NoLine

The **NoLine** cell is a PtgBool parse token that specifies whether line properties are applied to a geometric path. It MUST have a Geometry Section_Type parent element.

A value of one specifies that line properties are not applied to the geometric path; a value of zero specifies that line properties are applied to the geometric path.

2.4.4.226 NoLiveDynamics

The **NoLiveDynamics** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.227 NonPrinting

The **NonPrinting** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.228 NoObjHandles

The **NoObjHandles** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.229 NoProofing

The **NoProofing** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.230 NoQuickDrag

The **NoQuickDrag** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Geometry Section_Type parent element.

2.4.4.231 NoShow

The **NoShow** cell is a PtgBool parse token that specifies whether a geometric path is visible. It MUST have a Geometry Section_Type parent element.

A value of one specifies that the geometric path is hidden; a value of zero specifies that the geometric path is visible.

2.4.4.232 NoSnap

The **NoSnap** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Geometry Section_Type parent element.

2.4.4.233 ObjectKind

The **ObjectKind** cell is a PtgByte parse token that specifies the direction of text for a text field. It MUST have a Row_Type parent element that has a Field Section_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that the text field direction matches the text direction.
1	Specifies that the text field direction is horizontal.

2.4.4.234 ObjType

The **ObjType** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.235 OnPage

The **OnPage** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.236 OutputFormat

The **OutputFormat** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a DocumentSheet_Type parent element.

2.4.4.237 Overline

The **Overline** cell is a PtgBool parse token that specifies whether the text run has an over line character property. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

A value of one specifies that the text run has an over line; a value of zero specifies that the text run does not have an over line.

2.4.4.238 PageBottomMargin

The **PageBottomMargin** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.239 PageHeight

The **PageHeight** cell is a vLength custom token grouping that specifies the height of a drawing page. It MUST have a PageSheet_Type parent element. The value of the structure MUST be greater than zero.

2.4.4.240 PageLeftMargin

The **PageLeftMargin** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.241 PageLineJumpDirX

The **PageLineJumpDirX** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.242 PageLineJumpDirY

The **PageLineJumpDirY** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.243 PageLockDuplicate

The **PageLockDuplicate** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.244 PageLockReplace

The **PageLockReplace** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.245 PageRightMargin

The **PageRightMargin** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.246 PageScale

The **PageScale** cell is a vLength custom token grouping that specifies a unit of distance used to define the scale of the web drawing. It MUST have a PageSheet_Type parent element.

The default value of the structure is one inch.

2.4.4.247 PageShapeSplit

The **PageShapeSplit** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.248 PagesX

The **PagesX** cell is a PtgUnsShort parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.249 PagesY

The **PagesY** cell is a PtgUnsShort parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.250 PageTopMargin

The **PageTopMargin** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.251 PageWidth

The **PageWidth** cell is a vLength custom token grouping that specifies the width of a drawing page. It MUST have a PageSheet_Type parent element.

The value of the structure MUST be greater than zero.

2.4.4.252 PaperKind

The **PaperKind** cell is a PtgUnsShort parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.253 PaperSource

The **PaperSource** cell is a PtgUnsShort parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.254 Perspective

The **Perspective** cell is a vAngle or vScalar custom token grouping that specifies the angle of view for a shape with 3D rotation properties (section 2.2.7.3.7). It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

If the cell is a vScalar, the value of the structure MUST be expressed as an angleInternalUnitNumber custom internal unit number. The value of the structure MUST be greater than or equal to zero, and less than or equal to 120 degrees. The value of zero specifies that the perspective projection is not applied.

If the sibling **RotationType** (section 2.4.4.294) **Cell_Type** (section 2.3.4.2.5) element does not have a value of two, this value MUST be zero.

2.4.4.255 PinX

The **PinX** cell is a vLength custom token grouping that specifies the x-coordinate of the center of rotation of a shape. The value is defined in relation to the coordinate system of the shape's parent. It MUST have a ShapeSheet_Type parent element.

2.4.4.256 PinY

The **PinY** cell is a vLength custom token grouping that specifies the y-coordinate of the center of rotation of a shape. The value is defined in relation to the coordinate system of the shape's parent. It MUST have a ShapeSheet_Type parent element.

2.4.4.257 PlaceDepth

The **PlaceDepth** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.258 PlaceFlip

The **PlaceFlip** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.259 PlaceStyle

The **PlaceStyle** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.260 PlowCode

The **PlowCode** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.261 Pos

The **Pos** cell is a PtgByte parse token that specifies the position character properties used to format a text run. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that the position of the text run is unchanged.
1	Specifies that the text run is displayed as superscript.
2	Specifies that the text run is displayed as subscript.

2.4.4.262 Position

The **Position** cell is a vLength custom token grouping that specifies the tab stop position for a tabs property. It MUST have a Row_Type parent element that has a Tabs Section_Type parent element. The value is defined in relation to the coordinate system of the shape and relative to the edge of the shape's text block. It MUST have an **N** attribute with the following format.

Position#

Where # is an unsigned long integer, and MUST be less than or equal to 59.

2.4.4.263 PreviewQuality

The **PreviewQuality** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a DocumentSheet_Type parent element.

2.4.4.264 PreviewScope

The **PreviewScope** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a DocumentSheet_Type parent element.

2.4.4.265 Print

The **Print** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Layer Section_Type parent element.

2.4.4.266 PrintGrid

The **PrintGrid** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.267 PrintPageOrientation

The **PrintPageOrientation** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.268 Prompt

The **Prompt** cell is a PtgString parse token that is used during formula evaluation only. It MUST have either a Row_Type parent element that has a Connection Section_Type parent element, a Row_Type parent element that has a Control Section_Type parent element, a Row_Type parent element that has a Property Section_Type parent element, or a Row_Type parent element that has a User Section_Type parent element.

2.4.4.269 QuickStyleEffectsMatrix

The **QuickStyleEffectsMatrix** cell is a PtgByte parse token that specifies the index of the effects matrix quick style slice as specified in section 2.2.7.4.3. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to 0, and less than or equal to 103.

If the value of the structure is greater than or equal to 1 and less than or equal to 6, the set of effect properties is specified by a dynamic theme for the ShdwForegndTrans, ShdwPattern, ShapeShdwType, ShapeShdwOffsetX, ShapeShdwOffsetY, ShapeShdwObliqueAngle, ShapeShdwScaleFactor, ShapeShdwBlur, BevelTopType, BevelTopWidth, BevelTopHeight, BevelContourColor, BevelContourSize, BevelMaterialType, BevelLightingType, BevelLightingAngle, ReflectionTrans, ReflectionDist, ReflectionSize, ReflectionBlur, SketchEnabled, SketchSeed, SketchAmount, SketchLineWeight, SketchLineChange, SketchFillChange, GlowColor, GlowColorTrans, GlowSize, and SoftEdgesSize Cell_Type elements.

If the value of the structure is equal to 0, the set of effect properties is specified by the root style sheet for the ShdwForegndTrans, ShdwPattern, ShapeShdwType, ShapeShdwOffsetX, ShapeShdwOffsetY, ShapeShdwObliqueAngle, ShapeShdwScaleFactor, ShapeShdwBlur, BevelTopType, BevelTopWidth, BevelTopHeight, BevelContourColor, BevelContourSize, BevelMaterialType, BevelLightingType, BevelLightingAngle, ReflectionTrans, ReflectionDist, ReflectionSize, ReflectionBlur, SketchEnabled, SketchSeed, SketchAmount, SketchLineWeight, SketchLineChange, SketchFillChange, GlowColor, GlowColorTrans, GlowSize, and SoftEdgesSize Cell_Type elements.

If the value of the structure is greater than or equal to 100 and less than or equal to 103, the set of effect properties specified by a dynamic theme is derived by evaluation of the value of the structure. The following table specifies the meaning of each value.

Value of the structure	Evaluation
100	The set of effect properties specified by a dynamic theme is specified by the **effectIdx** attribute specified by the first CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
101	The set of effect properties specified by a dynamic theme is specified by the **effectIdx** attribute specified by the second CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
102	The set of effect properties specified by a dynamic theme is specified by the **effectIdx** attribute specified by the third CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
103	The set of effect properties specified by a dynamic theme is specified by the **effectIdx** attribute specified by the fourth CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type

Value of the structure	Evaluation
	element of a shape.

2.4.4.270 QuickStyleFillColor

The **QuickStyleFillColor** cell is a PtgByte parse token that specifies the index of the fill color quick style slice. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure specifies a color from the color scheme dynamic theme component for the FillForegnd and FillBkgnd Cell_Type elements, and the Color Cell_Type child element of a Row_Type that has a FillGradient Section_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that the **dk1** color from the color scheme dynamic theme component as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.9 is used.
1	Specifies that the **lt1** color from the color scheme dynamic theme component as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.22 is used.
2	Specifies that the **accent1** color from the color scheme dynamic theme component as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.1 is used.
3	Specifies that the **accent2** color from the color scheme dynamic theme component as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.2 is used.
4	Specifies that the **accent3** color from the color scheme dynamic theme component as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.3 is used.
5	Specifies that the **accent4** color from the color scheme dynamic theme component as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.4 is used.
6	Specifies that the **accent5** color from the color scheme dynamic theme component as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.5 is used.
7	Specifies that the **accent6** color from the color scheme dynamic theme component as specified in [ISO/IEC29500-1:2011] section 20.1.4.1.6 is used.
8	Specifies that the background color from the color scheme dynamic theme component is used.
100	Specifies that the color property specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 child element of a CT_VariationClrScheme element specified by the VariationColorIndex Cell_Type element of a shape is used.
101	Specifies that the color property specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 child element of a CT_VariationClrScheme element specified by the VariationColorIndex Cell_Type element of a shape is used.
102	Specifies that the color property specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 child element of a CT_VariationClrScheme element specified by the VariationColorIndex Cell_Type element of a shape is used.
103	Specifies that the color property specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 child element of a CT_VariationClrScheme element specified by the VariationColorIndex Cell_Type element of a shape is used.
104	Specifies that the color property specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 child element of a CT_VariationClrScheme element specified by the VariationColorIndex Cell_Type element of a shape is used.
105	Specifies that the color property specified by the **CT_Color** type specified in [ISO/IEC29500-

Value	Meaning
	1:2011] section A.2 child element of a CT_VariationClrScheme element specified by the VariationColorIndex Cell_Type element of a shape is used.
106	Specifies that the color property specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section A.2 child element of a CT_VariationClrScheme element specified by the VariationColorIndex Cell_Type element of a shape is used.

2.4.4.271 QuickStyleFillMatrix

The **QuickStyleFillMatrix** cell is a PtgByte parse token that specifies the index of the fill matrix quick style slice. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to 0, and less than or equal to 103.

If the value of the structure is greater than or equal to 1 and less than or equal to 6, the set of fill properties is specified by a dynamic theme for the FillForegndTrans, FillBkgndTrans, FillPattern, FillGradientDir, FillGradientAngle, FillGradientEnabled, RotateGradientWithShape, UseGroupGradient Cell_Type elements, and the ColorTrans and Position Cell_Type child elements of a Row_Type that has a FillGradient Section_Type parent element.

If the value of the structure is equal to 0, the set of fill properties is specified by the root style sheet for the cells belonging to the FillForegndTrans, FillBkgndTrans, FillPattern, FillGradientDir, FillGradientAngle, FillGradientEnabled, RotateGradientWithShape, UseGroupGradient Cell_Type elements, and the ColorTrans and Position Cell_Type child elements of a Row_Type that has a FillGradient Section_Type parent element.

If the value of the structure is greater than or equal to 100 and less than or equal to 103, the set of fill properties specified by a dynamic theme is derived by evaluation of the value of the structure. The following table specifies the meaning of each value.

Value of the structure	Evaluation
100	The set of fill properties specified by a dynamic theme is specified by the **fillIdx** attribute specified by the first CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
101	The set of fill properties specified by a dynamic theme is specified by the **fillIdx** attribute specified by the second CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
102	The set of fill properties specified by a dynamic theme is specified by the **fillIdx** attribute specified by the third CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
103	The set of fill properties specified by a dynamic theme is specified by the **fillIdx** attribute specified by the fourth CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.

2.4.4.272 QuickStyleFontColor

The **QuickStyleFontColor** cell is a PtgByte parse token that specifies the index of the font color quick style slice. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure specifies a color from the color scheme dynamic theme component for the Color Cell_Type child element of a Row_Type that has a Character Section_Type parent element.

The value of the structure MUST be specified by the table in QuickStyleFillColor cell.

2.4.4.273 QuickStyleFontMatrix

The **QuickStyleFontMatrix** cell is a PtgByte parse token that specifies the index of the font matrix quick style slice. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to 0, and less than or equal to 103.

If the value of the structure is greater than or equal to 1 and less than or equal to 6, the set of Fonts is specified by a dynamic theme for the cells belonging to the Font, AsianFont, ComplexScriptFont, and Style Cell_Type child elements of a Row_Type that has a Character Section_Type parent element.

If the value of the structure is equal to 0, the set of fonts is specified by the root style sheet for the cells belonging to the Font, AsianFont, ComplexScriptFont, and Style Cell_Type child elements of a Row_Type that has a Character Section_Type parent element.

If the value of the structure is greater than or equal to 100 and less than or equal to 103, the set of fonts specified by a dynamic theme is derived by evaluation of the value of the structure. The following table specifies the meaning of each value.

Value of the structure	Evaluation
100	The set of fonts specified by a dynamic theme is specified by the **fontIdx** attribute specified by the first CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
101	The set of fonts specified by a dynamic theme is specified by the **fontIdx** attribute specified by the second CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
102	The set of fonts specified by a dynamic theme is specified by the **fontIdx** attribute specified by the third CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
103	The set of fonts specified by a dynamic theme is specified by the **fontIdx** attribute specified by the fourth CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.

2.4.4.274 QuickStyleLineColor

The **QuickStyleLineColor** cell is a PtgByte parse token that specifies the index of the line color quick style slice. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure specifies a color from the color scheme dynamic theme component for the LineColor Cell_Type and the Color Cell_Type child element of a Row_Type that has a LineGradient Section_Type parent element.

The value of the structure MUST be specified by the table in QuickStyleFillColor cell.

2.4.4.275 QuickStyleLineMatrix

The **QuickStyleLineMatrix** cell is a PtgByte parse token that specifies the index of the line matrix quick style slice. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to 0, and less than or equal to 103.

If the value of the structure is greater than or equal to 1 and less than or equal to 6, the set of line properties is specified by a dynamic theme for the LinePattern, LineWeight, LineCap, BeginArrow, EndArrow, LineColorTrans, CompoundType, BeginArrowSize, EndArrowSize, Rounding, LineGradientDir, LineGradientAngle, and LineGradientEnabled Cell_Type elements and the ColorTrans and Position Cell_Type child elements of a Row_Type that has a LineGradient Section_Type parent element.

If the value of the structure is equal to 0, the set of line properties is specified by the root style sheet for the LinePattern, LineWeight, LineCap, BeginArrow, EndArrow, LineColorTrans, CompoundType, BeginArrowSize, EndArrowSize, Rounding, LineGradientDir, LineGradientAngle, and LineGradientEnabled Cell_Type elements and the ColorTrans and Position Cell_Type child elements of a Row_Type that has a LineGradient Section_Type parent element.

If the value of the structure is greater than or equal to 100 and less than or equal to 103, the set of line properties specified by a dynamic theme is derived by evaluation of the value of the structure. The following table specifies the meaning of each value.

Value of the structure	Evaluation
100	The set of line properties specified by a dynamic theme is specified by the **lineIdx** attribute specified by the first CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
101	The set of line properties specified by a dynamic theme is specified by the **lineIdx** attribute specified by the second CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
102	The set of line properties specified by a dynamic theme is specified by the **lineIdx** attribute specified by the third CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.
103	The set of line properties specified by a dynamic theme is specified by the **lineIdx** attribute specified by the fourth CT_VarStyle child element of a CT_VariationStyleScheme element specified by the VariationStyleIndex Cell_Type element of a shape.

2.4.4.276 QuickStyleShadowColor

The **QuickStyleShadowColor** cell is a PtgByte parse token that specifies the index of the shadow color quick style slice. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure specifies a color from the color scheme dynamic theme component for the ShdwForegnd Cell_Type.

The value of the structure MUST be specified by the table in QuickStyleFillColor cell.

2.4.4.277 QuickStyleType

The **QuickStyleType** cell is a PtgInt parse token that specifies whether the QuickStyleLineMatrix, QuickStyleFillMatrix, and QuickStyleEffectsMatrix Cell_Type elements of a shape (section 2.2.3), master, or style refer to the effect scheme dynamic theme component or the connector scheme dynamic theme component regardless of whether the shape, master, or style is a connector. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that the QuickStyleLineMatrix, QuickStyleFillMatrix, and QuickStyleEffectsMatrix Cell_Type elements refer to an effect scheme dynamic theme component if the shape, master, or style is a non-connector, or to a connector scheme dynamic theme component if the shape, master, or style is a connector.
1 and 2	Specifies that the QuickStyleLineMatrix, QuickStyleFillMatrix, and QuickStyleEffectsMatrix Cell_Type elements refer to an effect scheme dynamic theme component regardless if the shape, master, or style is a connector.
3	Specifies that the QuickStyleLineMatrix, QuickStyleFillMatrix, and QuickStyleEffectsMatrix Cell_Type elements refer to a connector scheme dynamic theme component regardless if the shape, master, or style is a connector.

2.4.4.278 QuickStyleVariation

The QuickStyleVariation cell is a PtgByte parse token that can affect formula evaluation of the ThemeVal function token. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be a bitwise OR combination of one or more of the values from the following table.

Value	Meaning
0x0	Specifies no effect on formula evaluation of the ThemeVal function token.
0x1	Specifies no effect on formula evaluation of the ThemeVal function token.
0x2	Specifies that formula evaluation of the "TextColor" vDynamicThemeString **ThemeProperty**

Value	Meaning
	argument for the ThemeVal function token is affected as follows. If the absolute difference in luminance in **hue-saturation-luminance (HSL) color space** between the formula evaluation of the "BackgroundColor" and "TextColor" vDynamicThemeString **ThemeProperty** arguments for the ThemeVal function token is greater than or equal to 16.66%, then formula evaluation of the "TextColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token is not affected. Otherwise, if the luminance of the formula evaluation of the "BackgroundColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token is less than or equal to 72.92%, then formula evaluation of the "TextColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token MUST return a color equal to RGB(255, 255, 255); otherwise it MUST return the color with the largest absolute difference in luminance from the formula evaluation of the "TextColor", "FillColor", and "LineColor" vDynamicThemeString **ThemeProperty** arguments for the ThemeVal function token.
0x4	Specifies that formula evaluation of the "LineColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token is affected as follows. If the absolute difference in luminance in hue-saturation-luminance (HSL) color space between the formula evaluation of the "BackgroundColor" and "LineColor" vDynamicThemeString **ThemeProperty** arguments for the ThemeVal function token is greater than or equal to 16.66%, then formula evaluation of the "LineColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token is not affected. Otherwise, if the luminance of the formula evaluation of the "BackgroundColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token is less than or equal to 72.92%, then formula evaluation of the "LineColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token MUST return a color equal to RGB(255, 255, 255); otherwise it MUST return the color with the largest absolute difference in luminance from the formula evaluation of the "FillColor" and "LineColor" vDynamicThemeString **ThemeProperty** arguments for the ThemeVal function token.
0x8	Specifies that formula evaluation of the "FillColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token may be affected. If the absolute difference in luminance in hue-saturation-luminance (HSL) color space between the formula evaluation of the "BackgroundColor" and "FillColor" vDynamicThemeString **ThemeProperty** arguments for the ThemeVal function token is greater than or equal to 16.66%, then formula evaluation of the "FillColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token is not affected. Otherwise, if the luminance of the formula evaluation of the "BackgroundColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token is less than or equal to 72.92%, then formula evaluation of the "FillColor" vDynamicThemeString **ThemeProperty** argument for the ThemeVal function token MUST return a color equal to RGB(255, 255, 255); otherwise it MUST return the color with the largest absolute difference in luminance from the formula evaluation of the "FillColor" and "LineColor" vDynamicThemeString **ThemeProperty** arguments for the ThemeVal function token.

2.4.4.279 ReadOnly

The **ReadOnly** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Actions Section_Type parent element.

2.4.4.280 ReflectionBlur

The **ReflectionBlur** cell is a vLength custom token grouping that specifies the amount of blur of a shape's reflection. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to zero, and less than or equal to (100.0 / 72.0) inches. The value of zero specifies that there is no blur.

2.4.4.281 ReflectionDist

The **ReflectionDist** cell is a vLength custom token grouping that specifies the distance that the reflection is offset from a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to zero, and less than or equal to (100.0 / 72.0) inches.

2.4.4.282 ReflectionSize

The **ReflectionSize** cell is a vScalar custom token grouping that specifies the percentage of the reflection size relative to a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent.

2.4.4.283 ReflectionTrans

The **ReflectionTrans** cell is a vScalar custom token grouping that specifies the reflection transparency of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage, and MUST be greater than or equal to zero and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent.

2.4.4.284 Relationships

The **Relationships** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.285 ReplaceCopyCells

The **ReplaceCopyCells** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.286 ReplaceLockFormat

The **ReplaceLockFormat** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element with an **ID** attribute equal to zero.

2.4.4.287 ReplaceLockShapeData

The **ReplaceLockShapeData** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element with an **ID** attribute equal to zero.

2.4.4.288 ReplaceLockText

The **ReplaceLockText** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element with an **ID** attribute equal to zero.

2.4.4.289 ResizeMode

The **ResizeMode** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element.

2.4.4.290 ResizePage

The **ResizePage** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.291 ReviewerID

The **ReviewerID** cell is a PtgShort parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Reviewer Section_Type parent element.

2.4.4.292 RightMargin

The **RightMargin** cell is a vLength custom token grouping that specifies the margin between the right border of a text block and the text it contains. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

2.4.4.293 RotateGradientWithShape

The **RotateGradientWithShape** cell is a PtgBool parse token that specifies whether the orientation of the fill color gradient property rotates along with the rotation of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

A value of one specifies that the orientation of the fill color gradient property rotates along with the rotation of the shape; a value of zero specifies that the orientation of the fill color gradient property does not rotate along with the rotation of the shape.

2.4.4.294 RotationType

The **RotationType** cell is a PtgByte parse token that specifies the type of projection of the effect properties of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies no 3D effects rotation.
1	Specifies that a parallel projection is applied to the 3D effect properties.

Value	Meaning
2	Specifies that the shape rotates in perspective projection.
3	Specifies that the shape rotates in oblique projection from the top left corner of the bounding box of the shape.
4	Specifies that the shape rotates oblique projection from the top right corner of the bounding box of the shape.
5	Specifies that the shape rotates oblique projection from the bottom left corner of the bounding box of the shape.
6	Specifies that the shape rotates oblique projection from the bottom right corner of the bounding box of the shape.

2.4.4.295 RotationXAngle

The **RotationXAngle** cell is a vAngle custom token grouping that specifies the counterclockwise rotation angle of a shape around the y-axis. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.296 RotationYAngle

The **RotationYAngle** cell is a vAngle custom token grouping that specifies the counterclockwise rotation angle of a shape around the x-axis. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.297 RotationZAngle

The **RotationZAngle** cell is a vAngle custom token grouping that specifies the counterclockwise rotation angle of a shape around the z-axis. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.298 Rounding

The **Rounding** cell is a vLength custom token grouping that specifies the rounding radius of the outline of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to zero inches. The value of zero specifies that there is no rounding. A value greater than zero specifies that any corner between two line segments, a line segment and an elliptical arc, or two elliptical arcs within the outline is rounded with a radius equal to the value.

2.4.4.299 RouteStyle

The **RouteStyle** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.300 ScaleX

The **ScaleX** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element. It MUST be greater than zero.

2.4.4.301 ScaleY

The **ScaleY** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element. It MUST be greater than zero.

2.4.4.302 SelectMode

The **SelectMode** cell is a PtgByte parse token that specifies information about the selection behavior of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that the subshapes of the shape are not selectable.
1	Specifies that the subshapes of the shape are selectable if other conditions for shape selection hold.
2	Specifies that the subshapes of the shape are selectable if other conditions for shape selection hold.

2.4.4.303 ShapeFixedCode

The **ShapeFixedCode** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.304 ShapeKeywords

The **ShapeKeywords** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.305 ShapePermeablePlace

The **ShapePermeablePlace** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.306 ShapePermeableX

The **ShapePermeableX** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.307 ShapePermeableY

The **ShapePermeableY** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.308 ShapePlaceFlip

The **ShapePlaceFlip** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.309 ShapePlaceStyle

The **ShapePlaceStyle** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.310 ShapePlowCode

The **ShapePlowCode** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.311 ShapeRouteStyle

The **ShapeRouteStyle** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.312 ShapeShdwBlur

The **ShapeShdwBlur** cell is a vLength custom token grouping that specifies the blur of a shadow effect set of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to zero, and equal or less than (100.0 / 72.0) points.

2.4.4.313 ShapeShdwObliqueAngle

The **ShapeShdwObliqueAngle** cell is a vAngle custom token grouping that specifies the angle of direction of an oblique shadow effect set of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as an angleInternalUnitNumber custom internal unit number. A value of zero in this cell specifies that the angle direction is straight up and is measured moving clockwise.

2.4.4.314 ShapeShdwOffsetX

The **ShapeShdwOffsetX** cell is a vLength custom token grouping that specifies the x-coordinate value of the shadow effect set on a shape. The value is defined in relation to the coordinate system of the shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The shadow distance specified by this cell and the ShapeShdwOffsetY cell MUST be greater than or equal to zero, and equal or less than (200.0 / 72.0) inches.

2.4.4.315 ShapeShdwOffsetY

The **ShapeShdwOffsetY** cell is a vLength custom token grouping that specifies the vertical value of the shadow effect set on a shape. The value is defined in relation to the coordinate system of the shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The shadow distance specified by this cell and the ShapeShdwOffsetX cell MUST be greater than or equal to zero, and equal or less than (200.0 / 72.0) inches.

2.4.4.316 ShapeShdwScaleFactor

The **ShapeShdwScaleFactor** cell is a vScalar custom token grouping that specifies the percentage by which the shadow effect set of a shape can be enlarged or reduced. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage. It MUST be greater than or equal to zero, and less than or equal to two. The value is normalized such that a value of 2 corresponds to 200 percent. As the value of the structure increases, the shadow is enlarged.

2.4.4.317 ShapeShdwShow

The **ShapeShdwShow** cell is a PtgByte parse token that specifies whether the shadow effect set of a shape should be displayed. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that the shadow effect set is displayed only if the shape has a Geometry Section_Type.
1	Specifies that the shadow effect set is displayed only if the shape has a Geometry Section_Type and the shape is a top-level shape.
2	Specifies that the shadow effect set is always displayed.

2.4.4.318 ShapeShdwType

The **ShapeShdwType** cell is a PtgByte parse token that specifies the shadow effect set type of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be from the following table.

Value	Meaning
0	Specifies that the page default shadow is used for the shadow effect set.
1	Specifies a simple shadow.
2	Specifies an oblique shadow.
3	Specifies an inner shadow.

2.4.4.319 ShapeSplit

The **ShapeSplit** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.320 ShapeSplittable

The **ShapeSplittable** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.321 Sharpen

The **Sharpen** cell is a vScalar custom token grouping that specifies the degree to which the contrast of adjacent pixels is increased to increase the sharpness of an image. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage. It MUST be greater than or equal to zero, and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies no increase in sharpness; a value of one specifies the maximum increase in sharpness.

2.4.4.322 ShdwForegnd

The **ShdwForegnd** cell is a vColor custom input type that specifies the color used for the shadow effect set of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.323 ShdwForegndTrans

The **ShdwForegndTrans** cell is a vScalar custom token grouping that specifies the color transparency level used for the shadow effect set of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure is specified by the ColorTrans cell.

2.4.4.324 ShdwObliqueAngle

The **ShdwObliqueAngle** cell is a vScalar custom token grouping that specifies the angle of oblique direction when the page default shadow is applied. It MUST have a PageSheet_Type parent element.

The angle specified in this cell is used whenever a shape specifies a page default shadow.

The value of the structure MUST be expressed as an angleInternalUnitNumber custom internal unit number. A value of zero specifies that the angle direction is straight up and is measured moving clockwise.

To set the behavior for an individual shape, use the ShapeShdwObliqueAngle cell.

2.4.4.325 ShdwOffsetX

The **ShdwOffsetX** cell is a vLength custom token grouping that specifies the horizontal offset of a page sheet's shadow effect set. The value is defined in relation to the coordinate system of the shape. It MUST have a PageSheet_Type parent element.

The shadow distance specified by this cell and the ShdwOffsetY cell MUST be greater than or equal to zero, and equal or less than (200.0 / 72.0) inches.

The value specified in **ShdwOffsetX** is used whenever a shape specifies a page default shadow.

To set the behavior for an individual shape, use the ShapeShdwOffsetX cell.

2.4.4.326 ShdwOffsetY

The **ShdwOffsetY** cell is a vLength custom token grouping that specifies the vertical offset of a page sheet's shadow effect set. The value is defined in relation to the coordinate system of the shape. It MUST have a PageSheet_Type parent element.

The shadow distance specified by this cell and the ShdwOffsetX cell MUST be greater than or equal to zero, and equal or less than (200.0 / 72.0) inches.

The value specified in **ShdwOffsetY** is used whenever a shape specifies a page default shadow.

To set the behavior for an individual shape, use the ShapeShdwOffsetY cell.

2.4.4.327 ShdwPattern

The **ShdwPattern** cell is a PtgByte parse token that specifies whether the shadow effect set is visible or not for a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be equal to zero or one. The value of one specifies that the shadow is visible, while the value of zero specifies that the shadow is not visible.

2.4.4.328 ShdwScaleFactor

The **ShdwScaleFactor** cell is a vScalar custom token grouping that specifies the normalized percentage by which the shadow effect set of a page sheet can be enlarged or reduced. It MUST have a PageSheet_Type parent element.

The value of the structure MUST be expressed as a percentage. It MUST be greater than or equal to zero, and less than or equal to two. The value is normalized such that a value of 2 corresponds to 200 percent. As the value of the structure increases, the shadow is enlarged.

The value specified in **ShdwScaleFactor** is used whenever a shape specifies a page default shadow.

To set the behavior for an individual shape, use the ShapeShdwScaleFactor cell.

2.4.4.329 ShdwType

The **ShdwType** cell is a PtgByte parse token that specifies the shadow effect set type of a page sheet. It MUST have a PageSheet_Type parent element.

The value specified in this cell is used whenever a shape specifies a page default shadow.

The value of the structure MUST be from the following table.

Value	Meaning
1	Specifies a simple shadow.
2	Specifies an oblique shadow.
3	Specifies an inner shadow.

To set the behavior for an individual shape, use the ShapeShdwType cell.

2.4.4.330 Size

The **Size** cell is a vLength custom token grouping that specifies the font size of a text run. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

2.4.4.331 SketchAmount

The **SketchAmount** cell is a PtgInt parse token that specifies the number of points, distributed uniformly across each path segment of a shape, where perturbations are performed for sketch effect. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

This cell MUST have a value greater than or equal to zero, and less than or equal to 25, with a default value of five.

2.4.4.332 SketchEnabled

The **SketchEnabled** cell is a PtgBool parse token that specifies whether sketch effect is applied to a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of zero specifies that the sibling cells SketchAmount, SketchFillChange, SketchLineChange, SketchLineWeight, and SketchSeed MUST be ignored. The default value is zero.

2.4.4.333 SketchFillChange

The **SketchFillChange** cell is a vScalar custom token grouping that specifies the amplitude of the fill perturbations for sketch effect. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage. It MUST be greater than or equal to zero, and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies no perturbation to the fill; a value of one specifies the maximum perturbation to the fill.

2.4.4.334 SketchLineChange

The **SketchLineChange** cell is a vScalar custom token grouping that specifies the amplitude of the path perturbations for sketch effect. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage. It MUST be greater than or equal to zero, and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100 percent. A value of zero specifies no perturbation to the path; a value of one specifies the maximum perturbation to the path.

2.4.4.335 SketchLineWeight

The **SketchLineWeight** cell is a vLength custom token grouping that specifies the additional line thickness to add to each path segment for sketch effect. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value MUST be greater than or equal to zero points, and less than or equal to 50 points. Its default value is three points.

2.4.4.336 SketchSeed

The **SketchSeed** cell is a PtgInt parse token that specifies a seed assigned to a shape for sketch effect. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value MUST have a value greater than or equal to zero, and less than or equal to 65535, with a default value of zero.

2.4.4.337 Snap

The **Snap** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Layer Section_Type parent element.

2.4.4.338 SoftEdgesSize

The **SoftEdgesSize** cell is a vLength custom token grouping that specifies the size of the soft edge in the effect properties of a shape. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to zero, and less than or equal to (100.0 / 72.0) inches. The value of zero specifies the shape doesn't have soft edges.

2.4.4.339 SortKey

The **SortKey** cell is a PtgString parse token that is used during formula evaluation only. It MUST have either a Row_Type parent element that has a Actions Section_Type parent element, a Row_Type parent element that has a Hyperlink Section_Type parent element, or a Row_Type parent element that has a Property Section_Type parent element.

2.4.4.340 SpAfter

The **SpAfter** cell is a vLength custom token grouping that specifies an amount of space inserted after each paragraph in the text block of a shape except for the last paragraph. Other paragraph properties can specify additional amounts of space. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

2.4.4.341 SpBefore

The **SpBefore** cell is a vLength custom token grouping that specifies an amount of space inserted before each paragraph in the text block of a shape except for the first paragraph. Other paragraph properties can specify additional amounts of space. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

2.4.4.342 SpLine

The **SpLine** cell is a vLength custom token grouping that specifies the height of a line of text in the text block of a shape. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

If the value of the structure is greater than or equal to zero, then the height of the line is equal to the value of the structure. If the value of the structure is less than zero, then the height of the line is equal to the absolute value of the structure multiplied by the largest font size of text in the line.

2.4.4.343 Status

The **Status** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Layer Section_Type parent element.

2.4.4.344 Strikethru

The **Strikethru** cell is a PtgBool parse token that specifies whether the text run has a strikethrough character property. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

A value of one specifies that the text run has a strikethrough; a value of zero specifies that the text run does not have a strikethrough.

2.4.4.345 Style

The **Style** cell is a PtgByte parse token that specifies character properties used to format a text run. It MUST have a Row_Type parent element that has a Character Section_Type parent element.

The value of the structure MUST be a bitwise OR combination of one or more of the values from the following table.

Value	Meaning
0x00	Specifies that the text run is unformatted.
0x01	Specifies that the text run has a bold character property.
0x02	Specifies that the text run has an italic character property.
0x04	Specifies that the text run has an underline character property.
0x08	Specifies that the text run has a small caps character property.

2.4.4.346 SubAddress

The **SubAddress** cell is a PtgString parse token that specifies a Uniform Resource Identifier (URI) **hyperlink location**. It MUST have a Row_Type parent element that has a Hyperlink Section_Type parent element.

The value of the sibling Address Cell_Type element determines the meaning of this value. If the value of Address Cell_Type element is empty, this value specifies the drawing page name and an optional shape name for the current web Drawing referenced by the hyperlink. If the value of the Address Cell_Type element is an **HTTP** URI, as described in [RFC2616], this value specifies the hyperlink location. Otherwise, this value is used during formula evaluation only.

If the value of Address Cell_Type element is empty, the value of the structure is defined by the following:

ABNF

```
value = page-name ["/" shape-name]
page-name = string-value
shape-name = string-value
```

Where string-value is an ABNF formula, specified in section 2.5.1.

2.4.4.347 TagName

The **TagName** cell is a PtgString parse token that is used during formula evaluation only. It MUST have a Row_Type parent element with an Actions Section_Type parent element or a Row_Type parent element with an ActionTag Section_Type parent element.

2.4.4.348 TextBkgnd

The **TextBkgnd** cell is a PtgColorRGB parse token that specifies the background color of a text block. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.349 TextBkgndTrans

The **TextBkgndTrans** cell is a vScalar custom token grouping that specifies the level of transparency of the background color of a text block. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage. It MUST be greater than or equal to zero, and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100

percent. A value of zero specifies that the background is completely opaque; a value of one specifies that the background is completely transparent.

2.4.4.350 TextDirection

The **TextDirection** cell is a vScalar custom token grouping that specifies the orientation of characters in a text block. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be zero or one. A value of zero specifies that the orientation is horizontal; a value of one specifies that the orientation is vertical.

2.4.4.351 TextPosAfterBullet

The **TextPosAfterBullet** cell is a vLength custom token grouping that specifies the distance from the beginning of the bullet and the text after the bullet in each bulleted item of a paragraph. It MUST have a Row_Type parent element that has a Paragraph Section_Type parent element.

If the value is less than or equal to zero, 0.25 inch is used as the distance from the beginning of the bullet and the text after the bullet; if the value is less than the width of the bullet, there is no distance between the bullet and the text after the bullet.

2.4.4.352 TheData

The **TheData** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.353 ThemeIndex

The **ThemeIndex** cell is a PtgInt parse token that specifies the index of the primary scheme dynamic theme component. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

The value of the **V** attribute MUST be greater than or equal to 0 and less than or equal to 65534.

If the value of the **V** attribute is equal to 65534, the index of the primary scheme dynamic theme component of a shape is specified by the **V** attribute of the **ThemeIndex** Cell_Type descendant element of the PageSheet_Type element containing the shape.

2.4.4.354 TheText

The **TheText** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.355 TopMargin

The **TopMargin** cell is a vLength custom token grouping that specifies the margin between the top border of a text block and the first line of text it contains. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

2.4.4.356 Transparency

The **Transparency** cell is a vScalar custom token grouping that specifies the level of transparency of an image. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be expressed as a percentage. It MUST be greater than or equal to zero, and less than or equal to one. The value is normalized such that a value of 1 corresponds to 100

percent. A value of zero specifies that the image is completely opaque; a value of one specifies that the image is completely transparent.

2.4.4.357 TxtAngle

The **TxtAngle** cell is a vAngle or vScalar custom token grouping that specifies the angle of rotation of a shape's text block. The value is defined in relation to the coordinate system of the shape. It MUST have a ShapeSheet_Type parent element. If **TxtAngle** is a vScalar, the value of the structure MUST be expressed as an angleInternalUnitNumber custom internal unit number. Increasing numbers indicate counterclockwise rotation.

2.4.4.358 TxtHeight

The **TxtHeight** cell is a vLength custom token grouping that specifies the height of a text block of a shape. It MUST have a ShapeSheet_Type parent element. It MUST be greater than or equal to zero.

2.4.4.359 TxtLocPinX

The **TxtLocPinX** cell is a vLength custom token grouping that specifies the x-coordinate of the center of rotation of a shape's text block. The value is defined in relation to the coordinate system of the text block. It MUST have a ShapeSheet_Type parent element.

2.4.4.360 TxtLocPinY

The **TxtLocPinY** cell is a vLength custom token grouping that specifies the y-coordinate of the center of rotation of a shape's text block. The value is defined in relation to the coordinate system of the text block. It MUST have a ShapeSheet_Type parent element.

2.4.4.361 TxtPinX

The **TxtPinX** cell is a vLength custom token grouping that specifies the x-coordinate of the center of rotation of a shape's text block. The value is defined in relation to the coordinate system of the shape. It MUST have a ShapeSheet_Type parent element.

2.4.4.362 TxtPinY

The **TxtPinY** cell is a vLength custom token grouping that specifies the y-coordinate of the center of rotation of a shape's text block. The value is defined in relation to the coordinate system of the shape. It MUST have a ShapeSheet_Type parent element.

2.4.4.363 TxtWidth

The **TxtWidth** cell is a vLength custom token grouping that specifies the width of a text block of a shape. It MUST have a ShapeSheet_Type parent element. It MUST be greater than or equal to zero.

2.4.4.364 Type

The **Type** cell is a vScalar custom token grouping that specifies the type of a Row_Type element. It MUST have a Row_Type parent element that has a Connection Section_Type, Field Section_Type, or Property Section_Type parent element.

If the cell is a descendant of a Property Section_Type parent element, it specifies a type of shape data stored in a sibling Value Cell_Type element. The value of the structure MUST be specified by a vDataType structure. If the cell is a descendant of a Field or Connection Section_Type parent element, it is used during formula evaluation only.

2.4.4.365 UICat

The **UICat** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Field Section_Type parent element.

2.4.4.366 UICod

The **UICod** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Field Section_Type parent element.

2.4.4.367 UIFmt

The **UIFmt** cell is a vScalar custom token grouping that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Field Section_Type parent element.

2.4.4.368 UIVisibility

The **UIVisibility** cell is a PtgBool parse token that specifies the visibility of a page. It MUST have a PageSheet_Type parent element.

A value of one specifies that the page is not visible; a value of zero specifies the page is visible.

2.4.4.369 UpdateAlignBox

The **UpdateAlignBox** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

2.4.4.370 UseGroupGradient

The **UseGroupGradient** cell is a PtgBool parse token that specifies whether a subshape uses the gradient starting and ending points of its parent shape for fill gradient. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

A value of one specifies that the subshape uses the gradient starting and ending points of its parent shape for fill gradient; a value of zero specifies that the subshape uses its own gradient starting and ending points for fill gradient. If the shape is not a subshape, the token is used during formula evaluation only.

2.4.4.371 Value

The **Value** cell is a vAny custom token grouping that specifies the value of a shape data, text field, or user-defined cell. It MUST have a Row_Type parent element that has either a Field Section_Type parent element, a Row_Type parent element that has a Property Section_Type parent element, or a Row_Type that has a User Section_Type parent element.

2.4.4.372 VariationColorIndex

The **VariationColorIndex** cell is a PtgInt parse token that specifies the index of the color scheme list of a dynamic theme variant. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the **V** attribute MUST be greater than or equal to 0 and less than or equal to 65534.

If the value of the **V** attribute is equal to 65534, the index into the color scheme list of a dynamic theme variant of a shape is specified by the **V** attribute of the **VariationColorIndex** Cell_Type descendant element of the PageSheet_Type element containing the shape.

The value of the **V** attribute specifies which of the four color scheme lists of a dynamic theme variant in a dynamic theme is used for evaluation in the QuickStyleLineColor, QuickStyleFillColor, QuickStyleShadowColor, and QuickStyleFontColor Cell_Type elements of a shape. The following table specifies which color scheme list of a dynamic theme variant is used.

Value of V attribute	Color scheme list of a dynamic theme variant
0	Color scheme list specified by the first CT_VariationClrScheme child element of a CT_VariationClrSchemeLst child element of a CT_OfficeStyleSheet element in a Theme_XML_Part.
1	Color scheme list specified by the second CT_VariationClrScheme child element of a CT_VariationClrSchemeLst child element of a CT_OfficeStyleSheet element in a Theme_XML_Part.
2	Color scheme list specified by third CT_VariationClrScheme child element of a CT_VariationClrSchemeLst child element of a CT_OfficeStyleSheet element in a Theme_XML_Part.
3	Color scheme list specified by the fourth CT_VariationClrScheme child element of a CT_VariationClrSchemeLst child element of a CT_OfficeStyleSheet element in a Theme_XML_Part.

2.4.4.373 VariationStyleIndex

The **VariationStyleIndex** cell is a PtgInt parse token that specifies the index of the style scheme list of a dynamic theme variant. It MUST have a ShapeSheet_Type or StyleSheet_Type parent element.

The value of the **V** attribute MUST be greater than or equal to 0 and less than or equal to 65534.

If the value of the **V** attribute is equal to 65534, the index into the color scheme list of a dynamic theme variant of a shape is specified by the **V** attribute of the **VariationStyleIndex** Cell_Type descendant element of the PageSheet_Type element containing the shape.

The value of the **V** attribute specifies which of the four style scheme lists of a dynamic theme variant in a dynamic theme is used for evaluation in the QuickStyleLineMatrix, QuickStyleFillMatrix, QuickStyleEffectsMatrix, and QuickStyleFontMatrix Cell_Type elements of a shape. The following table specifies which style scheme of a dynamic theme variant is used.

Value of V attribute	Style scheme list of a dynamic theme variant
0	Style scheme list specified by the first CT_VariationStyleScheme child element of a CT_VariationStyleSchemeLst child element of a CT_OfficeStyleSheet element in a Theme_XML_Part.
1	Style scheme list specified by the second CT_VariationStyleScheme child element of a CT_VariationStyleSchemeLst child element of a CT_OfficeStyleSheet element in a Theme_XML_Part.
2	Style scheme list specified by the third CT_VariationStyleScheme child element of a CT_VariationStyleSchemeLst child element of a CT_OfficeStyleSheet element in a

Value of V attribute	Style scheme list of a dynamic theme variant
	Theme_XML_Part.
3	Style scheme list specified by the fourth CT_VariationStyleScheme child element of a CT_VariationStyleSchemeLst child element of a CT_OfficeStyleSheet element in a Theme_XML_Part.

2.4.4.374 Verify

The **Verify** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Property Section_Type parent element.

2.4.4.375 VerticalAlign

The **VerticalAlign** cell is a vScalar custom token grouping that specifies the vertical alignment of text within a text block. It MUST have a ShapeSheet_Type or a StyleSheet_Type parent element.

The value of the structure MUST be greater than or equal to zero, and less than or equal to 255.

The value of the structure MUST be from the following table.

Value	Meaning
0	Top
1	Middle
2	Bottom
3, or greater	Top

2.4.4.376 ViewMarkup

The **ViewMarkup** cell is a PtgBool parse token that is used during formula evaluation only. It MUST have a DocumentSheet_Type parent element.

2.4.4.377 Visible

The **Visible** cell is a PtgBool parse token that specifies whether the shapes in a layer are visible. It MUST have a Row_Type parent element that has a Layer Section_Type parent element.

A value of one specifies that the shapes in this layer are visible; a value of zero specifies the shapes in this layer are hidden.

2.4.4.378 WalkPreference

The **WalkPreference** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a ShapeSheet_Type parent element or a StyleSheet_Type parent element.

2.4.4.379 Width

The **Width** cell is a vLength custom token grouping that specifies the width of the shape. It MUST have a ShapeSheet_Type parent element.

2.4.4.380 X

The **X** cell is a vLength or vScalar custom token grouping that specifies the x-coordinate of a point in the coordinate system of a shape. It MUST have a Row_Type parent element that has a Geometry Section_Type, Control Section_Type, Scratch Section_Type, ActionTag Section_Type, or Connection Section_Type parent element.

If it is a child element of a Row_Type element that has a Geometry Section_Type parent, it specifies the x-coordinate of a vertex in a geometry path. If it has an ArcTo, Ellipse, EllipticalArcTo, InfiniteLine, LineTo, MoveTo, NURBSTo, PolyLineTo, SplineKnot, or SplineStart Row_Type parent element, the cell is a vLength. If it has a RelCubBezTo, RelEllipticalArcTo, RelLineTo, RelMoveTo, or RelQuadBezTo parent element, the cell is a vScalar.

If it is a child element of a Row_Type element that has a Control, Scratch, ActionTag, or Connection Section_Type parent, **X** is a vLength that is used during formula evaluation only.

2.4.4.381 XCon

The **XCon** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a Row_Type parent element with a Control Section_Type parent element.

2.4.4.382 XDyn

The **XDyn** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Control Section_Type parent element.

2.4.4.383 XGridDensity

The **XGridDensity** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.384 XGridOrigin

The **XGridOrigin** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.385 XGridSpacing

The **XGridSpacing** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.386 XJustify

The **XJustify** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has an ActionTag Section_Type parent element.

2.4.4.387 XRulerDensity

The **XRulerDensity** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.388 XRulerOrigin

The **XRulerOrigin** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.389 Y

The **Y** cell is a vLength or vScalar custom token grouping that specifies the y-coordinate of a point in the coordinate system of a shape. It MUST have a Row_Type parent element that has a Geometry Section_Type, Control Section_Type, Scratch Section_Type, ActionTag Section_Type, or Connection Section_Type parent element.

If it is a child element of a Row_Type element that has a Geometry Section_Type parent, the cell specifies the y-coordinate of a vertex in a geometry path. If it has an ArcTo, Ellipse, EllipticalArcTo, InfiniteLine, LineTo, MoveTo, NURBSTo, PolyLineTo, SplineKnot, or SplineStart Row_Type parent element, the cell is a vLength. If it has a RelCubBezTo, RelEllipticalArcTo, RelLineTo, RelMoveTo, or RelQuadBezTo parent element, the cell is a vScalar.

If Y is a child element of a Row_Type element that has a Control, Scratch, ActionTag, or Connection Section_Type parent, it is a vLength that is used during formula evaluation only.

2.4.4.390 YCon

The **YCon** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Control Section_Type parent element.

2.4.4.391 YDyn

The **YDyn** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a Row_Type parent element that has a Control Section_Type parent element.

2.4.4.392 YGridDensity

The **YGridDensity** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.393 YGridOrigin

The **YGridOrigin** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.394 YGridSpacing

The **YGridSpacing** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.395 YJustify

The **YJustify** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a Row_Type parent element that has an ActionTag Section_Type parent element.

2.4.4.396 YRulerDensity

The **YRulerDensity** cell is a PtgByte parse token that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.4.397 YRulerOrigin

The **YRulerOrigin** cell is a vLength custom token grouping that is used during formula evaluation only. It MUST have a PageSheet_Type parent element.

2.4.5 Triggers

The **N** attribute of a Trigger_Type element MUST be equal to one of the values defined in the following sections. The usage of the trigger is specified in the corresponding section.

2.4.5.1 CategoryChanged

The **CategoryChanged** trigger is unused and MUST be ignored. It MUST have a ShapeSheet_Type parent element.

2.4.5.2 Path

The **Path** trigger is unused and MUST be ignored. It MUST have a Geometry Section_Type parent element.

2.4.5.3 RecalcBkgPageName

The **RecalcBkgPageName** trigger is unused and MUST be ignored. It MUST have a PageSheet_Type parent element.

2.4.5.4 RecalcColor

The **RecalcColor** trigger is unused and MUST be ignored. It MUST have a PageSheet_Type parent element.

2.4.5.5 RecalcCreateDT

The **RecalcCreateDT** trigger is unused and MUST be ignored. It MUST have a DocumentSheet_Type parent element.

2.4.5.6 RecalcData1

The **RecalcData1** trigger is unused and MUST be ignored. It MUST have a ShapeSheet_Type parent element.

2.4.5.7 RecalcData2

The **RecalcData2** trigger is unused and MUST be ignored. It MUST have a ShapeSheet_Type parent element.

2.4.5.8 RecalcData3

The **RecalcData3** trigger is unused and MUST be ignored. It MUST have a ShapeSheet_Type parent element.

2.4.5.9 RecalcEditDT

The **RecalcEditDT** trigger specifies that the DocLastEdit function token update trigger exists on the drawing page specified by its RefBy_Type child element. It MUST have a DocumentSheet_Type parent element.

A Trigger_Type element with an **N** attribute value of **RecalcEditDT** MUST exist if the web drawing contains a DocLastEdit function token, and a RefBy_Type child element MUST exist for every drawing page in the web drawing that has a DocLastEdit function token.

2.4.5.10 RecalcID

The **RecalcID** trigger is unused and MUST be ignored. It MUST have a ShapeSheet_Type parent element.

2.4.5.11 RecalcMasterName

The **RecalcMasterName** trigger is unused and MUST be ignored. It MUST have a ShapeSheet_Type parent element.

2.4.5.12 RecalcName

The **RecalcName** trigger is unused and MUST be ignored. It MUST have a ShapeSheet_Type parent element.

2.4.5.13 RecalcNowAndRand

The **RecalcNowAndRand** trigger specifies that the Now function token update trigger exists on the drawing page specified by its RefBy_Type child element. It MUST have a PageSheet_Type parent element.

A Trigger_Type element with an **N** attribute value of **RecalcNowAndRand** MUST be the child of every PageSheet_Type element that specifies a drawing page that contains a Now function token.

2.4.5.14 RecalcPageCount

The **RecalcPageCount** trigger is unused and MUST be ignored. It MUST have a DocumentSheet_Type parent element.

2.4.5.15 RecalcPageName

The **RecalcPageName** trigger is unused and MUST be ignored. It MUST have a ShapeSheet_Type or PageSheet_Type parent element.

2.4.5.16 RecalcPageNum

The **RecalcPageNum** trigger is unused and MUST be ignored. It MUST have a PageSheet_Type parent element.

2.4.5.17 RecalcPath

The **RecalcPath** trigger specifies that the Directory function token update trigger exists on the drawing page specified by its RefBy_Type child element. It MUST have a DocumentSheet_Type parent element.

A Trigger_Type element with an **N** attribute value of **RecalcPath** MUST exist if the web drawing contains a Directory function token, and a RefBy_Type child element MUST exist for every drawing page in the web drawing that has a DocLastSave function token.

2.4.5.18 RecalcPrintDT

The **RecalcPrintDT** trigger is unused and MUST be ignored. It MUST have a DocumentSheet_TypeDocumentSheet_Type (section 2.3.4.2.42) parent element.

2.4.5.19 RecalcSaveDT

The **RecalcSaveDT** trigger specifies that the DocLastSave function token update trigger exists on the drawing page specified by its RefBy_Type child element. It MUST have a DocumentSheet_Type parent element.

A Trigger_Type element with an **N** attribute value of **RecalcSaveDT** MUST exist if the web drawing contains a DocLastSave function token, and a RefBy_Type child element MUST exist for every drawing page in the web drawing that has a DocLastSave function token.

2.4.5.20 RecalcSummary

The **RecalcSummary** trigger specifies that at least one of the Category, Creator, Description, Keywords, Subject, or Title function token update triggers exists on the drawing page specified by its RefBy_Type child element. It MUST have a DocumentSheet_Type parent element.

A Trigger_Type element with an **N** attribute value of **RecalcSummary** MUST exist if the web drawing contains at least one of the Category, Creator, Description, Keywords, Subject, or Title function tokens, and a RefBy_Type child element MUST exist for every drawing page in the web drawing that has a Category, Creator, Description, Keywords, Subject, or Title function token.

2.4.5.21 RecalcType

The **RecalcType** trigger is unused and MUST be ignored. It MUST have a ShapeSheet_Type (section 2.3.4.2.88) parent element.

2.4.5.22 RelChanged

The **RelChanged** trigger is unused and MUST be ignored. It MUST have a **ShapeSheet_Type** (section 2.3.4.2.88) parent element.

2.4.5.23 ZOrderChanged

The **ZOrderChanged** trigger is unused and MUST be ignored. It MUST have a **PageSheet_Type** (section 2.3.4.2.68) parent element.

2.5 Formula Expressions and Evaluation

This section specifies the syntax and semantics for formula expressions.

2.5.1 Formula ABNF and Full Grammar Definition

A formula expression MUST conform to the following ABNF [RFC5234] grammar.

```
val = func / token / ref
val /= "(" val ")"      ; Enclose an expression in parens for higher precedence
func = Abs / ACos / Add / And / Ang2.5.3.5 / AngleToLoc / AngleToPar / ASin / ATan /
ATan2.5.3.9 / BitAnd / BitNot / BitOr / BitXor / BkgPageName / Blend / Bound / Cat / Category
/ Ceiling / CellIsThemed / Char / Company / Cos / CosH / Creator / CY / Date / DateTime /
DateValue / Day / DayOfYear / Deg / DependsOn / Description / Directory / Div / DocCreation /
DocLastEdit / DocLastPrint / DocLastSave / EEQ / EGE / EGT / ELE / ELT / ENE / FEQ / FGE /
FGT / FieldPicture / FileName / Find / FLE / Floor / FLT / FNE / Format / FormatEx /
FormulaExists / Gravity / Guard / HasCategory / Hour / HSL / Hue / HueDiff / HyperlinkBase /
ID / IF / IfError / Index / Int / IntersectX / IntersectY / Intup / Is2.5.3.77D / IsErr /
IsErrNA / IsError / IsErrValue / IsThemed / Keywords / Language / Left / Len / Ln / Loc /
LocalFormulaExists / LocToLoc / LocToPar / Log2.5.3.92 / Lookup / Lower / Lum / LumDiff /
Magnitude / Manager / MasterName / Max / Mid / Min / Minute / Modulus / Month / MsoShade /
MsoTint / Mul / NA / Name / Not / Now / Nurbs / Or / PageCount / PageName / PageNumber / Par
/ Pct / Pi / Pnt / Pntx / PntY / PolyLine / Pow / Rad / Rand / Ref / Replace / RGB / Right /
Round / Sat / SatDiff / Second / SetAtRef / SetAtRefEval / SetAtRefExpr / Shade / ShapeText /
Sign / Sin / SinH / Sqrt / StrSame / StrSameEx / Sub / Subject / Substitute / Sum / Tan /
TanH / TextHeight / TextWidth / Theme / ThemeCBV / ThemeGuard / ThemeProp / ThemeRestore /
ThemeVal / Time / TimeValue / Tint / Title / Tone / Trim / Trunc / UMinus / UniChar / UPlus /
Upper / Use / Version / WeekDay / Year / functiondef
functiondef = ALPHA *(ALPHA / DIGIT / "_") "(" [val *("," val)] ")"
token = PtgAcre / PtgAngDD / PtgAngDft / PtgAngDMS / PtgAngRad / PtgBool / PtgByte /
PtgColorRGB / PtgCy / PtgDate / PtgEDay / PtgEHour / PtgEMin / PtgErr / PtgESec / PtgEWeek /
PtgHectare / PtgInt / PtgNum / PtgNumCM / PtgNumDft / PtgNumF / PtgNumFI / PtgNumI / PtgNumKM
/ PtgNumM / PtgNumMI / PtgNumMM / PtgNumMultiDim / PtgNumNM / PtgNumPct / PtgNumYards /
PtgNurbs / PtgPageDft / PtgPnt / PtgPolyLine / PtgShort / PtgString / PtgTDurDft / PtgTypCD /
PtgTypCi / PtgTypDft / PtgTypDi / PtgTypPi / PtgTypPP / PtgTypPt / PtgUnsShort
ref           = [sheetref "!"] CellRef
sheetref       = CrossPageRef / DocSheetRef / MasterSheetRef / PageSheetRef / ShapeSheetRef /
StyleSheetRef
name          = 1*(ALPHA / DIGIT / " ")
nameid          = "'" name "'" / name ["." 1*DIGIT]
id          = 1*DIGIT
color-value        = "#" 6HEXDIG
dimension        = unsigned-int-value
int-value       = ["+" / "-"] 1*DIGIT
short-value       = ["+" / "-"] 1*DIGIT
unsigned-int-value     = ["+"] 1*DIGIT
unsigned-byte-value    = ["+"] 1*DIGIT
double-value        = ["+" / "-"] (*DIGIT ["."] 1*DIGIT / 1*DIGIT ".")
                      ["e" ["+" / "-"] 1*DIGIT]
bool-value        = ("true" / "1") / ("false" / "0")
string-value       = *utf8-char               ; [RFC3629] UTF-8 strings
utf8-char       = ascii-char / utf8-non-ascii-chars
ascii-char       = HTAB / LF / CR / SP / VCHAR   ; Whitespace and printing chars
utf8-non-ascii-chars = (%xC0-DF 1utf8-content) /
                       (%xE0-EF 2utf8-content) /
                       (%xF0-F7 3utf8-content) /
                       (%xF8-FB 4utf8-content) /
                       (%xFC-FD 5utf8-content)
utf8-content       = %x80-BF
```

2.5.2 Order of Operations

When evaluating a formula expression, a set of rules determines the order in which the different parts of the expression are evaluated. Rules with the lowest precedence numbers are evaluated first. Unless specified otherwise, expressions are evaluated from left to right.

Precedence	Operators	Meaning
1	( )	Parenthesized expressions are evaluated with the highest precedence in order of the most-nested to the least-nested expression, where most-nested is the expression enclosed in the most sets of matching parenthesis. In the following example, **val2** is the most-nested expression and would

Precedence	Operators	Meaning
		therefore be evaluated before **val1:** (val1 + (val2))
2	+ (unary), - (unary)	Identity and negation expressions.
3	^	Exponentiation expressions: Pow.
4	*, /	Multiplication and division expressions: Mul, Div.
5	+ (binary), - (binary)	Addition and subtraction expressions: Add, Sub.
6	&	String concatenation expressions: Cat.
7	<, _LT_, <=, _LE_, >, _GT_ >=, _GE_	Fuzzy and exact comparison expressions: FLT, ELT, FLE, ELE, FGT, EGT, FGE, EGE.
8	=, _EQ_, <>, _NE_,	Fuzzy and exact equality and inequality expressions: FEQ, EEQ, FNE, ENE.

2.5.3 Function Token Definitions

The following function token definition sections specify the function tokens that can be contained in a formula expression, in addition to those defined through sheet extensibility. The definition of each function specifies the function name, the type and sequence of expected arguments, and the type of token returned.

Information is also included about how the functions are evaluated. Unless a different reference context is explicitly specified, each function is evaluated in the reference context specified by the sheet containing the function's formula expression.

2.5.3.1 Abs

The **Abs** function performs an absolute value calculation.

ABNF:

```
Abs = "ABS(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: vNum

This function returns a vNum custom token grouping containing the absolute value of **Arg1**. The unit of the return value is equal to the unit of **Arg1**.

2.5.3.2 ACos

The **ACos** function performs an arccosine calculation.

ABNF:

```
ACos = "ACOS(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: PtgAngDft, PtgErr

This function returns a PtgAngDft parse token containing the arccosine of the value of **Arg1**. If **Arg1** is less than -1 or greater than one, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.3 Add

The **Add** function performs an addition calculation.

ABNF:

```
Add = "_ADD(" val "," val ")" / val "+" val
```

Required Arguments:

Name: **Arg1**

Type: vDoubleEx

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDoubleEx

An argument that specifies the second operand of the calculation.

Return Value:

Type: vNum, PtgCy, PtgErr

This function returns a vNum custom token grouping or a **PtgCy** parse token containing the sum of **Arg1** and **Arg2**. The type of the return value is calculated by the following algorithm.

```
SET returnType = PtgNum
SET returnDimension = 0
SET returnCurrencyID = 0
```

```
IF Arg1.Type = PtgCy AND Arg2.Type != PtgCy THEN
   SET returnType = PtgCy
   SET returnCurrencyID = currencyID of Arg1
ELSE IF Arg1.Type != PtgCy AND Arg2.Type = PtgCy THEN
   SET returnType = PtgCy
   SET returnCurrencyID = currencyID of Arg2
ELSE IF Arg1.Type = PtgCy AND Arg2.Type = PtgCy THEN
   SET returnType = PtgCy
   IF currencyID of Arg1 = currencyID of Arg2 OR currencyID of Arg2 = 1 THEN
      SET returnCurrencyID = currencyID of Arg1
   ELSE IF currencyID of Arg1 = 1 THEN
      SET returnCurrencyID = currencyID of Arg2
   ELSE
      SET returnType = PtgErr
      END IF
ELSE IF Arg1.Type = PtgDate THEN
   SET returnType = Arg1.Type
   SET returnDimension = 1
ELSE IF Arg2.Type= PtgDate THEN
   SET returnType = Arg2.Type
   SET returnDimension = 1
ELSE IF Arg1.Unit = PtgNumDft AND Arg2.Unit is a vUnitType THEN
   SET returnType = Arg2.Unit
   SET returnDimension = 1
ELSE IF Arg1.Unit is a vUnitType THEN
   SET returnType = Arg1.Unit
   IF Arg2.Dimension = 0 OR Arg2.Dimension = Arg1.Dimension THEN
      SET returnDimension = Arg1.Dimension
   ELSE
      SET returnDimension = 1
   END IF

   IF (returnType = PtgAcre OR returnType = PtgHectare) AND returnDimension != 2 THEN
      SET returnType = PtgNumDft
   END IF
ELSE IF Arg2.Unit is a vUnitType THEN
   SET returnType = Arg2.Unit
   IF Arg1.Dimension = 0 OR Arg2.Dimension = Arg1.Dimension THEN
      SET returnDimension = Arg2.Dimension
   ELSE
      SET returnDimension = 1
   END IF

   IF (returnType = PtgAcre OR returnType = PtgHectare) AND returnDimension != 2 THEN
      SET returnType = PtgNumDft
   END IF
END IF

IF returnType = PtgCy THEN
   Return type is PtgCy with currencyID = returnCurrencyID
ELSE IF returnType = PtgErr THEN
   Return type is PtgErr with error code of #VALUE!
ELSE IF returnDimension = 0 THEN
   Return type is PtgNum
ELSE IF returnDimension = 1 THEN
   Return type is returnType
ELSE
   Return type is PtgNumMultiDim with unit = returnType and dimension = returnDimension
END IF
```

2.5.3.4 And

The **And** function converts arguments to a Boolean value according to the conversion specified by vBoolean.

ABNF:

```
And = "AND(" val *( "," val ) ")"
```

Required Arguments:

Name: **Args**

Type: vBoolean

A set of arguments that specifies an operand.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **FALSE** if any of the arguments are equal to **FALSE**; otherwise, the value of **TRUE** is returned.

2.5.3.5 Ang360

The **Ang360** function normalizes an angle.

ABNF:

```
Ang360 = " ANG360(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: vAngle

This function returns a vAngle custom token grouping containing an angle equivalent to **Arg1**, normalized to be greater than or equal to zero and less than 2*pi, where the value of **Arg1** is assumed to have the unit of radians. If **Arg1** is a vAngle, the unit of the return value is equal to the unit of **Arg1**; otherwise, the function returns a PtgAngDft parse token.

2.5.3.6 AngleToLoc

The **AngleToLoc** function performs a transformation of an angle from the coordinate system of one shape into the coordinate system of another shape.

ABNF:

```
AngleToLoc = "ANGLETOLOC(" val "," val "," val ")"
```

Required Arguments:

Name: **Angle**

Type: vDouble

An argument that specifies an angle.

Name: **Source**

Type: CellRef

An argument that specifies a reference to a cell in a source shape.

Name: **Destination**

Type: CellRef

An argument that specifies a reference to a cell in a destination shape.

Return Value:

Type: vNum

This function returns a transformed angle in the coordinate system of the shape of **Destination** from **Angle** in a coordinate system of the shape of **Source**. If the **Source's** shape and the **Destination's** shape are not on the same drawing page, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.7 AngleToPar

The **AngleToPar** function performs a transformation of an angle from the coordinate system of one shape into the coordinate system of the parent of another shape.

ABNF:

```
AngleToPar = "ANGLETOPAR(" val "," val "," val ")"
```

Required Arguments:

Name: **Angle**

Type: vDouble

An argument that specifies an angle.

Name: **Source**

Type: CellRef

An argument that specifies a reference to a cell in a source shape.

Name: **Destination**

Type: CellRef

An argument that specifies a reference to a cell in a destination shape.

Return Value:

Type: vNum

This function returns a transformed angle in the coordinate system of the parent sheet of a shape of **Destination** from **Angle** in the coordinate system of shape of **Source**. If the **Source's** shape and the

Destination's shape are not on the same drawing page, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.8 ASin

The **ASin** function performs an arcsine calculation.

ABNF:

```
ASin = "ASIN(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: PtgAngDft, PtgErr

This function returns a PtgAngDft parse token containing the arcsine of the value of **Arg1**. If **Arg1** is less than -1 or greater than one, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.9 ATan2

The **ATan2** function calculates the angle between the positive x-axis and a vector.

ABNF:

```
ATan2 = "ATAN2(" val "," val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the y-component of the vector.

Name: **Arg2**

Type: vDouble

An argument that specifies the x-component of the vector.

Return Value:

Type: PtgAngDft

This function returns a PtgAngDft parse token containing the angle between the positive x-axis and the vector represented by **Arg1** and **Arg2**. The value is greater than -pi and less than or equal to pi. If **Arg1** is equal to zero and **Arg2** is equal to zero, the value is zero.

2.5.3.10 ATan

The **ATan** function performs an arctangent calculation.

ABNF:

```
ATan = "ATAN(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: PtgAngDft

This function returns a PtgAngDft parse token containing the arctangent of the value of **Arg1**.

2.5.3.11 BitAnd

The **BitAnd** function performs a bitwise AND operation.

ABNF:

```
BitAnd = "BITAND(" val "," val ")"
```

Required Arguments:

Name: **Arg1**

Type: vUnsignedInt

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vUnsignedInt

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the value of the bitwise AND operation between **Arg1** and **Arg2**.

2.5.3.12 BitNot

The **BitNot** function performs a bitwise NOT operation.

ABNF:

```
BitNot = "BITNOT(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vUnsignedInt

An argument that specifies the operand of the calculation.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the value of the bitwise NOT operation on **Arg1**.

2.5.3.13 BitOr

The **BitOr** function performs a bitwise OR operation.

ABNF:

```
BitOr = "BITOR(" val "," val ")"
```

Required Arguments:

Name: **Arg1**

Type: vUnsignedInt

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vUnsignedInt

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the value of the bitwise OR operation between **Arg1** and **Arg2**.

2.5.3.14 BitXor

The **BitXor** function performs a bitwise XOR operation.

ABNF:

```
BitXor = "BITXOR(" val "," val ")"
```

Required Arguments:

Name: **Arg1**

Type: vUnsignedInt

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vUnsignedInt

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the value of the bitwise XOR operation between **Arg1** and **Arg2**.

2.5.3.15 BkgPageName

The **BkgPageName** function returns the name of a background drawing page.

ABNF:

```
BkgPageName = [CrossPageRef "!"] "BKGPAGENAME(" [ val ] ")"
```

Optional Arguments:

Name: **Arg1**

Type: vSignedInt

An argument to specify the language of the return value.

Return Value:

Type: PtgString

This function returns the name of the background drawing page in the current reference context or reference context specified by CrossPageRef.

If the value of **Arg1** is 750, the **NameU** attribute of the Page_Type element associated with the background drawing page is returned.

If the value of **Arg1** is not 750, the **Name** attribute of the Page_Type element associated with the background drawing page is returned.

If the sheet does not have a background drawing page and the value of **Arg1** is 750, the string "<no background>" is returned.

If the sheet does not have a background drawing page and the value of **Arg1** is not 750, a language-dependent translation of the string "<no background>" is returned.

2.5.3.16 Blend

The **Blend** function performs a blend of two colors.

ABNF:

```
Blend = "BLEND(" val "," val "," val ")"
```

Required Arguments:

Name: **Color1**

Type: vColor

An argument that specifies the first color.

Name: **Color2**

Type: vColor

An argument that specifies the second color.

Name: **Fraction**

Type: vDouble

An argument that specifies the fractional amount of **Color2** in the blended color.

Return Value:

Type: PtgColorRGB, PtgErr

This function returns a PtgColorRGB parse token containing the blended color. If **Fraction** is less than zero or greater than one, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.17 Bound

The **Bound** function constrains a value by one or more ranges.

ABNF:

```
Bound = "BOUND(" val "," val 1*( "," val "," val "," val) ")"
```

Required Arguments:

Name: **Number**

Type: vDouble

An argument that specifies the value to be constrained.

Name: **BoundType**

Type: vUnsignedInt

An argument that specifies the bounding type. The valid values are described in the following table.

Value	Meaning
0	Constrain **Number** inclusive of any of the ranges specified.
1	Constrain **Number** exclusive of all the ranges specified.

2	Do not constrain **Number.**

Name: **IgnoreRange**

Type: vBoolean

An argument that specifies whether this range is included in constraining the **Number** or not.

Name: **ValueBeg**

Type: vDouble

An argument that specifies the beginning value of the range.

Name: **ValueEnd**

Type: vDouble

An argument that specifies the ending value of the range.

Optional Arguments:

Additional ranges MUST be specified using one or more additional groups of **IgnoreRange**, **ValueBeg**, and **ValueEnd**.

Name: **IgnoreRange**

Type: vBoolean

An argument that specifies if this range is not included in constraining the **Number**.

Name: **ValueBeg**

Type: vDouble

An argument that specifies the beginning value of the range.

Name: **ValueEnd**

Type: vDouble

An argument that specifies the ending value of the range.

Return Value:

Type: vNum, PtgErr

This function returns a vNum custom token grouping containing the constrained value of **Number,** as described in the following table.

Condition	Result
BoundType = 0	If **Number** is in at least one range with **IgnoreRange** equal to FALSE, the function returns **Number.** Otherwise, consider the set of all **ValueBeg** and **ValueEnd** values that belong to a range with **IgnoreRange** equal to FALSE. The function returns the value in this set that is closest to **Number**. If more than one value has the same minimum distance from

Condition	Result
	Number, the function returns the value in this set that has minimum distance from **Number** and that appears earliest in the argument list.
BoundType = 1	If **Number** is not in any range with **IgnoreRange** equal to FALSE, the function returns **Number**. Otherwise, consider the set of all **ValueBeg** and **ValueEnd** values that belong to a range with **IgnoreRange** equal to FALSE and that are not nested inside another range with **IgnoreRange** equal to FALSE. The function returns the value in this set that is closest to **Number**. If **Number** is in the exact middle of a **ValueBeg** value and a **ValueEnd** value in this set, the function returns the **ValueEnd** value.
BoundType = 2	The function returns **Number**.
Otherwise	The function returns a PtgErr parse token with an error code equal to #VALUE!.

If **Number** is a vAngle custom token grouping, the return value is normalized to be greater than or equal to zero and less than 2*pi. The unit of the return value is equal to the unit of **Number**. If **Number** is a PtgPnt or the wrong number of arguments is used, the function returns a PtgErr with an error code equal to #VALUE!.

2.5.3.18 Cat

The **Cat** function performs a concatenation of two strings.

ABNF:

```
Cat = "_CAT(" val "," val ")" / val "&" val
```

Required Arguments:

Name: **Arg1**

Type: vString

An argument that specifies the first string.

Name: **Arg2**

Type: vString

An argument that specifies the second string.

Return Value:

Type: PtgString

This function returns a PtgString parse token containing the concatenation of **Arg1** and **Arg2**.

2.5.3.19 Category

The **Category** function returns a Category property of a web drawing.

ABNF:

```
Category = "CATEGORY()"
```

Return Value:

Type: PtgString

This function returns the Category property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.20 Ceiling

The **Ceiling** function performs a ceiling calculation.

ABNF:

```
Ceiling = "CEILING(" val [ "," val ] ")"
```

Required Arguments:

Name: **Number**

Type: vDouble

An argument that specifies the value to be rounded.

Optional Arguments:

Name: **Multiple**

Type: vDouble

An argument that specifies the rounding increment. The default value is one if **Number** is greater than or equal to zero, and -1 if **Number** is less than zero.

Return Value:

Type: vNum, PtgErr

This function returns a vNum custom token grouping containing the next multiple of **Multiple** after **Number** that is further from zero. The unit of the return value is equal to the unit of **Number**.

If **Number** is a multiple of **Multiple**, the returned value is equal to **Number**.

If either **Number** or **Multiple** is equal to zero, the returned value is zero.

If **Number** and **Multiple** do not have the same sign, the function returns a PtgErr parse token with an error code equal to #NUM!.

If **Number** does not specify a vNum, the function returns a PtgErr with an error code equal to #VALUE!.

2.5.3.21 CellIsThemed

The **CellIsThemed** function returns a value of the passed argument.

ABNF:

```
CellIsThemed = "CELLISTHEMED(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vBoolean

An argument to be returned.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with the value of **Arg1**.

2.5.3.22 Char

The **Char** function performs a conversion from an integer to the corresponding character in the **American National Standards Institute (ANSI) character set**.

ABNF:

```
Char = "CHAR(" val ")"
```

Required Arguments:

Name: **Number**

Type: vUnsignedInt

An argument that specifies an integer.

Return Value:

Type: PtgString, PtgErr

This function returns a PtgString parse token containing the character that corresponds to **Number**. If **Number** is less than one or greater than 255, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.23 Company

The **Company** function returns a **Company** property of a web drawing.

ABNF:

```
Company = "COMPANY()"
```

Return Value:

Type: PtgString

This function returns the **Company** property, specified in [ISO/IEC29500-1:2011] section 22.2.2.5, from the App XML part of a web drawing.

2.5.3.24 Cos

The **Cos** function performs a cosine calculation.

ABNF:

```
Cos = "COS(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation. If **Arg1** is not a vAngle custom token grouping, the value of **Arg1** is assumed to have the unit of radians.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the cosine of the value of **Arg1**.

2.5.3.25 CosH

The **CosH** function performs the hyperbolic cosine calculation.

ABNF:

```
CosH = "COSH(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation. If **Arg1** is not a vAngle custom token grouping, the value of **Arg1** is assumed to have the unit of radians.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the hyperbolic cosine of **Arg1**.

2.5.3.26 Creator

The **Creator** function returns a Creator property of a web drawing.

ABNF:

```
Creator = "CREATOR()"
```

Return Value:

Type: PtgString

This function returns the **Creator** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.27 CY

The **CY** function returns a currency.

ABNF:

```
CY = "CY(" val [ "," val ] ")"
```

Required Arguments:

Name: **Value**

Type: vDouble

An argument that specifies the currency value.

Optional Arguments:

Name: **Cy**

Type: vCurrency

An argument that specifies the currency. It defaults to a value of -1, if missing.

Return Value:

Type: PtgCY, PtgErr

This function returns a PtgCY parse token containing **Value** and **Cy**. If **Cy** is not found in the table of vCurrency the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.28 Date

The **Date** function returns a date, according to the Gregorian calendar, from values representing a year, month, and day.

ABNF:

```
Date = "DATE(" val "," val "," val ")"
```

Required Arguments:

Name: **Year**

Type: vUnsignedInt

An argument that specifies a year. A value from zero through 29 (inclusive) corresponds to the range of years from 2000 through 2029 (inclusive). A value from 30 through 100 (inclusive) corresponds to the range of years from 1930 through 2000 (inclusive). A value greater than 100 corresponds to that year. It MUST be less than or equal to 9999.

Name: **Month**

Type: vSignedInt

An argument that specifies an offset in months from December first of the previous year.

Name: **Day**

Type: vSignedInt

An argument that specifies an offset in days from the last day of the previous month.

Return Value:

Type: PtgDate, PtgErr

This function returns a PtgDate parse token containing the date, according to the Gregorian calendar, represented by **Year**, **Month**, and **Day**. If the arguments specify a date before January 1st, 1899 or between January 1st, 1900 and November 30th, 1900 inclusively, the function returns a PtgErr parse token with an error code equal to #NUM!. If either **Month** or **Day** cannot be interpreted as a type of vSignedInt, the function returns a PtgErr with an error code equal to #VALUE!.

2.5.3.29 DateTime

The **DateTime** function converts a value to a date and time.

ABNF:

```
DateTime = "DATETIME(" val [ "," val ] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a value representing a date and time.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies a **language code identifier (LCID)** to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgDate, PtgErr

If **DateTimeArg** is a PtgString parse token, this function attempts to parse it using date and time format strings according to .NET globalization rules for **Locale**. For more information about .NET globalization rules, see [MSDN-ENCLOC]. If the string is successfully parsed, the function returns a PtgDate parse token containing the parsed date and time. If the string is not successfully parsed, the function returns a PtgErr parse token with an error code equal to #VALUE!.

If **DateTimeArg** is not a PtgString, the function returns a PtgDate containing the value of **DateTimeArg** interpreted as a type of vDouble. If **DateTimeArg** cannot be interpreted as a vDouble, the function returns a PtgErr with an error code equal to #VALUE!.

2.5.3.30 DateValue

The **DateValue** function returns a date component from a value representing a date and time.

ABNF:

```
DateValue = "DATEVALUE(" val ["," val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a value representing a date and time.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgDate, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion is successful, the function returns a PtgDate containing the date component of **DateTimeArg**. If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.31 Day

The **Day** function returns a day of the month, according to the Gregorian calendar, from a value representing a date and time.

ABNF:

```
Day = "DAY(" val ["," val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a date and time value.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgNum, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion is successful, the function returns a PtgNum parse token containing the day of the month component of **DateTimeArg.** If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.32 DayOfYear

The **DayOfYear** function returns a day of the year, according to the Gregorian calendar, from a value representing a date and time.

ABNF:

```
DayOfYear = "DAYOFYEAR(" val ["," val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a date and time value.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgNum, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion is successful, the function returns a PtgNum parse token containing the day of the year component from **DateTimeArg.** If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.33 Deg

The **Deg** function performs a conversion to an angle in degrees.

ABNF:

```
Deg = "DEG(" val ")"
```

Required Arguments:

Name: **Angle**

Type: vDouble

An argument that specifies an angle.

Return Value:

Type: PtgAngDD

This function returns a PtgAngDD parse token containing the value of **Angle**.

2.5.3.34 DependsOn

The **DependsOn** function returns a value of FALSE.

ABNF:

```
DependsOn = "DEPENDSON(" 1*( val ) ")"
```

Required Arguments:

Name: **Arg1**

Type: vAny

One or more arguments to be ignored.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of FALSE.

2.5.3.35 Description

The **Description** function returns the **Description** property of a web drawing.

ABNF:

```
Description = "DESCRIPTION()"
```

Return Value:

Type: PtgString

This function returns the **Description** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.36 Directory

The **Directory** function returns a path to a web drawing.

ABNF:

```
Directory = "DIRECTORY()"
```

Return Value:

Type: PtgString

This function returns the path to a web drawing.

2.5.3.37 Div

The **Div** function performs a division calculation.

ABNF:

```
Div = "_DIV(" val "," val ")" / val "/" val
```

Required Arguments:

Name: **Arg1**

Type: vDoubleEx

An argument that specifies the first operand of the division operation.

Name: **Arg2**

Type: vDoubleEx

An argument that specifies the second operand of the division operation.

Return Value:

Type: vNum, PtgCy, PtgErr

This function returns a **vNum** custom token grouping (section 2.5.7.4) or a **PtgCy** parse token (section 2.5.4.9) containing the result of **Arg1** divided by **Arg2**. The type of the return value is calculated by the following algorithm.

```
SET returnType = PtgNum
SET returnError = #VALUE!
SET returnDimension = 0
SET returnCurrencyID = 0

IF Arg2.Value = 0 THEN
   SET returnType = PtgErr
   SET returnError = #DIV/0
ELSE IF Arg1.Type = PtgCy AND Arg2.Type != PtgCy THEN
   SET returnType = PtgCy
   SET returnCurrencyID = currencyID of Arg1
ELSE IF Arg1.Type != PtgCy AND Arg2.Type = PtgCy THEN
   SET returnType = PtgCy
   SET returnCurrencyID = currencyID of Arg2
ELSE IF Arg1.Type = PtgCy AND Arg2.Type = PtgCy THEN
   SET returnType = PtgCy
   IF currencyID of Arg1 = currencyID of Arg2 OR currencyID of Arg2 = 1 THEN
      SET returnCurrencyID = currencyID of Arg1
   ELSE IF currencyID of Arg1 = 1 THEN
      SET returnCurrencyID = currencyID of Arg2
   ELSE
      SET returnType = PtgErr
      SET returnError = #VALUE!
```

```
   END IF
ELSE IF Arg1.Unit is a vUnitType AND Arg2.Unit is not a vUnitType THEN
   SET returnType = Arg1.Unit
   SET returnDimension = Arg1.Dimension
ELSE IF Arg1.Unit is not a vUnitType AND Arg2.Unit is a vUnitType THEN
   SET returnType = Arg2.Unit
   IF called via '_DIV' function rather than '/' operator THEN
      SET returnDimension = -Arg2.Dimension
   ELSE
      SET returnDimension = Arg2.Dimension
   END IF

ELSE
   IF Arg1.Unit = PtgNumPct AND Arg2.Unit = PtgNumPct THEN
      SET returnDimension = 1
   ELSE IF Arg1.Unit = PtgNumPct THEN
      SET returnDimension = -Arg2.Dimension
   ELSE IF Arg2.Unit = PtgNumPct THEN
      SET returnDimension = Arg1.Dimension
ELSE
   SET returnDimension = Arg1.Dimension - Arg2.Dimension
   IF Arg1.Unit is a vUnitType THEN
      SET returnType = Arg1.Unit
   ELSE
      SET returnType = Arg2.Unit
   END IF
END IF

IF returnType = PtgCy THEN
   Return type is PtgCy with currencyID = returnCurrencyID
ELSE IF returnType = PtgErr THEN
   Return type is PtgErr with error code of returnError
ELSE IF (returnType = PtgAcre OR returnType = PtgHectare) AND returnDimension != 2 THEN
   Return type is PtgNumMultiDim with unit = PtgNumDft and dimension = returnDimension
ELSE IF returnDimension = 0 THEN
   Return type is PtgNum
ELSE IF returnDimension = 1 THEN
   Return type is returnType
ELSE
   Return type is PtgNumMultiDim with unit = returnType and dimension = returnDimension
END IF
```

2.5.3.38 DocCreation

This function returns the DocCreation property of a Web drawing.

ABNF:

```
DocCreation = "DOCCREATION()"
```

Return Value:

Type: PtgDate

This function returns the **DocCreation** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.39 DocLastEdit

The **DocLastEdit** function returns a **Modified** property of a web drawing.

ABNF:

```
DocLastEdit = "DOCLASTEDIT()"
```

Return Value:

Type: PtgDate

This function returns the **Modified** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.40 DocLastPrint

The **DocLastPrint** function returns the **lastPrinted** property of a web drawing.

ABNF:

```
DocLastPrint = "DOCLASTPRINT()"
```

Return Value:

Type: PtgDate

This function returns the **lastPrinted** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.41 DocLastSave

The **DocLastSave** function returns a **Modified** property of a web drawing.

ABNF:

```
DocLastSave = "DOCLASTSAVE()"
```

Return Value:

Type: PtgDate

This function returns the **Modified** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.42 EEQ

The **EEQ** function calculates if **Arg1** is equal to **Arg2**.

ABNF:

```
EEQ = "_EEQ(" val "," val ")" / val "_EQ_" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if **Arg1** is equal to **Arg2**; otherwise, the value **FALSE** is returned.

2.5.3.43 EGE

The **EGE** function calculates if **Arg1** is greater than or equal to **Arg2**.

ABNF:

```
EGE = "_EGE(" val "," val ")" / val "_GE_" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if **Arg1** is greater than or equal to **Arg2**; otherwise, the value **FALSE** is returned.

2.5.3.44 EGT

The **EGT** function calculates if **Arg1** is greater than **Arg2**.

ABNF:

```
EGT = "_EGT(" val "," val ")" / val "_GT_" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if **Arg1** is greater than **Arg2**; otherwise, the value **FALSE** is returned.

2.5.3.45 ELE

The **ELE** function calculates if **Arg1** is less than or equal to **Arg2**.

ABNF:

```
ELE = "_ELE(" val "," val ")" / val "_LE_" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if **Arg1** is less than or equal to **Arg2**; otherwise, the value **FALSE** is returned.

2.5.3.46 ELT

The **ELT** function calculates if **Arg1** is less than **Arg2**.

ABNF:

```
ELT = "_ELT(" val "," val ")" / val "_LT_" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if **Arg1** is less than **Arg2**; otherwise, the value **FALSE** is returned.

2.5.3.47 ENE

The **ENE** function calculates if **Arg1** is not equal to **Arg2**.

ABNF:

```
ENE = "_ENE(" val "," val ")" / val "_NE_" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if **Arg1** is not equal to **Arg2**; otherwise, the value of **FALSE** is returned.

2.5.3.48 FEQ

The **FEQ** function calculates if the absolute value of the difference between **Arg1** and **Arg2** is less than or equal to 1E-9 (0.000000001).

ABNF:

```
FEQ = "_FEQ(" val "," val ")" / val "=" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if the absolute value of the difference between **Arg1** and **Arg2** is less than or equal to 1E-9; otherwise, the value **FALSE** is returned.

2.5.3.49 FGE

The **FGE** function calculates if the difference between **Arg1** and **Arg2** is greater than or equal to -1E-9 (-0.000000001).

ABNF:

```
FGE = "_FGE(" val "," val ")" / val ">=" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if the difference between **Arg1** and **Arg2** is greater than or equal to 1E-9; otherwise, the value **FALSE** is returned.

2.5.3.50 FGT

The **FGT** function calculates if **Arg1** is greater than **Arg2** by at least the amount of 1E-9 (0.000000001).

ABNF:

```
FGT = "_FGT(" val "," val ")" / val ">" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if **Arg1** is greater than **Arg2** by more than 1E-9 (0.000000001); otherwise, the value **FALSE** is returned.

2.5.3.51 FieldPicture

The **FieldPicture** function returns a format string.

ABNF:

```
FieldPicture = "FIELDPICTURE(" val ")"
```

Required Arguments:

Name: **Index**

Type: vFieldPicture

An argument that specifies a field picture index.

Return Value:

Type: PtgString

This function returns a field picture format string based on **Index** as specified in the vFieldPicture custom structure.

2.5.3.52 FileName

The **FileName** function returns a file name of a web drawing.

ABNF:

```
FileName = "FILENAME()"
```

Return Value:

Type: PtgString

This function returns the file name of a web drawing.

2.5.3.53 Find

The **Find** function returns the index of the first instance of a text string within another text string.

ABNF:

```
Find = "FIND(" val "," val [ "," [ val ] "," [ val ]] ")"
```

Required Arguments:

Name: **FindText**

Type: vString

An argument that specifies the string to be found.

Name: **WithinText**

Type: vString

An argument that specifies the string to search within.

Optional Arguments:

Name: **StartNum**

Type: vUnsignedInt

An argument that specifies the one-based position at which this function starts a search. The default value is one.

Name: **IgnoreCase**

Type: vBoolean

An argument that specifies whether the search is case insensitive. A value of **TRUE** specifies that case is ignored. The default value is **FALSE**.

Return Value:

Type: PtgInt, PtgErr

This function returns a PtgInt parse token containing the one-based position in **WithinText** at which **FindText** is found.

If **FindText** is empty and **StartNum** is less than or equal to the number of characters in **WithinText**, the function returns a PtgInt containing the value of **StartNum**.

If **FindText** is empty and **StartNum** is greater than the number of characters in **WithinText**, the function returns a PtgInt containing the value of the number of characters in **WithinText** + 1.

If **FindText** is not found or if **StartNum** is equal to zero or greater than the number of characters in **WithinText**, the function returns a PtgErr parse token with an error code equal to #VALUE!.

The search is performed according to .NET globalization rules based on the value of the **Language** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing. For more information about .NET globalization rules, see [MSDN-ENCLOC].

2.5.3.54 FLE

The **FLE** function calculates if the difference between **Arg1** and **Arg2** is less than or equal to 1E-9 (0.000000001).

ABNF:

```
FLE = "_FLE(" val "," val ")" / val "<=" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if the difference between **Arg1** and **Arg2** is less than or equal to 1E-9; otherwise, the value FALSE is returned.

2.5.3.55 Floor

The **Floor** function performs a floor calculation.

ABNF:

```
Floor = "FLOOR(" val [ "," val ] ")"
```

Required Arguments:

Name: **Number**

Type: vDouble

An argument that specifies the value to be rounded.

Optional Arguments:

Name: **Multiple**

Type: vDouble

An argument that specifies the rounding increment. The default value is one if **Number** is greater than or equal to zero, and -1 if **Number** is less than zero.

Return Value:

Type: vNum, PtgErr

This function returns a vNum custom token grouping containing the previous multiple of **Multiple** before **Number** that is closer to zero. The unit of the return value is equal to the unit of **Number**.

If **Number** is a multiple of **Multiple**, the returned value is equal to **Number**.

If either **Number** or **Multiple** is equal to zero, the returned value is zero.

If **Number** and **Multiple** do not have the same sign, the function returns a PtgErr parse token with an error code equal to #NUM!.

If **Number** does not specify a vNum, the function returns a PtgErr with an error code equal to #VALUE!.

2.5.3.56 FLT

The **FLT** function calculates if **Arg1** is less than **Arg2** by at least the amount of 1E-9 (0.000000001).

ABNF:

```
FLT = "_FLT(" val "," val ")" / val "<" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if **Arg1** is less than **Arg2** by more than 1E-9 (0.000000001); otherwise, the value FALSE is returned.

2.5.3.57 FNE

The **FNE** function calculates if **Arg1** differs from **Arg2** by at least the amount of 1E-9 (0.000000001).

ABNF:

```
FNE = "_FNE(" val "," val ")" / val "<>" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDouble

An argument that specifies the second operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if **Arg1** differs from **Arg2** by more than 1E-9 (0.000000001); otherwise, the value FALSE is returned.

2.5.3.58 Format

The **Format** function formats a value as a string using the specified format string.

ABNF:

```
Format = "FORMAT(" val "," val ")"
```

Required Arguments:

Name: **Value**

Type: vAny

An argument that specifies the value to be formatted.

Name: **FormatString**

Type: vFormatString

An argument that specifies the formatting information used to format **Value**.

Return Value:

Type: PtgString

This function returns a PtgString parse token containing **Value** formatted as a string with **FormatString.** The formatting is performed as described in the custom structure vFormatString.

2.5.3.59 FormatEx

The **FormatEx** function formats a value as a string.

ABNF:

```
FormatEx = "FORMATEX(" val "," val [ "," [ val ] [ "," [ val ] [ "," [ val ] [ "," [ val
]]]]] ")"
```

Required Arguments:

Name: **Value**

Type: vAny

An argument that specifies the value to be formatted.

Name: **FormatString**

Type: vFormatString

An argument that specifies the formatting information used to format **Value**.

Optional Arguments:

Name: **SrcUnit**

Type: vUnitString

An argument that specifies the unit of **Value**.

Name: **DstUnit**

Type: PtgString

An argument that specifies the unit displayed in the resulting string. It MUST be a vUnitString custom structure or a value equal to "NOCAST".

Name: **LangID**

Type: vLanguageID

An argument that specifies an LCID used to format **Value**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Name: **CalID**

Type: vCalendar

An argument that specifies a calendar system used to format **Value**. If the argument is omitted the value of zero is used.

Return Value:

Type: PtgString

This function returns a PtgString parse token containing **Value** with the unit specified by **SrcUnit**, converted into the unit specified by **DstUnit**, formatted as a string as specified by **FormatString**, **LangID**, and **CalID**. The formatting is performed as described in the custom structure vFormatString.

Value is converted into the unit specified by **DstUnit** as follows. If **DstUnit** contains the value of "NOCAST", the conversion is not performed. If **Value** is a vScalar custom token grouping, **Value** is multiplied by a factor that converts the unit of **SrcUnit** into the custom internal unit type associated with the type of **ScrUnit**. The resulting value, or **Value** if **Value** is a vUnitType, is multiplied by a factor that converts the custom internal unit type associated with the type of **DstUnit** into the unit of **DstUnit.**

2.5.3.60 FormulaExists

The **FormulaExists** function returns whether a cell contains a formula.

ABNF:

```
FormulaExists = "FORMULAEXISTS(" val ")"
```

Required Arguments:

Name: Arg1

Type: CellRef

An argument that specifies a reference to a cell in a shape.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with the value of **TRUE** if the cell contains a formula. Otherwise, it returns a PtgBool with the value of **FALSE**.

2.5.3.61 Gravity

The **Gravity** function returns an angle.

ABNF:

```
Gravity = "GRAVITY(" val [ "," val [ "," val ]] ")"
```

Required Arguments:

Name: **Angle**

Type: vDouble

An argument that specifies an angle. It is normalized to be greater than or equal to zero and less than 2*pi. If **Angle** is not a vAngle custom token grouping, the value of **Angle** is assumed to have the unit of degrees.

Optional Arguments:

Name: **Limit1**

Type: vDouble

An argument that specifies the first limit of rotation. It is normalized to be greater than or equal to zero and less than 2*pi. It defaults to a value of 90 degrees.

Name: **Limit2**

Type: vDouble

An argument that specifies the second limit of rotation. It is normalized to be greater than or equal to zero and less than 2*pi. It defaults to a value of 270 degrees.

Return Value:

Type: vAngle

This function returns a vAngle with a value of 180 degrees if **Angle** is between the value specified by **Limit1** and **Limit2**. Otherwise, the function returns a vAngle with a value of zero degrees.

2.5.3.62 Guard

The **Guard** function returns a value of the passed argument.

ABNF:

```
Guard = "GUARD(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vAny

An argument to be returned.

Return Value:

Type: vAny

This function returns the value of **Arg1**.

2.5.3.63 HasCategory

The **HasCategory** function returns whether there is any intersection of the category list of the shape and the category list of the operand. The category list of a shape is specified by the value its msvShapeCategories user row.

ABNF:

```
HasCategory = "HASCATEGORY(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vString

An argument that specifies a semicolon-delimited list of category names. This category list has the same form as the category list described in msvShapeCategories.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with the value of **TRUE** if the shape's category list contains any of the categories specified by **Arg1**'s category list. If either the shape or **Arg1** has an empty category list, this function returns a **PtgBool** with the value of **FALSE**.

2.5.3.64 Hour

The **Hour** function returns an hour from a value representing a date and time.

ABNF:

```
Hour = "HOUR(" val ["," val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a date and time value.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgNum, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion is successful, the function returns a PtgNum parse token containing the hour component from **DateTimeArg.** If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.65 HSL

The **HSL** function transforms a color specified in the hue-saturation-luminance (HSL) color space into a color specified in the RGB color space. **HSL** component values are in the range zero to 240, inclusive.

ABNF:

```
HSL = "HSL(" val "," val "," val ")"
```

Required Arguments:

Name: **Hue**

Type: vUnsignedInt

An argument that specifies the hue component.

Name: **Saturation**

Type: vUnsignedInt

An argument that specifies the saturation component.

Name: **Luminance**

Type: vUnsignedInt

An argument that specifies the luminance component.

Return Value:

Type: PtgColorRGB

This function returns a PtgColorRGB parse token containing the color value specified by **Hue**, **Saturation** and **Luminance**. If **Hue** is greater than 240 and less than or equal to 0xFF, this function sets the value to 240 and performs the operation. If the value of any of the arguments is greater than 0xFF, the function sets the argument's value to the result of the bitwise AND operation between the original value and 0xFF, before performing the operation.

2.5.3.66 Hue

The **Hue** function calculates the hue component, as specified in the hue-saturation-luminance (HSL) color space, of a color value.

ABNF:

```
Hue = "HUE(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vColor

An argument that specifies the color.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the hue of **Arg1**. The value is less than or equal to 240.

2.5.3.67 HueDiff

The **HueDiff** function calculates the difference in hue, as specified in the hue-saturation-luminance (HSL) color space, between two color values.

ABNF:

```
HueDiff = "HUEDIFF(" val "," val ")"
```

Required Arguments:

Name: **Arg1**

Type: vColor

An argument that specifies the first color.

Name: **Arg2**

Type: vColor

An argument that specifies the second color.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing a value that is equal to the hue of **Arg1** minus the hue of **Arg2**. This value is greater than or equal to -240 and less than or equal to 240.

2.5.3.68 HyperlinkBase

The **HyperlinkBase** function returns the **HyperlinkBase** property of a web drawing.

ABNF:

```
HyperlinkBase = "HYPERLINKBASE()"
```

Return Value:

Type: PtgString

This function returns the **HyperlinkBase** property, specified in [ISO/IEC29500-1:2011] section 22.2.2.11, from the App XML part of a web drawing.

2.5.3.69 ID

The **ID** function returns a value of the identifier of a shape.

ABNF:

```
ID = [ShapeSheetRef "!"] "ID()"
```

Return Value:

Type: PtgInt

This function returns a PtgInt parse token containing the **ID** attribute of the ShapeSheet_Type element associated with the shape sheet of a shape in the current reference context or reference context specified by ShapeSheetRef.

2.5.3.70 IF

The **IF** function returns the value of one of its operands based on the value of its first operand.

ABNF:

```
If = "IF(" val "," [ val ] "," [ val ] ")"
```

Required Arguments:

Name: **Arg1**

Type: vBoolean

An argument that specifies which argument to return based on its value.

Name: **Arg2**

Type: vAny

An argument that defaults to a PtgBool parse token with the value of **TRUE** if missing.

Name: **Arg3**

Type: vAny

An argument that defaults to a PtgBool with the value of **FALSE** if missing.

Return Value:

Type: vAny

This function returns one of its arguments according to the following table.

Arg1 Value	Return Value
TRUE	**Arg2**
FALSE	**Arg3**
Error	**Arg1**

2.5.3.71 IfError

The **IfError** function returns the value of the first operand if it does not evaluate to an error; otherwise, the value of the second operand is returned.

ABNF:

```
IfError = "IFERROR(" val "," [ val ] ")"
```

Required Arguments:

Name: **Arg1**

Type: vAny

An argument.

Name: **Arg2**

Type: vAny

An argument that defaults to a PtgBool with the value of **FALSE** if missing.

Return Value:

Type: vAny

This function returns either **Arg1** if it does not evaluate to an error or **Arg2** otherwise.

2.5.3.72 Index

The **Index** function returns a substring from a delimited text string.

ABNF:

```
Index = "INDEX(" val "," val [ "," [ val ] "," [ val ]] ")"
```

Required Arguments:

Name: **IndexPosition**

Type: vSignedInt

An argument that specifies a zero-based index into **List**.

Name: **List**

Type: vString

An argument that specifies a string consisting of a list (2) of substrings delimited by **Delimiter**. Consecutive delimiters or delimiters at the beginning and/or end of **List** specify empty substrings.

Optional Arguments:

Name: **Delimiter**

Type: vString

An argument that specifies the delimiter that separates substrings inside **List** with a default value of ";", if missing.

Name: **ErrorValue**

Type: vString, PtgErr

An argument that specifies a string or error to be returned if **Index** is out of range. The default value is an empty string.

Return Value:

Type: PtgString, PtgErr

This function returns a PtgString parse token equal to the substring at the zero-based index location **IndexPosition** in **List**, which is delimited by **Delimiter**.

If **IndexPosition** is less than zero or greater than or equal to the number of items in **List** and **ErrorValue** is of the parse token type PtgErr, the function returns **ErrorValue** as a PtgErr.

If **IndexPosition** is less than zero or greater than or equal to the number of items in **List** and **ErrorValue** is of custom structure type vString, the function returns a PtgString token that specifies **ErrorValue**.

2.5.3.73 Int

The **Int** function performs a rounding calculation down to the next lesser integer value.

ABNF:

```
Int = "INT(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the value to be rounded.

Return Value:

Type: vNum

This function returns a custom token grouping of the type vNum containing the next lesser integer value if **Arg1** is not an integer. The unit of the return value is equal to the unit of **Arg1.**

2.5.3.74 IntersectX

The **IntersectX** function calculates the x-coordinate of the point where two lines intersect. Each line is defined by a point on the line and an angle.

ABNF:

```
IntersectX = "INTERSECTX(" val "," val "," val "," val "," val "," val ")"
```

Required Arguments:

Name: **X1**

Type: vDouble

An argument that specifies the x-coordinate of a point defining the first line.

Name: **Y1**

Type: vDouble

An argument that specifies the y-coordinate of a point defining the first line.

Name: **Angle1**

Type: vDouble

An argument that specifies an angle of the first line relative to the positive x-axis.

Name: **X2**

Type: vDouble

An argument that specifies the x-coordinate of a point defining the second line.

Name: **Y2**

Type: vDouble

An argument that specifies the y-coordinate of a point defining the second line.

Name: **Angle2**

Type: vDouble

An argument that specifies the angle of a second line relative to the positive x-axis.

Return Value:

Type: vNum, PtgErr

This function returns a custom token grouping of the type vNum containing the x-coordinate of the point where two lines intersect. If any of the arguments **X1**, **Y1**, **X2**, **Y2** are a custom token grouping of the type vUnitType, the unit of the return value is equal to the unit of the first of those arguments that is a vUnitType. If none of those arguments is a vUnitType, the function returns a PtgNum parse token. If **Angle1** equals **Angle2**, the function returns a PtgErr parse token with an error code equal to #DIV/0.

2.5.3.75 IntersectY

The **IntersectY** function calculates the y-coordinate of the point where two lines intersect. Each line is defined by a point on the line and an angle.

ABNF:

```
IntersectY = "INTERSECTY(" val "," val "," val "," val "," val "," val ")"
```

Required Arguments:

Name: **X1**

Type: vDouble

An argument that specifies the x-coordinate of a point defining the first line.

Name: **Y1**

Type: vDouble

An argument that specifies the y-coordinate of a point defining the first line.

Name: **Angle1**

Type: vDouble

An argument that specifies the angle of the first line relative to the positive x-axis.

Name: **X2**

Type: vDouble

An argument that specifies the x-coordinate of the point defining the second line.

Name: **Y2**

Type: vDouble

An argument that specifies the y-coordinate of the point defining the second line.

Name: **Angle2**

Type: vDouble

An argument that specifies the angle of the second line relative to the positive x-axis.

Return Value:

Type: vNum, PtgErr

This function returns a custom token grouping of the type vNum containing the y-coordinate of the point where two lines intersect. If any of the arguments **X1**, **Y1**, **X2**, **Y2** are a custom token grouping of the type vUnitType, the unit of the return value is equal to the unit of the first of those arguments that is a vUnitType. If none of those arguments are a vUnitType, the function returns a PtgNum parse token. If **Angle1** equals **Angle2**, the function returns a PtgErr parse token with an error code equal to #DIV/0.

2.5.3.76 Intup

The **Intup** function performs a rounding calculation up to the next greater integer value.

ABNF:

```
Intup = "INTUP(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies a value to be rounded.

Return Value:

Type: vNum

This function returns a custom token grouping of the type vNum containing the next greater integer value if **Arg1** is not an integer. The unit of the return value is equal to the unit of **Arg1**.

2.5.3.77 Is1D

The **Is1D** function returns whether the shape contains BeginX, BeginY, EndX, or EndY cells.

ABNF:

```
Is1D = "IS1D()"
```

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with the value of **TRUE** if the shape contains any of the following cells: BeginX, BeginY, EndX, or EndY. Otherwise, it returns a PtgBool with the value of **FALSE**.

2.5.3.78 IsErr

The **IsErr** function calculates if the operand is a PtgErr parse token that does not have an error code equal to #N/A.

ABNF:

```
IsErr = "ISERR(" val ")"
```

Required Arguments:

Name: **Token**

Type: vAny

An argument that specifies the operand of the calculation.

Return Value:

Type: PtgBool

This function returns a parse token of the type PtgBool with a value of **TRUE** if **Token** is a PtgErr parse token and the value of **Token** is not equal to #N/A; otherwise, the value of **FALSE** is returned.

2.5.3.79 IsErrNA

The **IsErrNA** function calculates if the operand is a PtgErr parse token with an error code equal to #N/A.

ABNF:

```
IsErrNa = "ISERRNA(" val ")"
```

Required Arguments:

Name: **Token**

Type: vAny

An argument that specifies the operand of the calculation.

Return Value:

Type: PtgBool

This function returns a parse token of the type PtgBool with a value of **TRUE** if **Token** is a PtgErr and the value of **Token** is equal to #N/A; otherwise, the value **FALSE** is returned.

2.5.3.80 IsError

The **IsError** function calculates if the operand is a PtgErr parse token.

ABNF:

```
IsError = "ISERROR(" val ")"
```

Required Arguments:

Name: **Token**

Type: vAny

An argument that specifies the operand of the calculation.

Return Value:

Type: PtgBool

This function returns a parse token of the type PtgBool with a value of **TRUE** if **Token** is a PtgErr; otherwise, the value **FALSE** is returned.

2.5.3.81 IsErrValue

The **IsErrValue** function calculates if the operand is a parse token of the type PtgErr with an error code equal to #VALUE!.

ABNF:

```
IsErrValue = "ISERRVALUE(" val ")"
```

Required Arguments:

Name: **Token**

Type: vAny

An argument that specifies the operand of the calculation.

Return Value:

Type: PtgBool

This function returns a parse token of the type PtgBool with a value of **TRUE**, if **Token** is a PtgErr and the value of **Token** is equal to #VALUE!; otherwise, the value **FALSE** is returned.

2.5.3.82 IsThemed

The **IsThemed** function returns whether the shape has a theme applied.

ABNF:

```
IsThemed = "ISTHEMED()"
```

Return Value:

Type: PtgBool

This function returns a parse token of the type PtgBool with the value of **FALSE** if any of the following cells on the shape has a value equal to zero: ColorSchemeIndex, EffectSchemeIndex, ConnectorSchemeIndex, or FontSchemeIndex. Otherwise, it returns a PtgBool with the value of **TRUE**.

2.5.3.83 Keywords

The **Keywords** function returns the **Keyword** property of a web drawing.

ABNF:

```
Keyword = "KEYWORD()"
```

Return Value:

Type: PtgString

This function returns the **Keyword** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.84 Language

The **Language** function converts the operand from a **culture name** or an LCID to an implementation-specific language identifier.

ABNF:

```
Language = "LANGUAGE(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vLanguage

An argument that specifies the language.

Return Value:

Type: PtgInt

This function returns a parse token of the type PtgInt which uniquely identifies a language in the vLanguageString custom structure table. The value is specific to the implementation.

2.5.3.85 Left

The **Left** function returns a string containing the first character or characters in a string.

ABNF:

```
Left = "LEFT(" val [ "," val ] ")"
```

Required Arguments:

Name: **Text**

Type: vString

An argument that specifies a string.

Optional Arguments:

Name: **NumChars**

Type: vSignedLong

An argument that specifies the number of characters to be returned. The default value is one.

Return Value:

Type: PtgString

This function returns a parse token of the type PtgString containing the first **NumChars** characters of **Text**.

If **NumChars** is less than zero or greater than the number of characters in **Text**, the function returns a PtgString containing **Text**.

If **NumChars** is equal to zero, the function returns a PtgString containing an empty string.

2.5.3.86 Len

The **Len** function performs a calculation of the length of a string.

ABNF:

```
Len = "LEN(" val ")"
```

Required Arguments:

Name: **Text**

Type: vString

An argument that specifies a string.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the number of characters in **Text**.

2.5.3.87 Ln

The **Ln** function performs a natural logarithm calculation.

ABNF:

```
Ln = "LN(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: vNum, PtgErr

This function returns a custom token grouping of the type vNum containing the natural logarithm of **Arg1**. The unit of the return value is equal to the unit of **Arg1**. If **Arg1** is less than or equal to zero, the function returns a PtgErr parse token with an error code equal to #NUM!.

2.5.3.88 Loc

The **Loc** function performs a transformation of a point from the coordinate system of another shape into the coordinate system of the shape associated with the formula.

ABNF:

```
Loc = "LOC(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: PtgPnt

An argument that specifies a point in the coordinate system of a shape.

Return Value:

Type: PtgPnt

This function transforms the point value of **Arg1** in the coordinate system of the shape associated with **Arg1** into an equivalent point in the coordinate system of the shape associated with the formula and returns the transformed point. If the shape associated with the formula and the shape associated with **Arg1** are not on the same drawing page, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.89 LocalFormulaExists

The **LocalFormulaExists** function returns whether a cell contains a formula that is not inherited.

ABNF:

```
LocalFormulaExists = "LOCALFORMULAEXISTS(" val ")"
```

Required Arguments:

Name: Arg1

Type: CellRef

An argument that specifies a reference to a cell in a shape.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with the value of **TRUE** if the cell contains a formula and that formula is not inherited. Otherwise, it returns a PtgBool with the value of **FALSE**.

2.5.3.90 LocToLoc

The **LocToLoc** function performs a transformation of a point from the coordinate system of one shape into the coordinate system of another shape.

ABNF:

```
LocToLoc = "LOCTOLOC(" val "," val "," val ")"
```

Required Arguments:

Name: **Point**

Type: PtgPnt

An argument that specifies a point.

Name: **Source**

Type: CellRef

An argument that specifies a reference to a cell in a source shape.

Name: **Destination**

Type: CellRef

An argument that specifies a reference to a cell in a destination shape.

Return Value:

Type: PtgPnt

This function returns a transformed point in the coordinate system of the shape of **Destination** from the value of **Point** in a coordinate system of the shape of **Source**. If the **Source** shape and **Destination** shape are not on the same drawing page, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.91 LocToPar

The **LocToPar** function performs a transformation of a point from the coordinate system of one shape into the coordinate system of the parent of another shape.

ABNF:

```
LocToPar = "LOCTOPAR(" val "," val "," val ")"
```

Required Arguments:

Name: **Point**

Type: PtgPnt

An argument that specifies a point.

Name: **Source**

Type: CellRef

An argument that specifies a reference to a cell in a source shape.

Name: **Destination**

Type: CellRef

An argument that specifies a reference to a cell in a destination shape.

Return Value:

Type: PtgPnt

This function returns a transformed point in the coordinate system of the parent sheet of the shape of **Destination** from the value of **Point** in the coordinate system of the shape of **Source**. If the **Source's** shape and **Destination's** shape are not on the same drawing page, the function returns a PtgErr with an error code equal to #VALUE!.

2.5.3.92 Log10

The **Log10** function performs a base 10 logarithm calculation.

ABNF:

```
Log10 = "LOG10(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: vNum, PtgErr

This function returns a custom token grouping of the type vNum containing the base 10 logarithm of **Arg1**. The unit of the return value is equal to the unit of **Arg1**. If the value of **Arg1** is less than or equal to zero, the function returns a PtgErr parse token with an error code equal to #NUM!.

2.5.3.93 Lookup

The **Lookup** function returns the index of a substring in a list of substrings.

ABNF:

```
Lookup = "LOOKUP(" val "," val [ "," [ val ]] ")"
```

Required Arguments:

Name: **Substring**

Type: vString

An argument that specifies the substring to find.

Name: **List**

Type: vString

An argument that specifies a string consisting of a list of substrings delimited by **Delimiter**. Consecutive delimiters or delimiters at the beginning and/or end of **List** specify empty substrings.

Optional Arguments:

Name: **Delimiter**

Type: vString

An argument that specifies the delimiter that separates substrings inside **List** with a default value of ";", if missing.

Return Value:

Type: PtgNum

This function returns a parse token of the type PtgNum containing the zero-based index location of the substring in **List** that matches **Substring**. The leading and trailing spaces of substrings in **List** are ignored for the comparison. If **Substring** contains **Delimiter** or if **Substring** is not found in **List**, the function returns a PtgNum containing a value of -1.

The comparison is performed according to .NET globalization rules based on the value specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing. For more information about .NET globalization rules, see [MSDN-ENCLOC].

2.5.3.94 Lower

The **Lower** function performs lower case conversion.

ABNF:

```
Lower = "LOWER(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vString

An argument that specifies the string to convert.

Return Value:

Type: PtgString

This function returns a parse token of the type PtgString containing the string converted to lower case. The conversion is performed according to .NET globalization rules based on the value specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing. For more information about .NET globalization rules, see [MSDN-ENCLOC].

2.5.3.95 Lum

The **Lum** function calculates a luminance component, as specified in the hue-saturation-luminance (HSL) color space, of a color value.

ABNF:

```
Lum = "LUM(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vColor

An argument that specifies the color.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the luminance of **Arg1**. The value is less than or equal to 240.

2.5.3.96 LumDiff

The **LumDiff** function calculates the difference in luminance, as specified in the hue-saturation-luminance (HSL) color space, between two color values.

ABNF:

```
LumDiff = "LUMDIFF(" val "," val ")"
```

Required Arguments:

Name: **Arg1**

Type: vColor

An argument that specifies the first color.

Name: **Arg2**

Type: vColor

An argument that specifies the second color.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing a value that is equal to the luminance of **Arg1** minus the luminance of **Arg2**. This value is greater than or equal to -240 and less than or equal to 240.

2.5.3.97 Magnitude

The **Magnitude** function performs a magnitude calculation of a vector.

ABNF:

```
Magnitude = "MAGNITUDE(" val "," val "," val "," val ")"
```

Required Arguments:

Name: **ConstantRise**

Type: vDouble

An argument that specifies the constant factor for the rise.

Name: **Rise**

Type: vDouble

An argument that specifies a rise of the vector.

Name: **ConstantRun**

Type: vDouble

An argument that specifies a constant factor for the run.

Name: **Run**

Type: vDouble

An argument that specifies a run of the vector.

Return Value:

Type: PtgNum, PtgErr

The function returns a PtgNum parse token containing the calculated magnitude. The magnitude calculation is performed using the following formula:

SQRT((ConstantRise*Rise)^2 + (ConstantRun*Run)^2))

If **Rise** or **Run** is a PtgNumMultiDim parse token, the function returns a PtgErr parse token with an error code equal to #DIM!.

2.5.3.98 Manager

The **Manager** function returns the **Manager** property of a web drawing.

ABNF:

```
Manager = "MANAGER()"
```

Return Value:

Type: PtgString

This function returns the **Manager** property, specified in [ISO/IEC29500-1:2011] section 22.2.2.15, from the App XML part of a web drawing.

2.5.3.99 MasterName

The **MasterName** function returns the name of the master of a shape.

ABNF:

```
MasterName = [ShapeSheetRef "!"] "MASTERNAME(" [ val ] ")"
```

Optional Arguments:

Name: **Arg1**

Type: PtgInt

An argument to specify the language of the return value.

Return Value:

Type: PtgString

This function returns the name of the master of a shape in the current reference context or reference context specified by a ShapeSheetRef reference token.

If the value of **Arg1** is 750, the **NameU** attribute of the Master_Type element associated with the master is returned.

If the value of **Arg1** is not 750, the **Name** attribute of the Master_Type element associated with the master is returned.

If the shape does not have a master and the value of **Arg1** is 750, the string "<no master>" is returned.

If the shape does not have a master and the value of **Arg1** is not 750, language-dependent translation of the string "<no master>" is returned.

2.5.3.100 Max

The **Max** function performs a search for the largest number from a list of operands.

ABNF:

```
Max = "MAX(" val *("," val) ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand.

Optional Arguments:

Zero or more operands of type vDouble.

Return Value:

Type: vNum

This function returns a vNum custom token grouping containing the first occurrence of the largest operand. The unit of the return value is equal to the unit of that operand.

2.5.3.101 Mid

The **Mid** function returns part of a string, starting at the position specified, based on the number of characters specified.

ABNF:

```
Mid = "MID(" val "," val "," val ")"
```

Required Arguments:

Name: **Text**

Type: vString

An argument that specifies a text string.

Name: **StartPos**

Type: vSignedLong

An argument that specifies the one-based index in **Text** which represents the beginning of a substring to be returned.

Name: **NumChars**

Type: vSignedLong

An argument that specifies the number of characters to be returned.

Return Value:

Type: PtgString, PtgErr

This function returns a PtgString parse token containing a substring of **Text**, starting from **StartPos** up to the character at **StartPos** plus **NumChars** -1. If **StartPos** is zero, the function returns a PtgErr parse token with an error code equal to #VALUE!. If **StartPos** is less than zero or greater than the number of characters in **Text**, the function returns a PtgString containing an empty string. If **StartPos** plus **NumChars** -1 exceeds the length of **Text**, or if **NumChars** is less than zero, the function returns a PtgString containing the substring of **Text** starting from **StartPos** up to the end of **Text**.

2.5.3.102 Min

The **Min** function performs a search for the smallest number from a list of operands.

ABNF:

```
Min = "MIN(" val *("," val) ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand.

Optional Arguments:

Zero or more operands of type vDouble.

Return Value:

Type: vNum

This function returns a vNum custom token grouping containing the first occurrence of the smallest operand. The unit of the return value is equal to the unit of that operand.

2.5.3.103 Minute

The **Minute** function returns a minute from a value representing a date and time.

ABNF:

```
Minute = "MINUTE(" val ["," val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a date and time value.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgNum, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion is successful, the function returns a PtgNum parse token containing the minute component from **DateTimeArg.** If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.104 Modulus

The **Modulus** function performs a modulus calculation.

ABNF:

```
Modulus = "MODULUS(" val "," val ")"
```

Required Arguments:

Name: **Number**

Type: vDouble

An argument that specifies a number to be divided by **Divisor**.

Name: **Divisor**

Type: vFloat

An argument that specifies the divisor of the calculation.

Return Value:

Type: vNum, PtgErr

This function returns a vNum custom token grouping containing the remainder when **Number** is divided by **Divisor**. If **Number** is a vNum, the unit of the return value is equal to the unit of **Number**; otherwise, the function returns a PtgNum parse token. If **Divisor** is zero, the function returns a PtgErr parse token with an error code equal to #DIV/0.

2.5.3.105 Month

The **Month** function returns a month, according to the Gregorian calendar, from a value representing a date and time.

ABNF:

```
Month = "MONTH(" val [","  val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a date and time value.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgNum, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion is successful, the function returns a PtgNum parse token containing the month component from **DateTimeArg**. If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.106 MsoShade

The **MsoShade** function performs a modification to a color by decreasing the luminance component, as specified in the hue-saturation-luminance (HSL) color space, by a percentage.

ABNF:

```
MsoShade = "MSOSHADE(" val "," val ")"
```

Required Arguments:

Name: **Color**

Type: vColor

An argument that specifies a color.

Name: **Delta**

Type: vSignedInt

An argument that specifies the percentage to decrease the luminance of **Color**. If the value is less than zero, the luminance is increased.

Return Value:

Type: PtgColorRGB

This function returns a PtgColorRGB parse token containing the shaded color. If **Delta** is greater than zero, the luminance of **Color** is decreased by the percentage specified by the absolute value of **Delta**. If **Delta** is less than zero, the luminance of **Color** is increased by the following amount: the percentage specified by **Delta** multiplied by the result of 240 minus the luminance of **Color**. If **Delta** is less than -100, the function sets the value to -100 and performs the operation. If **Delta** is greater than 100, the function sets the value to 100 and performs the operation.

2.5.3.107 MsoTint

The **MsoTint** function modifies a color by increasing the luminance component, as specified in the hue-saturation-luminance (HSL) color space, by a percentage.

ABNF:

```
MsoTint = "MSOTINT(" val "," val ")"
```

Required Arguments:

Name: **Color**

Type: vColor

An argument that specifies a color.

Name: **Delta**

Type: vSignedInt

An argument that specifies the percentage to increase the luminance of **Color**. If the value is less than zero, the luminance is decreased.

Return Value:

Type: PtgColorRGB

This function returns a PtgColorRGB parse token containing the tinted color. If **Delta** is greater than zero, the luminance of **Color** is increased by the following amount: the percentage specified by **Delta** multiplied by the result of 240 minus the luminance of **Color**. If **Delta** is less than zero, the luminance of **Color** is decreased by the percentage specified by the absolute value of **Delta**. If **Delta** is less than -100, the function sets the value to -100 and performs the operation. If **Delta** is greater than 100, the function sets the value to 100 and performs the operation.

2.5.3.108 Mul

The **Mul** function performs a multiplication calculation.

ABNF:

```
Mul = "_MUL(" val "," val ")" / val "*" val
```

Required Arguments:

Name: **Arg1**

Type: vDoubleEx

An argument that specifies the first operand of the multiplication operation.

Name: **Arg2**

Type: vDoubleEx

An argument that specifies the second operand of the multiplication operation.

Return Value:

Type: vNum, PtgCy, PtgErr

This function returns a custom token grouping of the type **vNum** (section 2.5.7.4) or a **PtgCy** parse token (section 2.5.4.9) containing **Arg1** multiplied by **Arg2**. The type of the return value is by the following algorithm.

```
SET return type = PtgNum
SET returnDimension = 0
SET returnCurrencyID = 0

IF Arg1.Type = PtgCy AND Arg2.Type != PtgCy THEN
   SET returnType = PtgCy
   SET returnCurrencyID = currencyID of Arg1
ELSE IF Arg1.Type != PtgCy AND Arg2.Type = PtgCy THEN
   SET returnType = PtgCy
   SET returnCurrencyID = currencyID of Arg2
ELSE IF Arg1.Type = PtgCy AND Arg2.Type = PtgCy THEN
   SET returnType = PtgCy
   IF currencyID of Arg1 = currencyID of Arg2 OR currencyID of Arg2 = 1 THEN
      SET returnCurrencyID = currencyID of Arg1
   ELSE IF currencyID of Arg1 = 1 THEN
      SET returnCurrencyID = currencyID of Arg2
   ELSE
      SET returnType = PtgErr
   END IF
ELSE IF Arg1.Unit = PtgNumPct AND Arg2.Unit = PtgNumPct
   SET returnType = PtgNumPct
   SET returnDimension = 1
ELSE IF Arg1.Unit = PtgNumPct
   SET returnType = PtgNumPct
   SET returnDimension = Arg2.dimension
ELSE IF Arg2.Unit = PtgNumPct
   IF Arg1.Unit is a vUnitType
      SET returnType = Arg1.Unit
      SET returnDimension = Arg1.Dimension
   ELSE
      SET returnType = Arg2.Unit
      SET returnDimension = Arg1.Dimension
   END IF
ELSE IF Arg1.Unit is a vUnitType THEN
   SET returnType = Arg1.Unit
   SET returnDimension = Arg1.Dimension + Arg2.Dimension
ELSE IF Arg2.Unit is a vUnitType THEN
   SET returnType = Arg2.Unit
   SET returnDimension = Arg1.Dimension + Arg2.Dimension
END IF


IF returnType = PtgCy THEN
   Return type is PtgCy with currencyID = returnCurrencyID
ELSE IF returnType = PtgErr THEN
   Return type is PtgErr with error code of #VALUE!
```

```
ELSE IF (returnType = PtgAcre OR returnType = PtgHectare) AND returnDimension != 2 THEN
   Return type is PtgNumMultiDim with unit = PtgNumDft and dimension = returnDimension
ELSE IF returnDimension = 0 THEN
   Return type is PtgNum
ELSE IF returnDimension = 1 THEN
   Return type is returnType
ELSE
   Return type is PtgNumMultiDim with unit = returnType and dimension = returnDimension
END IF
```

2.5.3.109 NA

The **NA** function returns the error code #N/A.

ABNF:

```
NA = "NA()"
```

Return Value:

Type: PtgErr

This function returns a PtgErr parse token with the error code equal to #N/A.

2.5.3.110 Name

The **Name** function returns a name of a shape or style.

ABNF:

```
Name = [ (ShapeSheetRef / StyleSheetRef) "!"] "NAME(" [ val ] ")"
```

Optional Arguments:

Name: **Arg1**

Type: PtgInt

An argument to specify the language of the return value.

Return Value:

Type: PtgString

This function returns the name of the shape or style in the current reference context or reference context specified by ShapeSheetRef or StyleSheetRef.

If the value of **Arg1** is 750, the **NameU** attribute of the ShapeSheet_Type element or StyleSheet_Type element associated with the shape or style is returned.

If the value of **Arg1** is not 750, the **Name** attribute of the ShapeSheet_Type element or StyleSheet_Type element associated with the shape or style is returned.

2.5.3.111 Not

The **Not** function performs the Boolean NOT operation.

ABNF:

```
Not = "NOT(" val ")"
```

Required Arguments:

Name: **Value**

Type: vBoolean

An argument that specifies an operand.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token containing the value of the Boolean NOT operation on **Value**.

2.5.3.112 Now

The **Now** function returns the current date and time.

ABNF:

```
Now = "NOW()"
```

Return Value:

Type: PtgDate

This function returns a PtgDate parse token containing the value of current date and time.

2.5.3.113 Nurbs

The **Nurbs** function returns a non-uniform rational B-spline (NURBS).

ABNF:

```
Nurbs = "NURBS(" val "," val "," val "," val *("," val "," val "," val "," val) ")"
```

Required Arguments:

Name: **knotLast**

Type: vDouble

An argument that specifies the last knot.

Name: **degree**

Type: vSignedInt

An argument that specifies the degree of the B-spline.

Name: **xType**

Type: vUnsignedInt

An argument that specifies how to interpret the x-coordinates. If **xType** is zero, the input **xN** is interpreted as relative coordinates. Otherwise, the input **xN** is interpreted in the coordinate system of the shape.

Name: **yType**

Type: vUnsignedInt

An argument that specifies how to interpret the y-coordinates. If **yType** is zero, the input **yN** is interpreted as relative coordinates. Otherwise, the input **yN** is interpreted in the coordinate system of the shape.

Optional Arguments:

Additional x-coordinates, y-coordinates, knots, and weights MUST be specified using additional groups of **xN**, **yN**, **knotN**, and **weightN**.

Name: **xN**

Type: vDouble

An argument that specifies a x-coordinate.

Name: **yN**

Type: vDouble

An argument that specifies a y-coordinate.

Name: **knotN**

Type: vDouble

An argument that specifies a knot on the B-spline.

Name: **weightN**

Type: vDouble

An argument that specifies a weight on the B-spline.

Return Value:

Type: PtgNurbs, PtgErr

This function returns a PtgNurbs parse token containing **knotLast**, **degree**, **xType**, **yType**, **xN**, **yN**, **knotN**, and **weightN**. If the wrong number of arguments is used, the function returns a PtgErr parse token with a value equal to #VALUE!.

2.5.3.114 Or

The **Or** function returns a Boolean value.

ABNF:

```
Or = "OR(" val *( "," val ) ")"
```

Required Arguments:

Name: **Args**

Type: vBoolean

A set of arguments that specifies an operand of the calculation.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE** if any of the arguments are equal to **TRUE**; otherwise, the value of **FALSE** is returned.

2.5.3.115 PageCount

The **PageCount** function returns a count of the number of foreground drawing pages in a web drawing.

ABNF:

```
PageCount = "PAGECOUNT()"
```

Return Value:

Type: PtgInt

The number of foreground drawing pages in a web drawing.

2.5.3.116 PageName

The **PageName** function returns a name of a drawing page.

ABNF:

```
PageName = [ (CrossPageRef / PageSheetRef) "!"] "PAGENAME(" [ val ] ")"
```

Optional Arguments:

Name: **Arg1**

Type: PtgInt

An argument to specify the language of the return value.

Return Value:

Type: PtgString

This function returns the name of the drawing page in the current reference context or reference context specified by CrossPageRef or PageSheetRef.

If the value of **Arg1** is 750, the **NameU** attribute of the Page_Type element associated with the drawing page is returned.

If the value of **Arg1** is not 750, the **Name** attribute of the Page_Type element associated with the drawing page is returned.

If the function is called from outside the context of a page, an empty string is returned.

2.5.3.117 PageNumber

The **PageNumber** function returns an index of a drawing page.

ABNF:

```
PageNumber = "PAGENUMBER()"
```

Return Value:

Type: PtgInt

This function returns an index of the drawing page.

If the drawing page is a background page or if the function is called outside of a page context, zero is returned.

2.5.3.118 Par

The **Par** function performs a transformation of a point from the coordinate system of another shape into the coordinate system of the parent of the shape associated with the formula.

ABNF:

```
Par = "PAR(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: PtgPnt

An argument that specifies a point in the coordinate system of a shape.

Return Value:

Type: PtgPnt

This function transforms a point value of **Arg1** in the coordinate system of the shape associated with **Arg1** into an equivalent point in the coordinate system of the parent sheet of the shape associated with the formula. If the shape associated with the formula and the shape associated with the **Arg1** are not on the same drawing page, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.119 Pct

The **Pct** function performs a percent conversion on the operand.

ABNF:

```
Pct = "_PCT(" val ")"
```

Required Arguments:

Name: **Value**

Type: vDouble

An argument that specifies an operand that will be converted to a percent.

Return Value:

Type: PtgNumPct

This function returns a PtgNumPct parse token containing **Value** / 100.

2.5.3.120 Pi

The **Pi** function returns the mathematical constant pi.

ABNF:

```
Pi = "PI()"
```

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the value of the mathematical constant pi.

2.5.3.121 Pnt

The **Pnt** function returns a point in the coordinate system of a shape.

ABNF:

```
Pnt = "PNT(" val "," val ")"
```

Required Arguments:

Name: **X**

Type: vDouble

An argument that specifies the x-coordinate of the point.

Name: **Y**

Type: vDouble

An argument that specifies the y-coordinate of the point**.**

Return Value:

Type: PtgPnt

This function returns a PtgPnt parse token containing the value of **X** and **Y** as coordinates. If **X** or **Y** is a CellRef, the function returns a PtgPnt parse token containing the value of **X**, **Y**, and the shape specified by the CellRef.

2.5.3.122 Pntx

The **Pntx** function returns a value of the x-coordinate of a point.

ABNF:

```
PntX = "PNTX(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vNum, PtgPnt

An argument that specifies a point.

Return Value:

Type: vNum

If **Arg1** is a PtgPnt parse token, this function returns a custom token grouping of the type vNum with the value of the x-coordinate of the point. If **Arg1** is a vNum, the function returns **Arg1**.

2.5.3.123 PntY

The **PntY** function returns a value of a y-coordinate of a point.

ABNF:

```
PntY = "PNTY(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vNum, PtgPnt

An argument that specifies a point.

Return Value:

Type: vNum

If **Arg1** is a PtgPnt parse token, this function returns a custom token grouping of the type vNum with the value of the y-coordinate of the point. If **Arg1** is a vNum, the function returns **Arg1**.

2.5.3.124 PolyLine

The **PolyLine** function returns a polyline.

ABNF:

```
Polyline = "POLYLINE(" val "," val *("," val "," val) ")"
```

Required Arguments:

Name: **xType**

Type: vUnsignedInt

An argument that specifies how to interpret the x-coordinates. If **xType** is zero, the input **xN** is interpreted as relative coordinates. Otherwise, the input **xN** is interpreted in the coordinate system of the shape.

Name: **yType**

Type: vUnsignedInt

An argument that specifies how to interpret the y-coordinates. If **yType** is zero, the input **yN** is interpreted as relative coordinates. Otherwise, the input **yN** is interpreted in the coordinate system of the shape.

Optional Arguments:

Additional x and y-coordinates MUST be specified using additional pairs of **xN** and **yN**.

Name: **xN**

Type: vDouble

An argument that specifies an x-coordinate.

Name: **yN**

Type: vDouble

An argument that specifies a y-coordinate.

Return Value:

Type: PtgPolyline, PtgErr

This function returns a PtgPolyline parse token containing **xType**, **yType, xN**, and **yN**. If the wrong number of arguments are used, the function returns a PtgErr parse token with a value equal to #VALUE!.

2.5.3.125 Pow

The **Pow** function performs an exponentiation calculation.

ABNF:

```
Pow = "POW(" val "," val ")" / val "^" val
```

Required Arguments:

Name: **Base**

Type: vDouble

An argument that specifies a number to be raised to the power of **Exponent**.

Name: **Exponent**

Type: vDouble

An argument that specifies an exponent by which to raise **Base**.

Return Value:

Type: vNum, PtgErr

This function returns the custom token grouping vNum (section 2.5.7.4) or PtgErr parse token (section 2.5.4.14) containing Base raised to the power of Exponent. The type of the return value is calculated by the following algorithm.

```
SET returnType = PtgNum
SET returnError = #NUM!
SET returnDimension = 0

IF Base.Value = 0 THEN
   IF Exponent.Value = 0 THEN
      SET returnType = PtgErr
      SET returnError = #NUM!
   ELSE IF Exponent.Value < 0 THEN
      SET returnType = PtgErr
      SET returnError = #DIV/0
   END IF
ELSE IF Base.Value < 0 AND Exponent is not an integer THEN
   SET returnType = PtgErr
   SET returnError = #DIM!
ELSE IF Base.Unit = PtgNumPct THEN
   SET returnType = PtgNumPct
   SET returnDimension = 1
ELSE
   SET returnType = Base.Unit
   SET returnDimension = Base.Dimension * Exponent.Value
END IF

IF returnType = PtgErr THEN
   Return type is PtgErr with error code of returnError
ELSE IF returnDimension = 0 THEN
   Return type is PtgNum
ELSE IF returnDimension = 1 THEN
   Return type is returnType
ELSE
   Return type is PtgNumMultiDim with unit = returnType and dimension = returnDimension
END IF
```

2.5.3.126 Rad

The **Rad** function converts a value to an angle in radians.

ABNF:

```
Rad = "RAD(" val ")"
```

Required Arguments:

Name: **Angle**

Type: vDouble

An argument that specifies an angle.

Return Value:

Type: PtgAngRad

This function returns a parse token of the type PtgAngRad. If **Angle** is a vAngle custom token grouping, the value is equal to the value of **Angle**. Otherwise, **Angle** is interpreted as an angle in degrees and the return value is equal to:

pi / 180.0 * value of **Angle**

2.5.3.127 Rand

The **Rand** function generates a random number.

ABNF:

```
Rand = "RAND()"
```

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing a random double precision floating-point number greater than or equal to 0.0 and less than 1.0.

2.5.3.128 Ref

The **Ref** function returns an error code #REF!.

ABNF:

```
Ref = "REF()"
```

Return Value:

Type: PtgErr

This function returns a PtgErr parse token with the error code equal to #REF!.

2.5.3.129 Replace

The **Replace** function replaces part of a text string with another text string.

ABNF:

```
Replace = "REPLACE(" val "," val "," val "," val ")"
```

Required Arguments:

Name: **SourceText**

Type: vString

An argument that specifies a string to perform replacement on.

Name: **StartPos**

Type: vSignedLong

An argument that specifies the one-based starting position in **SourceText** where replacement begins.

Name: **NumChars**

Type: vSignedLong

An argument that specifies the number of characters in **SourceText** to be replaced by **ReplaceText**.

Name: **ReplaceText**

Type: vString

An argument that specifies a string to use for replacement.

Return Value:

Type: PtgString, PtgErr

This function returns a PtgString parse token containing **SourceText** modified with the replaced text. If **StartPos** is less than or equal to zero or greater than the number of characters in **SourceText**, the function returns a PtgString containing **SourceText** with **ReplaceText** appended at the end. If **StartPos** plus **NumChars** -1 exceeds the length of **SourceText**, or if **NumChars** is less than zero, the function returns a PtgString containing **SourceText** truncated starting from **StartPos** and with **ReplaceText** appended at the end.

2.5.3.130 RGB

The **RGB** function calculates an RGB color value as a combination of red, green and blue components.

ABNF:

```
RGB = "RGB(" val "," val "," val ")"
```

Required Arguments:

Name: **Red**

Type: vUnsignedInt

An argument that specifies the intensity of red.

Name: **Green**

Type: vUnsignedInt

An argument that specifies the intensity of green.

Name: **Blue**

Type: vUnsignedInt

An argument that specifies the intensity of blue.

Return Value:

Type: PtgColorRGB

This function returns a PtgColorRGB parse token containing a color value. The byte that specifies red in the PtgColorRGB is equal to **Red**. The byte that specifies green in the PtgColorRGB is equal to **Green**. The byte that specifies blue in the PtgColorRGB is equal to **Blue**. If the value of any of the arguments is greater than 0xFF, the function sets the argument's value to the result of the bitwise AND operation between the original value and 0xFF, before performing the operation.

2.5.3.131 Right

The **Right** function returns a string containing the last character or characters in a string.

ABNF:

```
Right = "RIGHT(" val [ "," val ] ")"
```

Required Arguments:

Name: **Text**

Type: vString

An argument that specifies a string.

Optional Arguments:

Name: **NumChars**

Type: vSignedLong

An argument that specifies the number of characters to be returned. The default value is one.

Return Value:

Type: PtgString

This function returns a PtgString parse token containing the last **NumChars** characters of **Text**.

If **NumChars** is less than zero or greater than the number of characters in **Text**, the function returns a PtgString containing **Text**.

If **NumChars** is equal to zero, the function returns a PtgString containing an empty string.

2.5.3.132 Round

The **Round** function performs a rounding calculation.

ABNF:

```
Round = "ROUND(" val "," val ")"
```

Required Arguments:

Name: **Number**

Type: vDouble

An argument that specifies a number to round.

Name: **Digits**

Type: vSignedInt

An argument that specifies the decimal place to use for the rounding operation.

Return Value:

Type: vNum

This function returns a vNum custom token grouping containing the rounded value of **Number**, as described by the following table.

Condition	Return value
Digits > 0	The function returns **Number** rounded to **Digits** places to the right of the decimal point.
Digits = 0	The function returns **Number** rounded to an integer.
Digits < 0	The function returns **Number** rounded to negative **Digits** places to the left of the decimal point.

The unit of the return value is equal to the unit of **Number**. If **Digits** is less than or equal to -9 or greater than 15, the function returns **Number**.

2.5.3.133 Sat

The **Sat** function calculates the saturation component, as specified in the hue-saturation-luminance (HSL) color space, of a color value.

ABNF:

```
Sat = "SAT(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vColor

An argument that specifies the color.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the saturation of **Arg1**. The value is less than or equal to 240.

2.5.3.134 SatDiff

The **SatDiff** function calculates the difference in saturation, as specified in the hue-saturation-luminance (HSL) color space, between two color values.

ABNF:

```
SatDiff = "SATDIFF(" val "," val ")"
```

Required Arguments:

Name: **Arg1**

Type: vColor

An argument that specifies the first color.

Name: **Arg2**

Type: vColor

An argument that specifies the second color.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing a value that is equal to the saturation of **Arg1** minus the saturation of **Arg2**. This value is greater than or equal to -240 and less than or equal to 240.

2.5.3.135 Second

The **Second** function returns the second component from a value representing a date and time.

ABNF:

```
Second = "SECOND(" val [","  val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a date and time value.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgNum, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion is successful, the function returns a PtgNum parse token containing the second component from **DateTimeArg.** If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.136 SetAtRef

The **SetAtRef** function returns the value of a cell.

ABNF:

```
SetAtRef = "SETATREF(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: CellRef

An argument that specifies a reference to a cell.

Return Value:

Type: vAny

This function returns the value of the cell referenced by **Arg1**.

2.5.3.137 SetAtRefEval

The **SetAtRefEval** function returns the value of the passed argument.

ABNF:

```
SetAtRefEval = "SETATREFEVAL(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vAny

An argument to be returned.

Return Value:

Type: vAny

This function returns the value of **Arg1**.

2.5.3.138 SetAtRefExpr

The **SetAtRefExpr** function returns the value of the passed argument.

ABNF:

```
SetAtRefExpr = "SETATREFEXPR(" [ val ] ")"
```

Optional Arguments:

Name: **Arg1**

Type: vAny

An argument to be returned. The default value is a PtgNum parse token with a value of zero.

Return Value:

Type: vAny

This function returns the value of **Arg1**.

2.5.3.139 Shade

The **Shade** function decreases the luminance, as specified in the hue-saturation-luminance (HSL) color space, of a color value.

ABNF:

```
Shade = "SHADE(" val "," val ")"
```

Required Arguments:

Name: **Color**

Type: vColor

An argument that specifies a color.

Name: **Delta**

Type: vSignedInt

An argument that specifies an amount to modify the luminance of **Color**.

Return Value:

Type: PtgColorRGB

This function returns a PtgColorRGB parse token containing a color value with decreased luminance. If **Delta** is less than zero, the luminance is increased.

2.5.3.140 ShapeText

The **ShapeText** function returns the text of a specified shape.

ABNF:

```
ShapeText = "SHAPETEXT"( val [ "," val ] ")"
```

Required Arguments:

Name: **Arg1**

Type: CellRef

This argument references a TheText cell.

Optional Arguments:

Name: **Arg2**

Type: vUnsignedInt

This argument specifies the operations to perform on the text.

The value is zero or a combination of values from this table.

Value	Meaning
0x01	Include discretionary hyphens.
0x02	Don't include expanded text from fields.
0x04	Convert tabs to a single space.
0x08	Convert tabs to spaces.
0x10	Convert CR and LF to spaces.
0x20	Convert typographer's quotes to straight quotes.
0x40	Convert contiguous **whitespace** into a single space.

Return Value:

Type: PtgString

The function returns a PtgString parse token. If the value of **Arg1** is a TheText cell of a sheet, this function returns the text from the shape after specified operations are applied. Otherwise, this function returns an empty string.

2.5.3.141 Sign

The **Sign** function returns the sign of a number.

ABNF:

```
Sign = "SIGN(" val [","  val] ")"
```

Required Arguments:

Name: **Number**

Type: vDouble

An argument that specifies the operand of the calculation.

Optional Arguments:

Name: **FuzzValue**

Type: vDouble

An argument that specifies the tolerance. The default value is 1E-9 (0.000000001).

Return Value:

Type: PtgInt

This function returns a PtgInt parse token representing the sign of **Number**. If **Number** is greater than the absolute value of **FuzzValue**, the function returns one. If **Number** is less than the negative of the absolute value of **FuzzValue**, the function returns -1; otherwise, the function returns zero.

2.5.3.142 Sin

The **Sin** function performs a sine calculation.

ABNF:

```
Sin = "SIN(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation. If **Arg1** is not a custom token grouping of the type vAngle, the value of **Arg1** is assumed to have the unit of radians.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the sine of **Arg1**.

2.5.3.143 SinH

The **SinH** function performs a hyperbolic sine calculation.

ABNF:

```
SinH = "SINH(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation. If **Arg1** is not a custom token grouping of the type vAngle, the value of **Arg1** is assumed to have the unit of radians.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the hyperbolic sine of **Arg1**.

2.5.3.144 Sqrt

The **Sqrt** function performs a square root calculation.

ABNF:

```
Sqrt = "SQRT(" val ")"
```

Required Arguments:

Name: **Number**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: vNum, PtgErr

This function returns a vNum custom token grouping containing the square root of **Number**. The unit of the return value is equal to the unit of **Number**. If the dimension of **Number** is greater than two, the function returns a PtgNumMultiDim parse token with value equal to the square root of **Number**, unit equal to the unit of **Number**, and dimension equal to the dimension of **Number** divided by two. If **Number** is less than zero, the function returns a PtgErr parse token with an error code equal to #NUM!. If the dimension of **Number** is not divisible by two, the function returns a PtgErr with an error code equal to #DIM!.

2.5.3.145 StrSame

The **StrSame** function determines whether two strings are equivalent.

ABNF:

```
StrSame = "STRSAME(" val "," val ["," val] ")"
```

Required Arguments:

Name: **FirstString**

Type: vString

An argument that specifies the first string to be compared.

Name: **SecondString**

Type: vString

An argument that specifies the second string to be compared.

Optional Arguments:

Name: **IgnoreCase**

Type: vBoolean

An argument that specifies whether the comparison is case insensitive. A value of **TRUE** indicates that case is ignored; otherwise, the default value is **FALSE**.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token with a value of **TRUE,** if the strings are equivalent; otherwise, the value of **FALSE** is returned. The comparison is performed according to .NET globalization rules based on the value specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing. For more information about .NET globalization rules, see [MSDN-ENCLOC].

2.5.3.146 StrSameEx

The **StrSameEx** function determines whether two text strings are the same.

ABNF:

```
StrSameEx = "STRSAMEEX(" val "," val "," val "," val ")"
```

Required Arguments:

Name: **FirstText**

Type: vString

An argument that specifies the first string to be compared.

Name: **SecondText**

Type: vString

An argument that specifies the second string to be compared.

Name: **Locale**

Type: vLanguageID

An argument that specifies the LCID of the culture used by the string comparison, according to .Net globalization rules. A value of zero specifies the invariant culture. For more information about .NET globalization rules, see [MSDN-ENCLOC].

Name: **Flag**

Type: vSignedLong

An argument that specifies additional rules to be used in the comparison. The value is a combination of values from the following table. The rules given in the following table correspond to members of the .NET CompareOptions enumeration, as described in [MSDN-CompareOptions].

Value	Meaning
0x00	No additional rules apply.
0x01	Ignore case.
0x02	Ignore non-spacing combining characters.
0x04	Ignore symbols and punctuation.
0x10000	Ignore differences between hiragana and katakana characters that represent the same phonetic sound.
0x20000	Ignore character width, or differences between the single-byte and double-byte representations of the same character.

Return Value:

Type: PtgBool

This function returns a PtgBool parse token containing **TRUE**, if the strings are the same. If the strings are not the same or if the **Locale** or **Flag** arguments are not valid, this function returns a PtgBool containing a value of **FALSE**.

2.5.3.147 Sub

The **Sub** function performs a subtraction calculation.

ABNF:

```
Sub = "_SUB(" val "," val ")" / val "-" val
```

Required Arguments:

Name: **Arg1**

Type: vDoubleEx

An argument that specifies the first operand of the calculation.

Name: **Arg2**

Type: vDoubleEx

An argument that specifies the second operand of the calculation.

Return Value:

Type: vNum, PtgCy, PtgErr

This function returns a custom token grouping of the type **vNum** (section 2.5.7.4) or a **PtgCy** parse token (section 2.5.4.9) containing **Arg1** minus **Arg2**. The type of the return value is calculated by the following algorithm.

```
SET returnType = PtgNum
SET returnDimension = 0
SET returnCurrencyID = 0

IF Arg1.Type = PtgCy AND Arg2.Type != PtgCy THEN
   SET returnType = PtgCy
   SET returnCurrencyID = currencyID of Arg1
ELSE IF Arg1.Type != PtgCy AND Arg2.Type = PtgCy THEN
   SET returnType = PtgCy
   SET returnCurrencyID = currencyID of Arg2
ELSE IF Arg1.Type = PtgCy AND Arg2.Type = PtgCy THEN
   SET returnType = PtgCy
   IF currencyID of Arg1 = currencyID of Arg2 OR currencyID of Arg2 = 1 THEN
      SET returnCurrencyID = currencyID of Arg1
   ELSE IF currencyID of Arg1 = 1 THEN
      SET returnCurrencyID = currencyID of Arg2
   ELSE
      SET returnType = PtgErr
   END IF
ELSE IF Arg1.Type = PtgDate AND Arg2.Type = PtgDate THEN
   SET returnType = PtgTDurDft
   SET returnDimension = 1
ELSE IF Arg1.Unit = PtgNumDft AND Arg2.Unit is a vUnitType THEN
   SET returnType = Arg2.Unit
   SET returnDimension = 1
ELSE IF Arg1.Unit is a vUnitType THEN
   SET returnType = Arg1.Unit
   IF Arg2.Dimension = 0 OR Arg2.Dimension = Arg1.Dimension THEN
      SET returnDimension = Arg1.Dimension
```

```
      ELSE
         SET returnDimension = 1
      END IF

      IF (returnType = PtgAcre OR returnType = PtgHectare) AND returnDimension != 2 THEN
         SET returnType = PtgNumDft
      END IF
   ELSE IF Arg2.Unit is a vUnitType THEN
      SET returnType = Arg2.Unit
      IF Arg1.Dimension = 0 OR Arg2.Dimension = Arg1.Dimension THEN
         SET returnDimension = Arg2.Dimension
      ELSE
         SET returnDimension = 1
      END IF

      IF (returnType = PtgAcre OR returnType = PtgHectare) AND returnDimension != 2 THEN
         SET returnType = PtgNumDft
      END IF
   END IF
   IF returnType = PtgCy THEN
      Return type is PtgCy with currencyID = returnCurrencyID
   ELSE IF returnType = PtgErr THEN
      Return type is PtgErr with error code of #VALUE!
   ELSE IF returnDimension = 0 THEN
      Return type is PtgNum
   ELSE IF returnDimension = 1 THEN
      Return type is returnType
   ELSE
      Return type is PtgNumMultiDim with unit = returnType and dimension = returnDimension
   END IF
```

2.5.3.148 Subject

The **Subject** function returns a **Subject** property of a web drawing.

ABNF:

```
Subject = "SUBJECT()"
```

Return Value:

Type: PtgString

This function returns the **Subject** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.149 Substitute

The **Substitute** function returns a string where substring substitution has been performed.

ABNF:

```
Substitute = "SUBSTITUTE(" val "," val "," val [ "," [ val ] [ "," [ val ]]] ")"
```

Required Arguments:

Name: **SourceString**

Type: vString

An argument that specifies a string to perform substitution on.

Name: **Substring**

Type: vString

An argument that specifies a substring in **SourceString** to be replaced.

Name: **SubstituteString**

Type: vString

An argument that specifies a string to use for substitution.

Optional Arguments:

Name: **Index**

Type: vSignedInt

An argument that specifies a one based index, among multiple instances of **Substring** in **SourceString,** is to be substituted.

Name: **IgnoreCase**

Type: vBoolean

An argument that specifies whether the search for **Substring** in **SourceString** is case insensitive. A value of **TRUE** specifies that case is ignored. If missing, the default value is **FALSE**.

Return Value:

Type: PtgString, PtgErr

This function returns a PtgString parse token containing **SourceString** modified with the substituted text. If **Index** is zero, the function returns a PtgErr parse token with an error code equal to #VALUE!. If **Index** is less than zero or greater than the number of **Substring** instances in **SourceString**, or if **Substring** is not found in **SourceString**, the function returns a PtgString containing **SourceString** without any substitution. If **Index** is missing, all instances of **Substring** are substituted. The search is performed according to .NET globalization rules based on the value specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing. For more information about .NET globalization rules, see [MSDN-ENCLOC].

2.5.3.150 Sum

The **Sum** function performs a sum calculation.

ABNF:

```
Sum = "SUM(" val *("," val) ")"
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the first operand of the calculation.

Optional Arguments:

Zero or more subsequent operands of type vDouble.

Return Value:

Type: vNum

This function returns a custom token grouping of the type vNum containing the sum of all operands. If any of the arguments is a vUnitType custom token grouping, the unit of the return value is equal to the unit of the first argument that is a vUnitType; otherwise, a PtgNum parse token is returned.

2.5.3.151 Tan

The **Tan** function performs a tangent calculation.

ABNF:

```
Tan = "TAN(" val ")"
```

Required Arguments:

Name: **Angle**

Type: vDouble

An argument that specifies the operand of the calculation. If **Arg1** is not a vAngle custom token, the value of **Arg1** is assumed to have the unit of radians.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the tangent of the value of **Arg1**.

2.5.3.152 TanH

The **TanH** function performs a hyperbolic tangent calculation.

ABNF:

```
TanH = "TANH(" val ")"
```

Required Arguments:

Name: **Angle**

Type: vDouble

An argument that specifies the operand of the calculation. If **Arg1** is not a vAngle custom token grouping, the value of **Arg1** is assumed to have the unit of radians.

Return Value:

Type: PtgNum

This function returns a PtgNum parse token containing the hyperbolic tangent of the value of **Arg1**.

2.5.3.153 TextHeight

The **TextHeight** function returns the height of the composed text in a shape.

ABNF:

```
TextHeight = "TEXTHEIGHT(" val "," val ")"
```

Required Arguments:

Name: **Shape**

Type: CellRef

An argument that specifies a reference to a cell in a shape.

Name: **MaximumWidth**

Type: vDouble

An argument that specifies the maximum allowable width of the composed text.

Return Value:

Type: PtgNumDft, PtgErr

This function returns a PtgNumDft parse token containing the height of the composed text in the shape referenced by **Shape**. If **MaximumWidth** is specified, the text composition fits the longest line of composed text within **MaximumWidth**. If **Shape** is not a reference to a cell specified by a TheTextCell_Type element, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.154 TextWidth

The **TextWidth** function returns the width of the composed text in a shape.

ABNF:

```
TextWidth = "TEXTWIDTH(" val [ "," val ] ")"
```

Required Arguments:

Name: **Shape**

Type: CellRef

An argument that specifies a reference to a cell in a shape.

Optional Arguments:

Name: **MaximumWidth**

Type: vDouble

An argument that specifies the maximum allowable width of the composed text.

Return Value:

Type: PtgNumDft, PtgErr

This function returns a PtgNumDft parse token containing the width of the composed text in the shape referenced by **Shape**. If **MaximumWidth** is specified, the text composition fits the longest line of composed text within **MaximumWidth**. If **Shape** is not a reference to a cell specified by a TheText Cell_Type element, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.155 Theme

The **Theme** function returns a fixed theme property value.

ABNF:

```
Theme = "THEME(" val ")"
```

Required Arguments:

Name: **ThemeProperty**

Type: vThemeString

An argument that specifies the property to be returned.

Return Value:

Type: vThemeColor, vThemeEffect, PtgErr

If the **ThemeProperty** is a fixed theme color scheme property as specified by the value "Color" in the "Scheme" column in the vThemeString table and the value of the structure of the Value Cell_Type child element of the msvThemeColors Row_Type child element of a User Section_Type element is less than 254, this function returns vThemeColor. The fixed theme index is specified by the value of the structure of the Value Cell_Type child element of the msvThemeColors Row_Type element and the fixed theme property is specified by **ThemeProperty**. If the **V** attribute of the Value Cell_Type child element of the msvThemeColors Row_Type element has a value equal to 254 and the argument of the USE function token of the **F** attribute of the Value Cell_Type child element is equal to the **UniqueID** attribute of a Master_Type element associated with a master, the **ThemeProperty** argument specifies a custom fixed color scheme property value specified by the Value Cell_Type child element of the User Row_Type element specified by the "User Row_Type" column in the vThemeString table.

If the **ThemeProperty** is a fixed theme effect scheme property as specified by the value "Effect" in the "Scheme" column in vThemeString table and the value of the structure of the Value Cell_Type child element of the msvThemeEffects Row_Type child element of a User Section_Type element is less than 254, this function returns vThemeEffect. The fixed theme index is specified by the value of the structure of the Value Cell_Type child element of the msvThemeEffects Row_Type element and the fixed theme property is specified by **ThemeProperty**. If the **V** attribute of the Value Cell_Type child element of the msvThemeEffects Row_Type element has a value equal to 254 and the argument of the USE function token of the **F** attribute of the Value Cell_Type child element is equal to the **UniqueID** attribute of a Master_Type element associated with a master, the **ThemeProperty** argument specifies a custom fixed effect scheme property value specified by the Value Cell_Type child element of the User Row_Type element specified by the "User Row_Type" column in the vThemeString table.

If **ThemeProperty** is not found in the table of vThemeString, this function returns a PtgErr parse token with a value equal to #VALUE!.

2.5.3.156 ThemeCBV

The **ThemeCBV** function returns a color with modified luminance and saturation, as specified in the hue-saturation-luminance (HSL) color space, using the tint ([ISO/IEC29500-1:2011] section

20.1.2.3.34) and shade ([ISO/IEC29500-1:2011] section 20.1.2.3.31) color modifiers specified by a fill gradient stop fill property of a dynamic theme.

ABNF:

```
ThemeCBV = "THEMECBV(" val "," val ")"
```

Required Arguments:

Name: **Color**

Type: vColor

An argument that specifies a color.

Name: **StopNum**

Type: PtgByte

An argument that specifies the fill gradient stop number.

Return Value:

Type: PtgColorRGB, PtgErr

This function returns the **Color** argument modified by the tint ([ISO/IEC29500-1:2011] section 20.1.2.3.34) and shade ([ISO/IEC29500-1:2011] section 20.1.2.3.31) color modifiers of the color of a **gs** element as specified by the **CT_GradientStop** type specified in [ISO/IEC29500-1:2011] section 20.1.8.36 of the one-based index position specified by the **StopNum** argument in a **gsLst** child element as specified by the **CT_GradientStopList** type specified in [ISO/IEC29500-1:2011] section 20.1.8.37 of a **gradFill** element as specified by the **CT_GradientFillProperties** type specified in [ISO/IEC29500-1:2011] section 20.1.8.33 of the Theme_XML_Part associated with the dynamic theme.

If the value of the structure of a ThemeIndex, ColorSchemeIndex, EffectSchemeIndex, ConnectorSchemeIndex, or FontSchemeIndex Cell_Type element in the containing shape, master, or style is equal to zero, this function returns a PtgColorRGB parse token with a value equal to the **Color** argument.

If the value of the **StopNum** argument is greater than 10 or less than one, this function returns a PtgErr parse token with a value equal to #VALUE!.

2.5.3.157 ThemeGuard

The **ThemeGuard** function returns the value of the passed argument.

ABNF:

```
ThemeGuard = "THEMEGUARD(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vAny

An argument to be returned.

Return Value:

Type: vAny

This function returns the value of **Arg1**.

2.5.3.158 ThemeProp

The **ThemeProp** function returns the embellishment and multiformat property values of a dynamic theme.

ABNF:

```
ThemeProp = "THEMEPROP(" val ")"
```

Required Arguments:

Name: **Arg**

Type: vString

Return Value:

Type: PtgByte, PtgErr

If the value of the **Arg** argument is equal to "embellishment", this function returns the embellishment property value from the dynamic theme. If the value of the **Arg** argument is equal to "multiformat", this function returns the multiformat property value from the dynamic theme.

If the value of the structure of a ThemeIndex, ColorSchemeIndex, EffectSchemeIndex, ConnectorSchemeIndex, or FontSchemeIndex Cell_Type element in the containing shape, master, or style is equal to zero, and if the value of the **Arg** argument is equal to "embellishment" or "multiformat", this function returns the value zero.

If the value of the **Arg** argument is not equal to "embellishment" or "multiformat", this function returns a PtgErr with a value equal to #VALUE!.

2.5.3.159 ThemeRestore

The **ThemeRestore** function returns a value of zero.

ABNF:

```
ThemeRestore = "THEMERESTORE(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vAny

An argument to be ignored.

Return Value:

Type: PtgInt

This function returns a PtgInt parse token with a value of zero.

2.5.3.160 ThemeVal

The **ThemeVal** function returns a dynamic theme property value.

ABNF:

```
ThemeVal = "THEMEVAL(" [ val ] "," [ val ] ")"
```

Optional Arguments:

Name: **ThemeProperty**

Type: vDynamicThemeString

An argument that specifies a property to be returned.

Name: **Arg2**

Type: vAny

An argument to be returned.

Return Value:

Type: PtgColorRGB, vFont, PtgNumPct, PtgNum, PtgTypPt, PtgNumI, PtgBool, vAngle, PtgByte, vLength, PtgInt, vAny, PtgErr

If this function is called without argument, it returns the dynamic theme property value of the Cell_Type element that it resides in directly without invoking theme inheritance. If this function is called with an argument, it returns the property value from the dynamic theme specified by the **ThemeProperty** argument directly without invoking theme inheritance.

If the **V** attribute of the ColorSchemeIndex Cell_Type element of a shape, master, or style is equal to 65535 and the argument of the USE function token of the **F** attribute of the ColorSchemeIndex Cell_Type element is equal to the **UniqueID** attribute of a Master_Type element associated with a master, the **ThemeProperty** argument specifies a custom dynamic theme color scheme property value specified by the Value Cell_Type child element of the User Row_Type elements specified in the "User Row_Type" column in the vDynamicThemeString table for the arguments "Dark1", "Light1", "AccentColor", "AccentColor2", "AccentColor3", "AccentColor4", "AccentColor5", "AccentColor6", and "BackgroundColor".

If the value of the structure of a ThemeIndex, ColorSchemeIndex, or FontSchemeIndex Cell_Type element in the containing shape, master, or style is equal to zero and **Arg2** is not specified, this function returns the property value specified by the "No Theme" column in the vDynamicThemeString table. If the value of the structure of a ThemeIndex, ColorSchemeIndex, EffectSchemeIndex, ConnectorSchemeIndex, or FontSchemeIndex Cell_Type element in the containing shape, master, or style is equal to zero and **Arg2** is specified, this function returns the value of **Arg2**.

If the value of the structure of a QuickStyleVariation Cell_Type element in the containing shape, master, or style is greater than or equal to 2 and less than or equal to 15, then the formula evaluation of the "TextColor", "LineColor", and "FillColor" vDynamicThemeString **ThemeProperty** arguments are affected as described in the QuickStyleVariation Cell_Type element table.

If the value of the **ThemeProperty** argument is not found in the vDynamicThemeString table, this function returns a PtgErr parse token with a value equal to #VALUE!.

2.5.3.161 Time

The **Time** function returns a time from values representing an hour, minute, and second.

ABNF:

```
Time = "TIME(" val "," val "," val ")"
```

Required Arguments:

Name: **Hour**

Type: vUnsignedInt

An argument that specifies an offset in hours from midnight.

Name: **Minute**

Type: vUnsignedInt

An argument that specifies an offset in minutes from **Hour**.

Name: **Second**

Type: vUnsignedInt

An argument that specifies an offset in seconds from **Minute**.

Return Value:

Type: PtgDate

This function returns a PtgDate parse token containing a time of day. The date-time-value component of the return value is the signed fractional value of the double precision number representing the input date-time, as specified in [MS-OAUT] section 2.2.25.

2.5.3.162 TimeValue

The **TimeValue** function returns a time component from a value representing a date and time.

ABNF:

```
TimeValue = "TIMEVALUE(" val ["," val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a value representing a date and time.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgDate, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion if successful, the function returns a PtgDate containing the time component of **DateTimeArg**. If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.163 Tint

The **Tint** function increases the luminance, as specified in the hue-saturation-luminance (HSL) color space, of a color value.

ABNF:

```
Tint = "TINT(" val "," val ")"
```

Required Arguments:

Name: **Color**

Type: vColor

An argument that specifies a color.

Name: **Delta**

Type: vSignedInt

An argument that specifies an amount to modify the luminance of **Color**.

Return Value:

Type: PtgColorRGB

This function returns a PtgColorRGB parse token containing a color value with increased luminance. If **Delta** is less than zero, the luminance is decreased.

2.5.3.164 Title

The **Title** function returns the **Title** property of a web drawing.

ABNF:

```
Title = "TITLE()"
```

Return Value:

Type: PtgString

This function returns the **Title** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

2.5.3.165 Tone

The **Tone** function decreases the saturation, as specified in the hue-saturation-luminance (HSL) color space, of a color value.

ABNF:

```
Tone = "TONE(" val "," val ")"
```

Required Arguments:

Name: **Color**

Type: vColor

An argument that specifies a color.

Name: **Delta**

Type: vSignedInt

An argument that specifies the amount to modify the saturation of **Color**.

Return Value:

Type: PtgColorRGB

This function returns a PtgColorRGB parse token containing a color value with decreased saturation. If **Delta** is less than zero, the saturation is increased.

2.5.3.166 Trim

The **Trim** function performs the removal of all whitespace from a string except for single spaces between words.

ABNF:

```
Trim = "TRIM(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vString

An argument that specifies a string from which to remove whitespace.

Return Value:

Type: PtgString

This function returns a PtgString parse token containing the string with whitespace removed. The following Unicode characters are considered as whitespace.

Value	Meaning
"\u0009"	Character Tabulation
"\u000a"	Line Feed
"\u000b"	Line Tabulation
"\u000c"	Form Feed

Value	Meaning
"\u000d"	Carriage Return
"\u0020"	Space
"\u00a0"	No-Break Space
"\u1680"	Ogham Space Mark
"\u180e"	Mongolian Vowel Separator
"\u2000"	En Quad
"\u2001"	Em Quad
"\u2002"	En Space
"\u2003"	Em Space
"\u2004"	Three-Per-Em Space
"\u2005"	Four-Per-Em Space
"\u2006"	Six-Per-Em Space
"\u2007"	Figure Space
"\u2008"	Punctuation Space
"\u2009"	Thin Space
"\u200a"	Hair Space
"\u202f"	Narrow No-Break Space
"\u205f"	Medium Mathematical Space
"\u3000"	Ideographic Space

2.5.3.167 Trunc

The **Trunc** function performs a truncation operation.

ABNF:

```
Trunc = "TRUNC(" val "," val ")"
```

Required Arguments:

Name: **Number**

Type: vDouble

An argument that specifies a number to truncate.

Name: **Digits**

Type: vSignedInt

An argument that specifies the decimal place to use for the truncation operation.

Return Value:

Type: vNum

This function returns a vNum custom token grouping containing the truncated value of **Number**, as described by the following table.

Condition	Result
Digits > 0	The function returns **Number** truncated to **Digits** places to the right of the decimal point.
Digits = 0	The function returns **Number** truncated to an integer.
Digits < 0	The function returns **Number** truncated to negative **Digits** places to the left of the decimal point.

The unit of the return value is equal to the unit of **Number**. If **Digits** is less than -308 or greater than 15, the function returns **Number**.

2.5.3.168 UMinus

The **UMinus** function performs the negation calculation.

ABNF:

```
UMinus = "_UMINUS(" val ")" / "-" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: vNum

This function returns a vNum custom token grouping containing the negation of the value of **Arg1**. The unit of the return value is equal to the unit of **Arg1**.

2.5.3.169 UniChar

The **UniChar** function returns a Unicode character.

ABNF:

```
Unichar = "UNICHAR(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vUnsignedInt

An argument that specifies a number.

Return Value:

Type: PtgString, PtgErr

This function returns the Unicode character of **Arg1**. If **Arg1** is less than one and greater than 65535, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.170 UPlus

The **UPlus** function performs an identity calculation.

ABNF:

```
UPlus = "_UPLUS(" val ")" / "+" val
```

Required Arguments:

Name: **Arg1**

Type: vDouble

An argument that specifies the operand of the calculation.

Return Value:

Type: vNum

This function returns a vNum custom token grouping containing the value of **Arg1**. The unit of the return value is equal to the unit of **Arg1**.

2.5.3.171 Upper

The **Upper** function performs an upper case conversion.

ABNF:

```
Upper = "UPPER(" val ")"
```

Required Arguments:

Name: **Arg1**

Type: vString

An argument that specifies a string to convert.

Return Value:

Type: PtgString

This function returns a PtgString parse token containing the string converted to upper case. The conversion is performed according to .NET globalization rules based on the value specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing. For more information about .NET globalization rules, see [MSDN-ENCLOC].

2.5.3.172 Use

The **Use** function returns a value of 254.

ABNF:

```
Use = "USE(" val ")"
```

Required Arguments:

Name: **Master**

Type: vString

An argument that specifies a name or **GUID** of a master.

Return Value:

Type: PtgInt

This function returns a PtgInt parse token with a value of 254.

2.5.3.173 Version

The **Version** function returns a **AppVersion** property of a web drawing.

ABNF:

```
Version = "VERSION()"
```

Return Value:

Type: PtgInt

This function returns the integer part of the **AppVersion** property, specified in [ISO/IEC29500-1:2011] section 22.2.2.2, from the App XML part of a web drawing.

2.5.3.174 WeekDay

The **WeekDay** function returns a day of the week, according to the Gregorian calendar, from a value representing a date and time.

ABNF:

```
WeekDay = "WEEKDAY(" val [","  val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a date and time value.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgNum, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion is successful, the function returns a PtgNum parse token containing the weekday component from **DateTimeArg.** If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.3.175 Year

The **Year** function returns a year, according to the Gregorian calendar, from a value representing a date and time.

ABNF:

```
Year = "YEAR(" val [","  val] ")"
```

Required Arguments:

Name: **DateTimeArg**

Type: vAny

An argument that specifies a date and time value.

Optional Arguments:

Name: **Locale**

Type: vLanguageID

An argument that specifies an LCID to use when parsing **DateTimeArg**. The default value is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Return Value:

Type: PtgNum, PtgErr

This function attempts a conversion, as described by the DateTime function, of **DateTimeArg** to a PtgDate parse token. If the conversion is successful, the function returns a PtgNum parse token containing the year component from **DateTimeArg**. If the conversion fails, the function returns a PtgErr parse token with an error code equal to #VALUE!.

2.5.4 Parse Token Definitions

The following parse token definition sections specify the tokens that can be part of a formula expression and that can be persisted in a web drawing in Cell_Type elements.

The definition of a token specifies an ABNF [RFC5234] grammar and a Cell_Type representation. The ABNF specifies the format of the token in a formula expression. The Cell_Type representation specifies the properties that define the token, as described in the operand tokens section.

2.5.4.1 PtgAcre

The **PtgAcre** structure specifies a Unit Number with a unit of acres.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgAcre = double-value "ACRE"
```

double-value is a double precision floating-point number expressed as acres.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as square inches.
U	MUST be "AC".

2.5.4.2 PtgAngDD

The **PtgAngDD** structure specifies a Unit Number with a unit of decimal degrees.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgAngDD = double-value "DEG"
```

double-value is a double precision floating-point number expressed as decimal degrees.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as an angleInternalUnitNumber.
U	MUST be "DEG".

2.5.4.3 PtgAngDft

The **PtgAngDft** structure specifies a Unit Number with a unit of radians.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgAngDft = double-value "DA"
```

double-value is a double precision floating-point number expressed as radians.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as an angleInternalUnitNumber.
U	MUST be "DA".

2.5.4.4 PtgAngDMS

The **PtgAngDMS** structure specifies a Unit Number with a unit of degrees-minutes-seconds.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgAngDMS = unsigned-int-value %xB0 [unsigned-int-value "&apos;" [double-value %x22]]
```

The first **unsigned-int-value** is an unsigned integer expressed as degrees.

The second **unsigned-int-value** is an unsigned integer expressed as minutes.

double-value is a double precision floating-point number expressed as seconds.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as an angleInternalUnitNumber.
U	MUST be "AD".

2.5.4.5 PtgAngRad

The **PtgAngRad** structure specifies a Unit Number with a unit of radians.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgAngRad = double-value "RAD"
```

double-value is a double precision floating-point number expressed as radians.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as an angleInternalUnitNumber.
U	MUST be "RAD".

2.5.4.6 PtgBool

The **PtgBool** structure specifies a **Boolean** value.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgBool = bool-value
```

bool-value specifies a Boolean.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be zero if the value of the structure is equal to **FALSE** and MUST be one if value of the structure is equal to **TRUE**.
U	MUST NOT exist or MUST be "BOOL".

2.5.4.7 PtgByte

The **PtgByte** structure specifies an unsigned byte value.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgByte = unsigned-byte-value
```

unsigned-byte-value specifies an unsigned byte.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be greater than or equal to zero, and less than 256.
U	MUST NOT exist or MUST be "NUM".

2.5.4.8 PtgColorRGB

The **PtgColorRGB** structure specifies an RGB color value represented as a 3-byte signed integer.

ABNF:

```
PtgColorRGB = color-value
```

The most significant byte of **color-value** specifies the intensity of the color red. The second most significant byte of **color-value** specifies the intensity of the color green. The least significant byte of **color-value** specifies the intensity of the color blue.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be a value that satisfies the ABNF.
U	MUST NOT exist, or MUST be "COLOR".

2.5.4.9 PtgCy

The **PtgCy** structure specifies a currency value.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgCy = double-value string-value
```

double-value specifies a double precision floating-point number expressed as a value of the currency specified by **string-value**.

string-value specifies a string that contains a currency. It MUST be a value defined by **vCurrency**.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be a value that satisfies the ABNF.
U	MUST be "CY".

2.5.4.10 PtgDate

The **PtgDate** structure specifies a Unit Number with a unit of time.

This structure MUST NOT appear in the ABNF of a formula expression. It can be the result of a formula evaluation.

When stored as a Cell_Type, this structure has the following attribute values.

Attribute	Value
V	MUST be a date and time of day, in complete extended format, as specified in [ISO-8601] section 4.3.2.
U	MUST be "DATE".

2.5.4.11 PtgEDay

The **PtgEDay** structure specifies a Unit Number with a unit of days.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgEDay = double-value "ED"
```

double-value is a double precision floating-point number expressed as days.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a durationInternalUnitNumber.
U	MUST be "ED".

2.5.4.12 PtgEHour

The **PtgEHour** structure specifies a Unit Number with a unit of hours.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgEHour = double-value "EH"
```

double-value is a double precision floating-point number expressed as hours.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a durationInternalUnitNumber.
U	MUST be "EH".

2.5.4.13 PtgEMin

The **PtgEMin** structure specifies a Unit Number with a unit of minutes.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgEMin = double-value "EM"
```

double-value is a double precision floating-point number expressed as minutes.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a durationInternalUnitNumber.
U	MUST be "EM".

2.5.4.14 PtgErr

The **PtgErr** structure specifies an error code.

This structure MUST NOT appear in the ABNF of a formula expression. It can be the result of a formula evaluation.

When stored in a Cell_Type element, this structure has an **E** attribute that MUST be one of the values from the following table.

Value	Meaning
#DIM!	A dimensional value that exceeds the dimension range.
#DIV/0!	Division by zero.
#VALUE!	An argument or operand of the wrong type.
#REF!	A reference to a cell that does not exist.
#NUM!	An invalid number.
#N/A	Not available value.

2.5.4.15 PtgESec

The **PtgESec** structure specifies a Unit Number with a unit of seconds.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgESec = double-value "ES"
```

double-value is a double precision floating-point number expressed as seconds.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a durationInternalUnitNumber.
U	MUST be "ES".

2.5.4.16 PtgEWeek

The **PtgEWeek** structure specifies a Unit Number with a unit of weeks.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgEWeek = double-value "EW"
```

double-value is a double precision floating-point number expressed as weeks.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a durationInternalUnitNumber.
U	MUST be "EW".

2.5.4.17 PtgHectare

The **PtgHectare** structure is a Unit Number with a unit of hectares.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgHectare = double-value "HA"
```

double-value is a double precision floating-point number expressed as hectares.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as square inches.
U	MUST be "HA".

2.5.4.18 PtgInt

The **PtgInt** structure specifies a signed four-byte integer value.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgInt = int-value
```

int-value specifies a signed four-byte integer.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be greater than or equal to -2^{31} and be less than 2^{31}.
U	MUST NOT exist or MUST be "NUM".

2.5.4.19 PtgNum

The **PtgNum** structure specifies a double precision floating-point number.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNum = double-value
```

double-value is a double precision floating-point number.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the value of the structure expressed as a double precision floating-point number.
U	MUST NOT exist.

2.5.4.20 PtgNumCM

The **PtgNumCM** structure specifies a Unit Number with a unit of centimeters.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumCM = double-value "CM"
```

double-value is a double precision floating-point number expressed as centimeters.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a **lengthInternalUnitNumber** (section 2.5.8.3).
U	MUST be "CM".

2.5.4.21 PtgNumDft

The **PtgNumDft** structure specifies a Unit Number with a unit of inches.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumDft = double-value "DL"
```

double-value is a double precision floating-point number expressed as inches.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a lengthInternalUnitNumber (section 2.5.8.3).
U	MUST be "DL".

2.5.4.22 PtgNumF

The **PtgNumF** structure specifies a Unit Number with a unit of feet.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumF = double-value "FT"
```

double-value is a double precision floating-point number expressed as feet.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a **lengthInternalUnitNumber** (section 2.5.8.3).
U	MUST be "FT".

2.5.4.23 PtgNumFI

The **PtgNumFI** structure specifies a Unit Number with a unit of feet and inches.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumFI = unsigned-int-value "FT" [SP double-value "IN"]
```

unsigned-int-value is an unsigned integer expressed as feet.

double-value is a double precision floating-point number expressed as inches.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a **lengthInternalUnitNumber** (section 2.5.8.3).
U	MUST be "F_I".

2.5.4.24 PtgNumI

The **PtgNumI** structure specifies a Unit Number with a unit of inches.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumI = double-value "IN"
```

double-value is a double precision floating-point number expressed as inches.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a **lengthInternalUnitNumber** (section 2.5.8.3).
U	MUST be "IN" or "IN_F".

2.5.4.25 PtgNumKM

The **PtgNumKM** structure specifies a Unit Number with a unit of kilometers.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumKM = double-value "KM"
```

double-value is a double precision floating-point number expressed as kilometers.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a **lengthInternalUnitNumber** (section 2.5.8.3).

Attribute	Value
U	MUST be "KM".

2.5.4.26 PtgNumM

The **PtgNumM** structure specifies a Unit Number with a unit of meters.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumM = double-value "M"
```

double-value is a double precision floating-point number expressed as meters.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a **lengthInternalUnitNumber** (section 2.5.8.3).
U	MUST be "M".

2.5.4.27 PtgNumMI

The **PtgNumMI** structure specifies a Unit Number with a unit of miles.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumMI = double-value "MI"
```

double-value is a double precision floating-point number expressed as miles.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a **lengthInternalUnitNumber** (section 2.5.8.3).
U	MUST be "MI" or "MI_F".

2.5.4.28 PtgNumMM

The **PtgNumMM** structure specifies a Unit Number with a unit of millimeters.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumMM = double-value "MM"
```

double-value is a double precision floating-point number expressed as millimeters.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a **lengthInternalUnitNumber** (section 2.5.8.3).
U	MUST be "MM".

2.5.4.29 PtgNumMultiDim

The **PtgNumMultiDim** structure specifies a Unit Number that is multidimensional.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumMultiDim = double-value string-value "^" dimension
```

double-value is a double precision floating-point number expressed as the unit indicated by **string-value**.

string-value is a string that specifies the unit of measurement, and MUST be a value from the following table.

Value	Unit
DEG	Degrees
DA	Radians
RAD	Radians
DATE	Days
ED	Days
EH	Hours
EM	Minutes
ES	Seconds
EW	Weeks
CM	Centimeters
DL	Inches

Value	Unit
FT	Feet
IN	Inches
KM	Kilometers
MM	Millimeters
M	Meters
MI	Miles
NM	Nautical miles
YD	Yards
DP	Inches
DE	Days
C	Ciceros
D	Didots
DT	Points
P	Picas
PT	Points

dimension is an integer that specifies the number of dimensions.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be a value that satisfies the **ABNF**.
U	MUST be "MULTIDIM".

2.5.4.30 PtgNumNM

The **PtgNumNM** structure specifies a Unit Number with a unit of nautical miles.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumNM = double-value "NM"
```

double-value is a double precision floating-point number expressed as nautical miles.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a lengthInternalUnitNumber.
U	MUST be "NM".

2.5.4.31 PtgNumPct

The **PtgNumPct** structure specifies a Unit Number expressed as a percentage.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumPct = double-value "%"
```

double-value is a double precision floating-point number.

Attribute	Value
V	MUST be the double precision floating-point number double-value. The value is normalized such that the value of 1 corresponds to 100 percent.
U	MUST be "PER".

2.5.4.32 PtgNumYards

The **PtgNumYards** structure specifies a Unit Number with a unit of yards.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgNumYards = double-value "YD"
```

double-value is a double precision floating-point number expressed as yards.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a lengthInternalUnitNumber.
U	MUST be "YD".

2.5.4.33 PtgPageDft

The **PtgPageDft** structure specifies a Unit Number with a unit of inches.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgPageDft = double-value "DP"
```

double-value is a double precision floating-point number expressed as inches.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a lengthInternalUnitNumber.
U	MUST be "DP".

2.5.4.34 PtgPnt

The **PtgPnt** token specifies the coordinates of a two-dimensional point.

When found in a formula expression, this structure is identical to the Pnt function token, and MUST satisfy the following ABNF.

ABNF:

```
PtgPnt = "PNT(" val "," val ")"
```

The first **val** specifies a double precision floating-point number expressed as an x-coordinate.

The second **val** specifies a double precision floating-point number expressed as a y-coordinate.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be a value that satisfies the following ABNF. `PtgPnt = "PNT(" double-value "," double-value ")"` The first **double-value** specifies a double precision floating-point number expressed as an x-coordinate. The second **double-value** specifies a double precision floating-point number expressed as a y-coordinate.
U	MUST be "PNT".

2.5.4.35 PtgShort

The **PtgShort** structure specifies a signed two-byte integer value.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgShort = short-value
```

short-value specifies a signed two-byte integer.

When stored in a Cell Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be greater than or equal to -32768, and be less than 32768.
U	MUST NOT exist or MUST be "NUM".

2.5.4.36 PtgString

The **PtgString** structure specifies a string.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgString = DQUOTE string-value DQUOTE
```

When stored in a Cell Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the string-value (section 2.5.1).
U	MUST be "STR".

2.5.4.37 PtgTDurDft

The **PtgTDurDft** structure specifies a Unit Number with a unit of days.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgTDurDft = double-value "DE"
```

double-value specifies a double precision floating-point number expressed as days.

When stored in a Cell Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a durationInternalUnitNumber (section 2.5.8.2).
U	MUST be "DE".

2.5.4.38 PtgTypCD

The **PtgTypCD** structure specifies a Unit Number with a unit of ciceros and didots.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgTypCD = unsigned-int-value "C" [double-value]
```

unsigned-int-value is an unsigned integer expressed as ciceros.

double-value is a double precision floating-point number expressed as didots.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a typographicInternalUnitNumber (section 2.5.8.4).
U	MUST be "C_D".

2.5.4.39 PtgTypCi

The **PtgTypCi** structure specifies a Unit Number with a unit of ciceros.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgTypCi = double-value "C"
```

double-value is a double precision floating-point number expressed as ciceros.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a typographicInternalUnitNumber.
U	MUST be "C".

2.5.4.40 PtgTypDft

The **PtgTypDft** structure specifies a Unit Number with a unit of points.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgTypDft = double-value "DT"
```

double-value is a double precision floating-point number expressed as points.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a typographicInternalUnitNumber (section 2.5.8.4).
U	MUST be "DT".

2.5.4.41 PtgTypDi

The **PtgTypDi** structure specifies a Unit Number with a unit of didots.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgTypDi = double-value "D"
```

double-value is a double precision floating-point number expressed as didots.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a typographicInternalUnitNumber (section 2.5.8.4).
U	MUST be "D".

2.5.4.42 PtgTypPi

The **PtgTypPi** structure specifies a Unit Number with a unit of picas.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgTypPi = double-value "P"
```

double-value is a double precision floating-point number expressed as picas.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as an typographicInternalUnitNumber (section 2.5.8.4).
U	MUST be "P".

2.5.4.43 PtgTypPP

The **PtgTypPP** structure specifies a Unit Number with a unit of picas and points.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgTypPP = unsigned-int-value "P" [double-value]
```

unsigned-int-value is an unsigned integer expressed as picas.

double-value is a double precision floating-point number expressed as points.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a **typographicInternalUnitNumber** (section 2.5.8.4).
U	MUST be "P_PT".

2.5.4.44 PtgTypPt

The **PtgTypPt** structure specifies a Unit Number with a unit of points.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgTypPt = double-value "PT"
```

double-value is a double precision floating-point number expressed as points.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be the numeric value of the structure expressed as a typographicInternalUnitNumber (section 2.5.8.4).
U	MUST be "PT".

2.5.4.45 PtgUnsShort

The **PtgUnsShort** structure specifies a two-byte unsigned integer value.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
PtgUnsShort = unsigned-int-value
```

unsigned-int-value specifies a two-byte unsigned integer.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be greater than or equal to zero, and be less than 65536.
U	MUST NOT exist or MUST be "NUM".

2.5.4.46 PtgNurbs

The **PtgNurbs** structure specifies a non-uniform rational B-spline (NURBS).

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
Nurbs = "NURBS(" knotLast "," degree "," xType "," yType *("," xN "," yN "," knotN ","
weightN) ")"
knotLast = val
degree = val
xType = val
yType = val
xN = val
yN = val
knotN = val
weightN = val
```

knotLast is a double precision floating-point number that specifies the last knot.

degree is an integer value that specifies the degree of the B-spline.

xType is an unsigned integer value that specifies how to interpret the x-coordinates. If **xType** is zero, the input **xN** is interpreted as relative coordinates. Otherwise, the input **xN** is interpreted in the coordinate system of the shape containing the structure.

yType is an unsigned integer value that specifies how to interpret the y-coordinates. If **yType** is zero, the input **yN** is interpreted as relative coordinates. Otherwise, the input **yN** is interpreted in the coordinate system of the shape containing the structure.

xN is a double precision floating-point number that specifies an x-coordinate.

yN is a double precision floating-point number that specifies a y-coordinate.

knotN is a double precision floating-point number that specifies a knot on the B-spline.

weightN is a double precision floating-point number that specifies a weight on the B-spline.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value

V	MUST be a value that satisfies the following ABNF.
	
	`Nurbs = "NURBS(" knotLast "," degree "," xType "," yType *("," xN "," yN "," knotN "," weightN) ")"`
	
	`knotLast = double-value`
	`degree = int-value`
	`xType = unsigned-int-value`
	`yType = unsigned-int-value`
	`xN = double-value`
	`yN = double-value`
	`knotN = double-value`
	`weightN = double-value`
	
	Specific definition is provided earlier.

U	MUST NOT exist.

2.5.4.47 PtgPolyLine

The **PtgPolyLine** structure specifies a polyline.

When found in a formula expression, this structure MUST satisfy the following ABNF.

ABNF:

```
Polyline = "POLYLINE(" xType "," yType *("," xN "," yN) ")"

xType = val
yType = val
xN = val
yN = val
```

xType is an unsigned integer value that specifies how to interpret the x-coordinates. If **xType** is zero, the input **xN** is interpreted as relative coordinates. Otherwise, the input **xN** is interpreted in the coordinate system of the shape containing the structure.

yType is an unsigned integer value that specifies how to interpret the y-coordinates. If **yType** is zero, the input **yN** is interpreted as relative coordinates. Otherwise, the input **yN** is interpreted in the coordinate system of the shape containing the structure.

xN is a double precision floating-point number that specifies an x-coordinate.

yN is a double precision floating-point number that specifies a y-coordinate.

When stored in a Cell_Type element, this structure has the following attribute values.

Attribute	Value
V	MUST be a value that satisfies the following ABNF. `Polyline = "POLYLINE(" xType "," yType *("," xN "," yN) ")"`

Attribute	Value

	`xType = unsigned-int-value`
	`yType = unsigned-int-value`
	`xN = double-value`
	`yN = double-value`
	
	Specific definition is provided earlier.

U	MUST be "POLYLINE".

2.5.5 Reference Token Definitions

The reference token definitions in the following sections specify the reference tokens that can be contained in a formula expression.

2.5.5.1 CellRef

The **CellRef** token specifies a reference to a cell in a reference context.

ABNF:

```
cellRef = [sheetref "!"] ( name / NamedCellRef / IndexedCellRef / SingleLetterNamedCellRef )

sheetref = CrossPageRef / DocSheetRef / MasterSheetRef / PageSheetRef / ShapeSheetRef /
StyleSheetRef
```

name is a name of a cell in the web drawing. It MUST be equal to the **N** attribute of a Cell_Type element that is a child element of the Sheet_Type element in the current reference context or a reference context specified by **sheetref**.

2.5.5.2 CrossPageRef

The **CrossPageRef** token specifies a reference to the shape sheet of a shape on a drawing page. It also changes the current reference context to the specified drawing page.

ABNF:

```
CrossPageRef = "pages[" string-value "]!" ShapeSheetRef
```

string-value is a name of a drawing page. It MUST be equal to the **NameU** attribute of a Page_Type element in the web drawing.

ShapeSheetRef is a reference to a shape sheet in the drawing page specified by string-value.

2.5.5.3 DocSheetRef

The **DocSheetRef** token specifies a reference to the document sheet of a web drawing. It also changes the current reference context to the document sheet.

ABNF:

```
DocSheetRef = "thedoc"
```

2.5.5.4 IndexedCellRef

The **IndexedCellRef** token specifies a reference to a cell in the current reference context.

ABNF:

```
IndexedCellRef = SectionRef "." name [ "[" id "]" ]
```

name is the name of a cell in the web drawing. It MUST be equal to the **N** attribute of Cell_Type element that is a child element of the Row_Type element that is specified by **id**.

id is the index of a row in the web drawing. It MUST be equal to the value of the **IX** attribute plus one of a Row_Type element that is a child element of the Section_Type element that is specified by **SectionRef**. If **id** is omitted, the value of one is used.

2.5.5.5 MasterSheetRef

The **MasterSheetRef** token specifies a reference to the shape sheet of a shape in a master. It also changes the current reference context to the specified master.

ABNF:

```
MasterSheetRef = "masters[" string-value "]!" ShapeSheetRef
```

string-value is the name of a master in the web drawing. It MUST be equal to the **NameU** attribute of a Master_Type element in the web drawing.

ShapeSheetRef is a reference to a shape sheet in the master specified by **string-value**.

2.5.5.6 NamedCellRef

The **NamedCellRef** token specifies a reference to a cell in the current reference context.

ABNF:

```
NamedCellRef = SectionRef "." RowName [ "." CellName ]
RowName = name
CellName = name
```

RowName is the name of a row in the web drawing. It MUST be equal to the **N** attribute of a Row_Type element that is a child element of the Section_Type element that is specified by **SectionRef**.

CellName is the name of a cell in the web drawing. It MUST be equal to the **N** attribute of a Cell_Type element that is a child element of the Row_Type element that is specified by **RowName**.

If **CellName** is omitted, the cell is specified in the following table based on the value of **SectionRef**. **SectionRef** MUST be one of values in the following table.

SectionRef value	Default cell name
"User"	Value
"Prop"	Value
"Actions"	Menu
"Controls"	X
"Hyperlink"	Description
"SmartTags"	X

2.5.5.7 PageSheetRef

The **PageSheetRef** token specifies a reference to the page sheet of the drawing page in the current reference context. It also changes the current reference context to the page sheet.

ABNF:

```
PageSheetRef = "thepage"
```

2.5.5.8 SectionRef

The **SectionRef** token specifies a reference to a section in the current reference context.

ABNF:

```
SectionRef = "Geometry" id / "Actions" / "Char" / "Connections" / "Controls" / "Fields" /
"FillGradientStops" / "Hyperlink" / "Layers" / "LineGradientStops" / "Para" / "Prop" /
"Reviewer" / "Scratch" / "SmartTags" / "Tabs" / "User" / name
```

id is the index of a Geometry section in the web drawing. It MUST be equal to the value of the **IX** attribute plus one of a Section_Type element.

name is the name of a section in the Extension part of the web drawing. It MUST be equal to **N** attribute of the one of SectionDef_Type elements.

Other **SectionRef** values specify a reference to a section according to the following table.

ABNF	Section
"Actions"	Actions
"Char"	Character
"Connections"	Connection
"Controls"	Control
"Fields"	Field
"FillGradientStops"	FillGradient

ABNF	Section
"Geometry"	Geometry
"Hyperlink"	Hyperlink
"Layers"	Layer
"LineGradientStops"	LineGradient
"Para"	Paragraph
"Prop"	Property
"Reviewer"	Reviewer
"Scratch"	Scratch
"SmartTags"	ActionTag
"Tabs"	Tabs
"User"	User

2.5.5.9 ShapeSheetRef

The **ShapeSheetRef** token specifies a reference to the shape sheet of a shape in the current reference context.

ABNF:

```
ShapeSheetRef = nameid / "sheet." id
```

nameid is the identifier of a shape sheet. It MUST be equal to the **NameU** attribute of a ShapeSheet_Type element in the current reference context.

id is the identifier of a shape sheet. It MUST be equal to the **ID** attribute of a ShapeSheet_Type element in the current reference context.

2.5.5.10 SingleLetterNamedCellRef

The **SingleLetterNamedCellRef** token specifies a reference to a cell in the current reference context.

ABNF:

```
SingleLetterNamedCellRef = SectionRef "." SingleLetterCellName id
SingleLetterCellName = ALPHA
```

id is the index of a row in the web drawing. It MUST be equal to the **IX** attribute plus one of a Row_Type element that is a child element of the Section_Type element specified by **SectionRef**.

SingleLetterCellName is the name of a cell in the row specified by **id**. It MUST be equal to the **N** attribute of a Cell_Type element that is a child element of the Row_Type element that is specified by **id**.

2.5.5.11 StyleSheetRef

The **StyleSheetRef** token specifies a reference to a style sheet in a web drawing. It also changes the current reference context to the specified style sheet.

ABNF:

```
StyleSheetRef = "styles!" string-value
```

string-value is the name of a style. It MUST be equal to the **NameU** attribute of the StyleSheet_Type element in the web drawing.

2.5.6 Custom Input Type Definitions

The custom input definitions in the following sections specify custom input types. Custom input types are used to specify the token types for function arguments. Functions that require a particular class of token inputs, and thus require a conversion from many possible source token types, declare arguments with custom input types and rely on the custom input type to derive a valid input argument from a source token. The definition of each custom input type specifies how the source token is derived, the computed properties of the custom input type, and under which cases the input argument is not valid.

2.5.6.1 vBoolean

The **vBoolean** custom input type specifies a PtgBool parse token that is derived from a source token that MUST be a vNumAny custom token grouping or a PtgString or PtgCy parse token. It contains the following computed property.

Value: This property specifies the **Value** of a Boolean value derived from the source token as follows.

If the source token is a PtgBool, the **Value** is equal to the source token **Value**. If the source token is a PtgString that contains a formula that evaluates to **FALSE** or zero, the **Value** is **FALSE**; if the source token is a PtgString that contains a formula that evaluates to **TRUE** or a value other than zero, the **Value** is **TRUE**. Otherwise, the input argument is not valid.

If the source token is a vNum or PtgCy with a **Value** equal to zero, the **Value** is **FALSE**; otherwise, **Value** is **TRUE**.

2.5.6.2 vColor

The **vColor** custom input type specifies a PtgColorRGB parse token derived from a source token that MUST be a vNumAny custom token grouping or a PtgColorRGB, PtgString, or PtgCy parse token. It contains the following computed property.

Value: This property specifies the **Value** of a color value represented as a 3-byte signed integer derived from the source token **Value** as follows.

If the source token type is a PtgColorRGB, the **Value** is equal to the source token **Value**.

If the source token can be interpreted as a vSignedLong custom input type, the **Value** is equal to the color in the color table that is indexed by the source token **Value** interpreted as a vSignedLong; if the source token **Value** interpreted as a vSignedLong is not a valid index in the color table, the **Value** is equal to the color at index zero in the color table.

In all other cases, the input argument is not valid.

2.5.6.3 vDouble

The **vDouble** custom input type specifies a vNum custom token grouping derived from a source token that MUST be a vNumAny custom token grouping or a PtgString, or PtgCy parse token. It contains the following computed properties.

Value: This property specifies the **Value** of a numeric value or date value, derived from the source token **Value** as follows.

If the source token is a PtgDate parse token, the **Value** is equal to the source token **Value**.

If the source token **Value** can be converted to a double, as described in [MSDN-ToDouble], the **Value** is the result of the conversion.

If the source token is a PtgString containing a numeric value, or the concatenation of a numeric value and a unit and the numeric value can be converted to a double, as described in [MSDN-ToDouble], the **Value** is the result of the conversion.

If the source token **Value** is equal to **TRUE** (case insensitive), the **Value** is equal to one. If the source token **Value** is equal to **FALSE** (case insensitive), the **Value** is equal to zero.

In all other cases, the input argument is not valid.

Unit: This property specifies the **Unit** of the numeric value or date value. If the source token is a vUnitType, the **Unit** is equal to the **Unit** of the source token. If the source token is a PtgString containing the concatenation of a numeric value and a unit, the **Unit** is equal to the unit derived from the PtgString. Otherwise, the numeric value does not have a **Unit**.

Dimension: This property specifies the **Dimension** of the numeric value or date value. If the source token type is a PtgNumMultiDim parse token, the **Dimension** is equal to the dimension of the source token. If the source token is a PtgAcre or PtgHectare parse token, the **Dimension** is equal to two. If the source token is any other vUnitType custom token grouping, the **Dimension** is equal to one. Otherwise, **Dimension** is equal to zero.

2.5.6.4 vDoubleEx

The **vDoubleEx** custom input type specifies either a vNum custom token grouping or a PtgCy parse token derived from a source token that MUST be a vNumAny custom token grouping or a PtgString or PtgCy parse token. The **vDoubleEx** is the vDouble custom input type with the following exceptions and additional computed properties.

Unit: This property specifies the **Unit** of a numeric value, date value, or currency value. If the source token is a PtgCy, the property specifies the **Unit** of a currency value, which is defined as "CY". Otherwise, **Unit** is equal to the type of the source token interpreted as a vDouble.

Currency: This property specifies the **Currency** of a currency value. If the source token is a PtgCy, the **Currency** is equal to the **Currency** of the source token. Otherwise, it does not have a **Currency** and the custom input type specifies a vNum.

2.5.6.5 vFloat

The **vFloat** custom input type specifies a single precision floating-point number derived from a source token that MUST be a vNumAny custom token grouping or a PtgString or PtgCy parse token. It is a vDouble custom input type with the additional restriction that the **Value** property MUST conform to the range of values specified for a single precision floating-point number, as defined in [IEEE754].

If the source token is a PtgDate parse token, the date format specified in [MS-OAUT] section 2.2.25 is used to determine whether the range of the date value conforms to a single precision floating-point number, as defined in [IEEE754].

If the source token is not a vDouble or the source token **Value** interpreted as a vDouble does not conform to the range of values specified for a single precision floating-point number, the input argument is not valid.

2.5.6.6 vSignedInt

The **vSignedInt** custom input type specifies a signed integer derived from a source token that MUST be a vNumAny custom token grouping or a PtgString, or PtgCy parse token. It is a vSignedLong custom input type with the additional restriction that the **Value** property MUST be greater than or equal to -32768 and less than or equal to 32767.

If the source token is not a vSignedLong or the source token **Value** interpreted as a vSignedLong is less than -32768 or greater than 32767, the input argument is not valid.

2.5.6.7 vSignedLong

The **vSignedLong** custom input type specifies a signed long integer derived from a source token that MUST be a vNumAny custom token grouping or a PtgString or PtgCy parse token. It contains the following computed properties.

Value: This property specifies a signed long integer derived from the source token as follows.

If the source token **Value** can be converted to a double, as described in [MSDN-ToDouble] or as specified in [MS-OAUT] section 2.2.25 for a date value, the **Value** is equal to a signed long integer calculated as follows:

Use the conversion method described in [MSDN-ToDouble] or [MS-OAUT] section 2.2.25 to obtain a double value. If the double value is not an integer, round the value towards zero to the next integer. If the integer is less than -2147483648 or greater than 2147483647, the input argument is not valid.

If the source token **Value** is equal to **TRUE** (case insensitive), the **Value** is equal to one. If the source token **Value** is equal to **FALSE** (case insensitive), the **Value** is equal to zero.

In all other cases, the input argument is not valid.

Unit: This property specifies the **Unit** of the operand token and this property has no value for this custom input type.

2.5.6.8 vString

The **vString** custom input type specifies a PtgString parse token derived from a source token that MUST be a vNumAny custom token grouping or a PtgString. It contains the following computed property.

Value: This property specifies a string derived from the source token **Value** as follows.

If the source token is a PtgString, the **Value** is equal to the source token **Value**.

If the source token is a vNumAny, the **Value** is the string form of the source token.

In all other cases, the input argument is not valid.

2.5.6.9 vUnsignedInt

The **vUnsignedInt** custom input type specifies an unsigned integer derived from a source token that MUST be a vNumAny custom token grouping or a PtgString or PtgCy parse token. It is a vUnsignedLong custom input type with the additional restriction that **Value** MUST be less than 65536.

If the source token **Value** interpreted as a custom input type of vSignedLong is less than 65536, **Value** is equal to the source token **Value** interpreted as a vSignedLong. Otherwise, **Value** is equal to the source token **Value** interpreted as a vSignedLong modulo 65536.

2.5.6.10 vUnsignedLong

The **vUnsignedLong** custom input type specifies an unsigned long integer derived from a source token that MUST be a vNumAny custom token grouping or a PtgString or PtgCy parse token. It contains the following computed properties.

Value: This property specifies an unsigned long integer derived from the source token **Value** as follows.

If the source token **Value** can be converted to a double, as described in [MSDN-ToDouble] or as specified in [MS-OAUT] section 2.2.25 for a date value, the **Value** is equal to an unsigned long integer calculated as follows.

Use the conversion method described in [MSDN-ToDouble] or [MS-OAUT] section 2.2.25 to obtain a double value. If the double value is not an integer, round the value towards zero to the next integer. If the integer is greater than or equal to zero, the **Value** is equal to the integer modulo 4294967296. Otherwise, **Value** is equal to the sum of the integer modulo 4294967296 and 4294967296.

If the source token **Value** is equal to **TRUE** (case insensitive), the **Value** is equal to one. If the source token value is equal to **FALSE** (case insensitive), the **Value** is equal to zero.

In all other cases, the input argument is not valid.

Unit: This property specifies the **Unit** of the operand token and this property has no value for this custom input type.

2.5.7 Custom Token Groupings

The custom token groupings in the following sections define groupings of tokens, referred to as custom token groupings. A custom token grouping specifies a set of tokens that represents a specific concept. Custom token groupings can contain other custom token groupings. The association of a token to a custom token grouping is not exclusive. These groupings do not exist in the format of the file. The purpose of these groupings is to improve the readability of the document by allowing tokens representing similar concepts to be referred to collectively.

2.5.7.1 vAngle

The **vAngle** custom token grouping is an aggregation of types that represent an angle. The **vAngle** contains the following tokens:

PtgAngDD, PtgAngDft, PtgAngDMS, PtgAngRad

2.5.7.2 vAny

The **vAny** custom token grouping is an aggregation of types that represent data. The **vAny** contains the following tokens:

vNumAny, PtgString, PtgColorRGB, PtgCy, PtgPnt, PtgErr, PtgNURBS, PtgPolyline

2.5.7.3 vLength

The **vLength** custom token grouping is an aggregation of types that can be used to represent a length, including a distance, duration, or typography measurement. The value of any type in this grouping MUST be interpreted as a lengthInternalUnitNumber. The **vLength** contains the following tokens:

PtgEDay, PtgEHour, PtgEMin, PtgESec, PtgEWeek, PtgNumCM, PtgNumDft, PtgNumF, PtgNumFI, PtgNumI, PtgNumKM, PtgNumM, PtgNumMI, PtgNumMM, PtgNumNM, PtgNumYards, PtgPageDft, PtgTDurDft, PtgTypCD, PtgTypCi, PtgTypDft, PtgTypDi, PtgTypPi, PtgTypPP, PtgTypPt, vScalar

2.5.7.4 vNum

The **vNum** custom token grouping is an aggregation of types that represent either a Unit Number or a number with no units. The **vNum** contains the following tokens:

vUnitType, vScalar

2.5.7.5 vNumAny

The **vNumAny** custom token grouping is an aggregation of types that represent any number. The **vNumAny** contains the following tokens:

vNum, PtgBool

2.5.7.6 vScalar

The **vScalar** custom token grouping is an aggregation of types that represent a number with no unit of measure. The **vScalar** contains the following tokens:

PtgByte, PtgInt, PtgNum, PtgShort, PtgUnsShort

2.5.7.7 vUnitType

The **vUnitType** custom token grouping is an aggregation of types that represent a Unit Number. The **vUnitType** contains the following tokens:

vAngle, PtgAcre, PtgDate, PtgEDay, PtgEHour, PtgEMin, PtgESec, PtgEWeek, PtgHectare, PtgNumCM, PtgNumDft, PtgNumF, PtgNumFI, PtgNumI, PtgNumKM, PtgNumM, PtgNumMI, PtgNumMM, PtgNumMultiDim, PtgNumNM, PtgNumPct, PtgNumYards, PtgPageDft, PtgTDurDft, PtgTypCD, PtgTypCi, PtgTypDft, PtgTypDi, PtgTypPi, PtgTypPP, PtgTypPt

2.5.8 Custom Internal Unit Types

The custom internal unit types in the following sections specify internal unit types. All internal unit values MUST conform to a double precision floating-point number.

2.5.8.1 angleInternalUnitNumber

The **angleInternalUnitNumber** custom internal unit type specifies a value with a unit of radians.

2.5.8.2 durationInternalUnitNumber

The **durationInternalUnitNumber** custom internal unit type specifies a value with a unit of days.

2.5.8.3 lengthInternalUnitNumber

The **lengthInternalUnitNumber** custom internal unit type specifies a value with a unit of inches.

2.5.8.4 typographicInternalUnitNumber

The **typographicInternalUnitNumber** custom internal unit type specifies a value with a unit of inches.

2.5.9 Custom Structures

The custom structures in the following sections specify custom structures and the set of valid values for these structures.

2.5.9.1 vCalendar

The **vCalendar** custom structure is an unsigned integer that specifies the calendar system to use when formatting dates and times.

It MUST be a value from the following table.

Value	Description
0	Western
1	Arabic Hijri
2	Hebrew Lunar
3	Chinese National
4	Japanese Emperor
5	Thai Buddhist
6	Korean Danki
7	Japanese Saka Era
8	Transliterated English
9	Transliterated French
10	Gregorian US English
11	Gregorian Middle East French
12	Gregorian Arabic
13	Um-al-Qura

2.5.9.2 vCurrency

The **vCurrency** custom structure is a signed integer or a string that specifies a currency.

It MUST be a value from the following table.

ID	String	Meaning
-1		Undefined currency type
0	SYS	System
1	XXX	Omit currency symbol
10	EUR	European Union: euro
11	USD	United States: dollar
12	ATS	Austria: schilling
13	AUD	Australia: dollar
14	BEF	Belgium: franc
15	CAD	Canada: dollar
16	CHF	Switzerland: franc
17	CNY	China (Mainland): yuan
18	DEM	Germany: mark
19	DKK	Denmark: krone
20	ESP	Spain: peseta
21	FIM	Finland: markka
22	FRF	France: franc
23	GBP	United Kingdom: pound
24	GRD	Greece: drachma
25	HKD	Hong Kong SAR: dollar
26	HUF	Hungary: forint
27	IDR	Indonesia: rupiah
28	IEP	Ireland: punt
29	ILS	Israel: shekel
30	ITL	Italy: lira
31	JPY	Japan: yen
32	KRW	Korea, Republic of (South): won
33	LUF	Luxembourg: franc
34	MXN	Mexico: peso
35	MYR	Malaysia: ringgit
36	NLG	Netherlands: guilder
37	NOK	Norway: krone
38	NZD	New Zealand: dollar
39	PHP	Philippines: peso
40	PLZ	Poland: zloty (obsolete, use 89)

ID	String	Meaning
41	PTE	Portugal: escudo
42	ROL	Romania: leu
43	RUR	Russia: rouble (obsolete, use 90)
44	SEK	Sweden: kroner
45	SGD	Singapore: dollar
46	THB	Thailand: baht
47	TWD	Taiwan: dollar
48	XEU	European Currency Unit (ECU)
49	YUN	Yugoslavia: dinar (obsolete, use 91)
50	ZAR	South Africa: rand
56	ARS	Argentina: peso
57	BMD	Bermuda: dollar
58	BOB	Bolivia: boliviano
59	BRR	Brazil: cruzeiro real (obsolete, use 88)
60	BSD	Bahamas: dollar
61	CLP	Chile: peso
62	COP	Colombia: peso
63	CRC	Costa Rica: colon
64	CZK	Czech Republic: koruna
65	DOP	Dominican Republic: peso
66	ECS	Ecuador: sucre
67	EGP	Egypt: pound
68	HNL	Honduras: lempira
69	INR	India: rupee
70	JMD	Jamaica: dollar
71	JOD	Jordan: dinar
72	KWD	Kuwait: dinar
73	MOP	Macao SAR: pataca
74	NIO	Nicaragua: cordoba oro
75	PAB	Panama: balboa
76	PEN	Peru: nuevo sol
77	PKR	Pakistan: rupee
78	PYG	Paraguay: guarani
79	SAR	Saudi Arabia: riyal

ID	String	Meaning
80	SIT	Slovenia: tolar
81	SKK	Slovakia: koruna
82	SVC	El Salvador: colon
83	TRY	Turkey: new lira
84	TTD	Trinidad and Tobago: dollar
85	UYU	Uruguay: peso
86	VEB	Venezuela: bolivar
87	VND	Viet Nam: dong
88	BRL	Brazil: real
89	PLN	Poland: zloty
90	RUB	Russia: rouble
91	YUM	Serbia and Montenegro: dinar
92	BYB	Belarus: ruble
93	UAH	Ukraine: hryvnia
94	AFA	Afghanistan: afghani
95	ALL	Albania: lek
96	DZD	Algeria: dinar
97	ADP	Andorra: peseta
98	AOA	Angola: kwanza
99	XCD	East Caribbean Dollar
100	AMD	Armenia: dram
101	AWG	Aruba: guilder
102	AZM	Azerbaijan: manat
103	BHD	Bahrain: dinar
104	BDT	Bangladesh: taka
105	BBD	Barbados: dollar
106	BYR	Belarus: ruble
107	BZD	Belize: dollar
108	XOF	CFA Franc BCEAO
109	BTN	Bhutan: ngultrum
110	BAM	Bosnia and Herzegovina: convertible marks
111	BWP	Botswana: pula
112	BND	Brunei: dollar
113	BGL	Bulgaria: lev (historic)

ID	String	Meaning
114	BGN	Bulgaria: lev
115	BIF	Burundi: franc
116	KHR	Cambodia: riel
117	XAF	CFA Franc BEAC
118	CVE	Cape Verde: escudo
119	KYD	Cayman Islands: dollar
120	KMF	Comoros: franc
121	CDF	Congo (DRC): franc
122	HRK	Croatia: kuna
123	CUP	Cuba: peso
124	CYP	Cyprus: pound
125	DJF	Djibouti: franc
126	ERN	East Timor: escudo
127	TPE	Eritrea: nakfa
128	EEK	Estonia: kroon
129	ETB	Ethiopia: birr
130	FKP	Falkland Islands (Islas Malvinas): pound
131	FJD	Fiji Islands: dollar
132	XPF	CFP Franc
133	GMD	The Gambia: dalasi
134	GEL	Georgia: lari
135	GHC	Ghana: cedi
136	GIP	Gibraltar: pound
137	GTQ	Guatemala: quetzal
138	GNF	Guinea: franc
139	GWP	Guinea-Bissau: peso
140	GYD	Guyana: dollar
141	HTG	Haiti: gourde
142	ISK	Iceland: krona
143	IRR	Iran: rial
144	IQD	Iraq: dinar
145	KZT	Kazakhstan: tenge
146	KES	Kenya: shilling
147	KPW	North Korean: won

ID	String	Meaning
148	KGS	Kyrgyzstan: som
149	LAK	Laos: kip
150	LVL	Latvia: lats
151	LBP	Lebanon: pound
152	LSL	Lesotho: loti
153	LRD	Liberia: dollar
154	LYD	Libya: dinar
155	LTL	Lithuania: litus
156	MKD	Former Yugoslav Republic of Macedonia: denar
157	MGF	Madagascar: franc
158	MWK	Malawi: kwacha
159	MVR	Maldives: rufiyaa
160	MTL	Malta: lira
161	MRO	Mauritania: ouguiya
162	MUR	Mauritius: rupee
163	MDL	Moldova: leu
164	MNT	Mongolia: tugrik
165	MAD	Morocco: dirham
166	MZM	Mozambique: metical
167	MMK	Myanmar: kyat
168	NAD	Namibia: dollar
169	NPR	Nepal: rupee
170	ANG	Netherlands Antilles: guilder
171	NGN	Nigeria: naira
172	OMR	Oman: rial
173	PGK	Papua New Guinea: kina
174	QAR	Qatar: rial
175	RWF	Rwanda: franc
176	SHP	Saint Helena: pound
177	WST	Samoa: tala
178	STD	Sao Tome and Principe: dobra
179	SCR	Seychelles: rupee
180	SLL	Sierra Leone: leone
181	SBD	Solomon Islands: dollar

ID	String	Meaning
182	SOS	Somalia: shilling
183	LKR	Sri Lanka: rupee
184	SDD	Sudan: dinar
185	SRG	Suriname: guilder
186	SZL	Swaziland: lilangeni
187	SYP	Syria: pound
188	TJR	Tajikistan: ruble
189	TJS	Tajikistan: somoni
190	TZS	Tanzania: shilling
191	TOP	Tonga: pa'anga
192	TND	Tunisia: dinar
193	TMM	Turkmenistan: manat
194	UGX	Uganda: shilling
195	AED	United Arab Emirates: dirham
196	UZS	Uzbekistan: sum
197	VUV	Vanuatu: vatu
198	YER	Yemen: rial
199	ZMK	Zambia: kwacha
200	ZWD	Zimbabwe: dollar
201	VEF	Venezuela: bolivar fuerte
202	MGA	Madagascar: ariary
203	RSD	Serbia: dinar
204	CSD	Serbia: dinar (Former Serbia and Montenegro)

2.5.9.3 vDataType

The **vDataType** custom structure is an unsigned integer that specifies a data type.

It MUST be a value from the following table.

Value	Meaning
0	Specifies that the data is formatted as a string.
1	Specifies that the data is formatted as a string.
2	Specifies that the data is formatted as a number.
3	Specifies that the data is formatted as a Boolean.

Value	Meaning
4	Specifies that the data is formatted as a string.
5	Specifies that the data is formatted as a date and time value.
6	Specifies that the data is formatted as a duration value.
7	Specifies that the data is formatted as a currency value.

2.5.9.4 vFieldPicture

The **vFieldPicture** custom structure is an unsigned integer that specifies a predefined format picture string.

It MUST be a value from the following table.

Value	Format string
0	0.####
1	0.#### u
2	0
3	0 u
4	0.0
5	0.0 u
6	0.00
7	0.00 u
8	0.000
9	0.000 u
10	<,FEET/INCH>0.000 u
11	<,rad>0.#### u
12	<,deg>0.# u
13	<,FEET/INCH># #/# u
14	<,FEET/INCH># #/## u
15	0 #/#
16	0 #/# u
17	0 #/##
18	0 #/## u
20	ddddd
21	dddddd

Value	Format string
22	M/d/y
23	MM/dd/yy
24	MMM d, yyyy
25	MMMM d, yyyy
26	d/M/YY
27	dd/MM/yy
28	d MMM, yyyy
29	d MMMM, yyyy
30	T
31	h:mm
32	hh:mm
33	H:mm
34	HH:mm
35	h:mm tt
36	Hh:mm tt
37	@
38	@-
39	@+
40 ~ 81	M/d/yyyy
200	M/d/yyyy
201	dddd, MMMM dd, yyyy
202	MMMM d, yyyy
203	M/d/yy
204	yyyy-MM-dd
205	d-MMM-yy
206	M.d.yyyy
207	MMM. d, yy
208	D MMMM yyyy
209	MMMM yy
210	MMM-yy
211	M/d/yyyy h:mm am/pm
212	M/d/yyyy h:mm:ss am/pm

Value	Format string
213	H:mm am/pm
214	H:m:ss am/pm
215	HH:mm
216	HH:mm:ss
217	M/d/yyyy
218	M/d/yyyy
219	M/d/yyyy

2.5.9.5 vFont

The **vFont** custom structure is a string that specifies the name of a system font. If the value is "0" or "", the structure does not specify a font.

2.5.9.6 vFormatString

The **vFormatString** custom structure is a string that specifies the formatting information to determine how a value is displayed. If **vFormatString** is an empty string, the result string is an empty string.

A **vFormatString** is composed of casting notation, field pictures, and format strings.

Casting notation: The casting notation specifies the unit of the value to be formatted and the display unit of the value. If present it MUST appear at the beginning of the **vFormatString**. If the casting notation is present, the notation is stripped from the format string before the formatting is performed. It MUST satisfy the following ABNF.

ABNF:

```
CastingNotation = "<" [ SourceUnit ] "," [ DisplayUnit ] ">"
SourceUnit = string-value
DisplayUnit = string-value
```

The first **string-value** specifies a unit of the value to be formatted. It MUST be a vUnitString or a value equal to "$".

The second **string-value** specifies a unit displayed in the resulting string. It MUST be a vUnitString or a value equal to "$".

The value to be formatted is converted as follows. If the value to be formatted is a vScalar and **SourceUnit** is a vUnitString, the value is multiplied by a factor that converts the unit of **SourceUnit** into the custom internal unit type associated with the type of the unit of **SourceUnit**. If the value to be formatted is a vScalar and **SourceUnit** is equal to "$", the value is converted to PtgCy. If **DisplayUnit** is a vUnitString, the resulting value is multiplied by a factor that converts the custom internal unit type associated with the type of the unit of **DisplayUnit** into the unit of **DisplayUnit.** If **DisplayUnit** is "$", the resulting value is converted to PtgCy.

Field picture: A field picture is a predefined index that specifies format strings. If the format string contains a field picture, it is replaced by the corresponding format string as specified in vFieldPicture before the formatting is performed. It MUST satisfy the following ABNF.

ABNF:

```
Field-picture = "{<" unsigned-int-value ">}" / "esc("unsigned-int-value ")"
```

The first **unsigned-int-value** specifies a field picture index. It MUST be a vFieldPicture. If it is not found in the table of vFieldPicture, the default value is following. If the value that is formatted is a PtgCy or vNumAny (except PtgDate, PtgEDay, PtgEHour, PtgEMin, PtgESec, or PtgEWeek), the default value is zero. If the value that is formatted is a PtgDate, PtgEDay, PtgEHour, PtgEMin, PtgESec, or PtgEWeek, the default value is 200. Otherwise, the formatting returns an empty string.

The second **unsigned-int-value** specifies a field picture index. It MUST be a vFieldPicture. If it is greater than 18, the default value is zero.

String format strings: If the format string contains the character "@", the value string is formatted into the result string as specified by the following table. The result strings produced by these format strings are influenced by the language specified for the formatting. The default value for the language is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

Format Character	Name	Description
\	Escape character	The next character is inserted into the result string and is not interpreted as a format character. For example, to display the backslash character, "\\" is used.
"text" 'text'	Literal string	The text enclosed in quotation marks is inserted into the result string and the characters are not interpreted as format characters.
@	Text placeholder	The value string is inserted into the result string.
@+	Uppercase text placeholder	The value string is converted to uppercase and inserted into the result string.
@-	Lowercase text placeholder	The value string is converted to lowercase and inserted into the result string. For example, FORMAT("Hello", "@ @+ @-") displays "Hello HELLO hello".

Date/Time format strings: If the format string does not contain the characters "@", "#", or "0", and does contain the characters "h", "H", "m", "M", "s", "S", "d", "D", "y", "Y", "t", "T", "g", "G", "n", "N", "e", "E", "w", "W", "c", or "C", the value is formatted as specified by the following table. The result strings produced by these format strings are influenced by the language and calendar that are specified for the formatting. The default value for the language is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] section 11, from the Core XML part of a web drawing.

In the following table, the format character values listed on multiple lines in the same table row are equivalent.

Format Character	Name	Description
d	Day placeholder	The day of the month as a number is inserted into the result string.
dd	Day placeholder	The day of the month as a two-digit number, with a leading zero if necessary, is inserted into the result string.
ddd w	Short day of week placeholder	The day as an abbreviation is inserted into the result string.
dddd ww	Long day of week placeholder	The day as a full name is inserted into the result string.
M	Month placeholder	The month as a number is inserted into the result string.
MM	Month placeholder	The month as a two-digit number, with a leading zero if necessary, is inserted into the result string.
MMM	Month placeholder	The name of the month in abbreviated form is inserted into the result string.
MMMM	Month placeholder	The full name of the month is inserted into the result string.
yy	Year placeholder	The year as a two-digit number is inserted into the result string.
yyyy	Year placeholder	The year as a four-digit number is inserted into the result string.
h	Hour placeholder	The hour in 12-hour form is inserted into the result string.
hh	Hour placeholder	The hour as a two-digit number, with a leading zero if necessary, in 12-hour form is inserted into the result string.
H	Hour placeholder	The hour in 24-hour form is inserted into the result string.
HH	Hour placeholder	The hour as a two-digit number, with a leading zero if necessary, in 24-hour form is inserted into the result string.
m	Minute placeholder	The minute is inserted into the result string.
mm	Minute placeholder	The minute as a two-digit number, with a leading zero if necessary, is inserted into the result string.
s	Second placeholder	The second is inserted into the result string.
ss	Second placeholder	The second as a two-digit number, with a leading zero if necessary, is inserted into the result string.
t	AM/PM abbreviation	The first character of the AM/PM designator for the value is inserted into the result string.
tt am/pm	AM/PM designator	The AM/PM designator for the value is inserted into the result string.
ddddd	Short date placeholder	The value is formatted using a format string "M/d/yyyy" and inserted to the result string. For example, FORMAT(DATETIME("6/25/07 12:05"), "ddddd") displays 6/25/2007.

Format Character	Name	Description
dddddd	Long date placeholder	The value is formatted using a format string "dddd, MMMM dd, yyyy" and inserted to the result string. For example, FORMAT(DATETIME("6/25/07 12:05"), "dddddd") displays Monday, June 25, 2007.
c	Short date/time placeholder	The value is formatted using a format string "M/d/yyyy h:mm:ss tt" and inserted to the result string. If the time value is zero, the value is formatted using a format string "M/d/yyyy" and inserted to the result string. For example, FORMAT(DATETIME("6/25/07 12:05"), "c") displays 6/25/2007 12:05:00 PM.
C	Long date/time placeholder	The value is formatted using a format string "dddd, MMMM dd, yyyy h:mm:ss tt" and inserted to the result string. For example, FORMAT(DATETIME("6/25/07 12:05"), "C") displays Monday, June 25, 2007 12:05:00 PM.
T	Long time placeholder	The value is formatted using a format string "h:mm:ss tt" and inserted to the result string. For example, FORMAT(DATETIME("6/25/07 12:05"), "T") displays 12:05:00 PM.
/	Date separator	The date separator is inserted into the result string.
:	Time separator	The time separator is inserted into the result string.
[y] [Y]	Elapsed years placeholder	The elapsed year value is inserted into the result string.
[yy] [YY]	Elapsed years placeholder	The elapsed year value, as a two-digit number with a leading zero if necessary, is inserted into the result string.
[w] [W]	Elapsed weeks placeholder	The elapsed week value is inserted into the result string.
[ww] [WW]	Elapsed weeks placeholder	The elapsed week value, as a two-digit number with a leading zero if necessary, with a leading zero is inserted into the result string.
[d] [D]	Elapsed days placeholder	The elapsed day value is inserted into the result string.
[dd] [DD]	Elapsed days placeholder	The elapsed day value, as a two-digit number with a leading zero if necessary, is inserted into the result string.
[h] [H]	Elapsed hours placeholder	The elapsed hour value is inserted into the result string.
[hh] [HH]	Elapsed hours placeholder	The elapsed hour value, as a two-digit number with a leading zero if necessary, is inserted into the result string.
[m]	Elapsed minutes placeholder	The elapsed minute value is inserted into the result string.

Format Character	Name	Description
[mm]	Elapsed minutes placeholder	The elapsed minute value, as a two-digit number with a leading zero if necessary, is inserted into the result string.
[s]	Elapsed seconds placeholder	The elapsed second value is inserted into the result string.
[ss]	Elapsed seconds placeholder	The elapsed second value, as a two-digit number with a leading zero if necessary, is inserted into the result string.

Numeric format strings: If the format string does not contain the characters "@", and does contain the characters "#" or "0", the value is formatted as specified by the following table. The result strings produced by some of these format strings are influenced by the language that is specified for the formatting. The default value for the language is specified by the **Language** property, defined in [ISO/IEC29500-2:2011] Section 11, from the Core XML part of a web drawing.

Format Character	Name	Description
#	Digit placeholder	If the value being formatted has a digit in the position where the '#' appears in the format string, that digit is inserted into the result string. If the value has more digits than there are placeholders to the left of the decimal, all digits are inserted into the result string. If the value has more digits than there are placeholders to the right of the decimal, the fraction is rounded to the number of placeholders. For example, FORMAT(0.0239, "#.###") displays as .024.
0	Zero placeholder	The digit in the position where the '0' appears in the format string is inserted into the result string. If the value has more digits than there are placeholders to the left of the decimal, all digits are inserted into the result string. If the value has more digits than there are placeholders to the right of the decimal, the fraction is rounded to the number of placeholders. For example, FORMAT(0.0239, "0.00000") displays as 0.02390.
.	Decimal separator	The first '.' character in the format string determines the location of the decimal separator in the formatted value. Any additional '.' characters are ignored.
,	Thousands separator	If the format string contains a ',' character between two digit placeholders (0 or #) and if one is present to the left of the decimal point, the result string has a thousands separator inserted between each group of three digits to the left of the decimal separator.
E+	Scientific notation	If the format string contains at least one digit placeholder to the right of this character, the number is formatted using scientific notation with an 'E' inserted between the mantissa and the exponent. The number of digit placeholders following the scientific notation indicator determines the minimum number of digits to output for the exponent. A sign character '+' always precedes the exponent. For example, FORMAT(12345.67,"###.#E+0") displays as 123.5E+2.
e+	Scientific notation	If the format string contains at least one digit placeholder to the right of this character, the number is formatted using scientific notation with an 'e' inserted between the mantissa and the exponent. The number of digit

Format Character	Name	Description
		placeholders following the scientific notation indicator determines the minimum number of digits to output for the exponent. A sign character '+' always precedes the exponent.
E-	Scientific notation	If the format string contains at least one digit placeholder to the right of this character, the number is formatted using scientific notation with an 'E' inserted between the mantissa and the exponent. The number of digit placeholders following the scientific notation indicator determines the minimum number of digits to output for the exponent. A sign character '-' only precedes negative exponents.
e-	Scientific notation	If the format string contains at least one digit placeholder to the right of this character, the number is formatted using scientific notation with an 'e' inserted between the mantissa and the exponent. The number of digit placeholders following the scientific notation indicator determines the minimum number of digits to output for the exponent. A sign character '-' only precedes negative exponents.
/	Fraction placeholder	The value is formatted as a whole number and fraction, and inserted into the result string. The number of digits of the denominator is determined by the number of digit placeholders following the fraction placeholder. The maximum number of digits of a denominator is 5. The value is rounded to the nearest fraction and the simplest form of the fraction is inserted into the result string. For example, FORMAT(12.43, "# #/#") displays as 12 3/7.
{space}	Space placeholder	A space character is inserted into the result string.
u	Lower case, Short label placeholder	A unit measurement of the numeric value in lower case, and an abbreviated form is inserted into the result string. For example, FORMAT(12.43in, "#.## u") displays as 12.43 in.
U	Upper case, Short label placeholder	A unit measurement of the numeric value in upper case, and an abbreviated form is inserted into the result string. For example, FORMAT(12.43in, "#.## U") displays as 12.43 IN.
uu	Lower case, Long label placeholder	A unit measurement of the numeric value in lower case is inserted into the result string. For example, FORMAT(12.43in, "#.## uu") displays as 12.43 inches.
UU	Upper case, Long label placeholder	A unit measurement of the numeric value in upper case is inserted into the result string. For example, FORMAT(12.43in, "#.## UU") displays as 12.43 INCHES.
uuu	Lower case, Universal label placeholder	A unit measurement of the numeric value in lower case, and a universal form is inserted into the result string. For example, FORMAT(12.43in, "#.## uuu") displays as 12.43 in.
UUU	Upper case, Universal label	A unit measurement of the numeric value in upper case, and a universal form is inserted into the result string.

Format Character	Name	Description
	placeholder	For example, FORMAT(12.43in, "#.## UUU") displays as 12.43 IN.

2.5.9.7 vLanguage

The **vLanguage** custom structure is either a vLanguageID or vLanguageString.

2.5.9.8 vLanguageID

The **vLanguageID** custom structure is an unsigned integer that specifies an LCID that MUST be from the vLanguageString table.

2.5.9.9 vLanguageString

The **vLanguageString** custom structure is a string that specifies a culture name that MUST be from the following table.

LCID	Culture Name	Language
0x0036	af	Afrikaans
0x0436	af-ZA	Afrikaans (South Africa)
0x001C	sq	Albanian
0x041C	sq-AL	Albanian (Albania)
0x0084	gsw	Alsatian
0x0484	gsw-FR	Alsatian (France)
0x005E	am	Amharic
0x045E	am-ET	Amharic (Ethiopia)
0x0001	ar	Arabic
0x1401	ar-DZ	Arabic (Algeria)
0x3C01	ar-BH	Arabic (Bahrain)
0x0C01	ar-EG	Arabic (Egypt)
0x0801	ar-IQ	Arabic (Iraq)
0x2C01	ar-JO	Arabic (Jordan)
0x3401	ar-KW	Arabic (Kuwait)
0x3001	ar-LB	Arabic (Lebanon)
0x1001	ar-LY	Arabic (Libya)
0x1801	ar-MA	Arabic (Morocco)
0x2001	ar-OM	Arabic (Oman)

LCID	Culture Name	Language
0x4001	ar-QA	Arabic (Qatar)
0x0401	ar-SA	Arabic (Saudi Arabia)
0x2801	ar-SY	Arabic (Syria)
0x1C01	ar-TN	Arabic (Tunisia)
0x3801	ar-AE	Arabic (U.A.E.)
0x2401	ar-YE	Arabic (Yemen)
0x002B	hy	Armenian
0x042B	hy-AM	Armenian (Armenia)
0x004D	as	Assamese
0x044D	as-IN	Assamese (India)
0x002C	az	Azeri
0x742C	az-Cyrl	Azeri (Cyrillic)
0x082C	az-Cyrl-AZ	Azeri (Cyrillic, Azerbaijan)
0x782C	az-Latn	Azeri (Latin)
0x042C	az-Latn-AZ	Azeri (Latin, Azerbaijan)
0x006D	ba	Bashkir
0x046D	ba-RU	Bashkir (Russia)
0x002D	eu	Basque
0x042D	eu-ES	Basque (Basque)
0x0023	be	Belarusian
0x0423	be-BY	Belarusian (Belarus)
0x0045	bn	Bengali
0x0845	bn-BD	Bengali (Bangladesh)
0x0445	bn-IN	Bengali (India)
0x781A	bs	Bosnian
0x641A	bs-Cyrl	Bosnian (Cyrillic)
0x201A	bs-Cyrl-BA	Bosnian (Cyrillic, Bosnia and Herzegovina)
0x681A	bs-Latn	Bosnian (Latin)
0x141A	bs-Latn-BA	Bosnian (Latin, Bosnia and Herzegovina)
0x007E	br	Breton
0x047E	br-FR	Breton (France)
0x0002	bg	Bulgarian

LCID	Culture Name	Language
0x0402	bg-BG	Bulgarian (Bulgaria)
0x0003	ca	Catalan
0x0403	ca-ES	Catalan (Catalan)
0x7804	zh	Chinese
0x0004	zh-Hans	Chinese (Simplified)
0x0804	zh-CN	Chinese (Simplified, PRC)
0x1004	zh-SG	Chinese (Simplified, Singapore)
0x7C04	zh-Hant	Chinese (Traditional)
0x0C04	zh-HK	Chinese (Traditional, Hong Kong S.A.R.)
0x1404	zh-MO	Chinese (Traditional, Macao S.A.R.)
0x0404	zh-TW	Chinese (Traditional, Taiwan)
0x0083	co	Corsican
0x0483	co-FR	Corsican (France)
0x001A	hr	Croatian
0x041A	hr-HR	Croatian (Croatia)
0x101A	hr-BA	Croatian (Latin, Bosnia and Herzegovina)
0x0005	cs	Czech
0x0405	cs-CZ	Czech (Czech Republic)
0x0006	da	Danish
0x0406	da-DK	Danish (Denmark)
0x008C	prs	Dari
0x048C	prs-AF	Dari (Afghanistan)
0x0065	dv	Divehi
0x0465	dv-MV	Divehi (Maldives)
0x0013	nl	Dutch
0x0813	nl-BE	Dutch (Belgium)
0x0413	nl-NL	Dutch (Netherlands)
0x0009	en	English
0x0C09	en-AU	English (Australia)
0x2809	en-BZ	English (Belize)
0x1009	en-CA	English (Canada)
0x2409	en-029	English (Caribbean)

LCID	Culture Name	Language
0x4009	en-IN	English (India)
0x1809	en-IE	English (Ireland)
0x2009	en-JM	English (Jamaica)
0x4409	en-MY	English (Malaysia)
0x1409	en-NZ	English (New Zealand)
0x3409	en-PH	English (Republic of the Philippines)
0x4809	en-SG	English (Singapore)
0x1C09	en-ZA	English (South Africa)
0x2C09	en-TT	English (Trinidad and Tobago)
0x0809	en-GB	English (United Kingdom)
0x0409	en-US	English (United States)
0x3009	en-ZW	English (Zimbabwe)
0x0025	et	Estonian
0x0425	et-EE	Estonian (Estonia)
0x0038	fo	Faroese
0x0438	fo-FO	Faroese (Faroe Islands)
0x0064	fil	Filipino
0x0464	fil-PH	Filipino (Philippines)
0x000B	fi	Finnish
0x040B	fi-FI	Finnish (Finland)
0x000C	fr	French
0x080C	fr-BE	French (Belgium)
0x0C0C	fr-CA	French (Canada)
0x040C	fr-FR	French (France)
0x140C	fr-LU	French (Luxembourg)
0x180C	fr-MC	French (Monaco)
0x100C	fr-CH	French (Switzerland)
0x0062	fy	Frisian
0x0462	fy-NL	Frisian (Netherlands)
0x0056	gl	Galician
0x0456	gl-ES	Galician (Galician)
0x0037	ka	Georgian

LCID	Culture Name	Language
0x0437	ka-GE	Georgian (Georgia)
0x0007	de	German
0x0C07	de-AT	German (Austria)
0x0407	de-DE	German (Germany)
0x1407	de-LI	German (Liechtenstein)
0x1007	de-LU	German (Luxembourg)
0x0807	de-CH	German (Switzerland)
0x0008	el	Greek
0x0408	el-GR	Greek (Greece)
0x006F	kl	Greenlandic
0x046F	kl-GL	Greenlandic (Greenland)
0x0047	gu	Gujarati
0x0447	gu-IN	Gujarati (India)
0x0068	ha	Hausa
0x7C68	ha-Latn	Hausa (Latin)
0x0468	ha-Latn-NG	Hausa (Latin, Nigeria)
0x000D	he	Hebrew
0x040D	he-IL	Hebrew (Israel)
0x0039	hi	Hindi
0x0439	hi-IN	Hindi (India)
0x000E	hu	Hungarian
0x040E	hu-HU	Hungarian (Hungary)
0x000F	is	Icelandic
0x040F	is-IS	Icelandic (Iceland)
0x0070	ig	Igbo
0x0470	ig-NG	Igbo (Nigeria)
0x0021	id	Indonesian
0x0421	id-ID	Indonesian (Indonesia)
0x005D	iu	Inuktitut
0x7C5D	iu-Latn	Inuktitut (Latin)
0x085D	iu-Latn-CA	Inuktitut (Latin, Canada)
0x785D	iu-Cans	Inuktitut (Syllabics)

LCID	Culture Name	Language
0x045D	iu-Cans-CA	Inuktitut (Syllabics, Canada)
0x003C	ga	Irish
0x083C	ga-IE	Irish (Ireland)
0x0034	xh	isiXhosa
0x0434	xh-ZA	isiXhosa (South Africa)
0x0035	zu	isiZulu
0x0435	zu-ZA	isiZulu (South Africa)
0x0010	it	Italian
0x0410	it-IT	Italian (Italy)
0x0810	it-CH	Italian (Switzerland)
0x0011	ja	Japanese
0x0411	ja-JP	Japanese (Japan)
0x004B	kn	Kannada
0x044B	kn-IN	Kannada (India)
0x003F	kk	Kazakh
0x043F	kk-KZ	Kazakh (Kazakhstan)
0x0053	km	Khmer
0x0453	km-KH	Khmer (Cambodia)
0x0086	qut	K'iche
0x0486	qut-GT	K'iche (Guatemala)
0x0087	rw	Kinyarwanda
0x0487	rw-RW	Kinyarwanda (Rwanda)
0x0041	sw	Kiswahili
0x0441	sw-KE	Kiswahili (Kenya)
0x0057	kok	Konkani
0x0457	kok-IN	Konkani (India)
0x0012	ko	Korean
0x0412	ko-KR	Korean (Korea)
0x0040	ky	Kyrgyz
0x0440	ky-KG	Kyrgyz (Kyrgyzstan)
0x0054	lo	Lao
0x0454	lo-LA	Lao (Lao P.D.R.)

LCID	Culture Name	Language
0x0026	lv	Latvian
0x0426	lv-LV	Latvian (Latvia)
0x0027	lt	Lithuanian
0x0427	lt-LT	Lithuanian (Lithuania)
0x7C2E	dsb	Lower Sorbian
0x082E	dsb-DE	Lower Sorbian (Germany)
0x006E	lb	Luxembourgish
0x046E	lb-LU	Luxembourgish (Luxembourg)
0x042F	mk-MK	Macedonian (Former Yugoslav Republic of Macedonia)
0x002F	mk	Macedonian (FYROM)
0x003E	ms	Malay
0x083E	ms-BN	Malay (Brunei Darussalam)
0x043E	ms-MY	Malay (Malaysia)
0x004C	ml	Malayalam
0x044C	ml-IN	Malayalam (India)
0x003A	mt	Maltese
0x043A	mt-MT	Maltese (Malta)
0x0081	mi	Maori
0x0481	mi-NZ	Maori (New Zealand)
0x007A	arn	Mapudungun
0x047A	arn-CL	Mapudungun (Chile)
0x004E	mr	Marathi
0x044E	mr-IN	Marathi (India)
0x007C	moh	Mohawk
0x047C	moh-CA	Mohawk (Mohawk)
0x0050	mn	Mongolian (Cyrillic)
0x7850	mn-Cyrl	Mongolian (Cyrillic)
0x0450	mn-MN	Mongolian (Cyrillic, Mongolia)
0x7C50	mn-Mong	Mongolian (Traditional Mongolian)
0x0850	mn-Mong-CN	Mongolian (Traditional Mongolian, PRC)
0x0061	ne	Nepali
0x0461	ne-NP	Nepali (Nepal)

LCID	Culture Name	Language
0x0014	no	Norwegian
0x7C14	nb	Norwegian (Bokmål)
0x7814	nn	Norwegian (Nynorsk)
0x0414	nb-NO	Norwegian, Bokmål (Norway)
0x0814	nn-NO	Norwegian, Nynorsk (Norway)
0x0082	oc	Occitan
0x0482	oc-FR	Occitan (France)
0x0048	or	Oriya
0x0448	or-IN	Oriya (India)
0x0063	ps	Pashto
0x0463	ps-AF	Pashto (Afghanistan)
0x0029	fa	Persian
0x0429	fa-IR	Persian
0x0015	pl	Polish
0x0415	pl-PL	Polish (Poland)
0x0016	pt	Portuguese
0x0416	pt-BR	Portuguese (Brazil)
0x0816	pt-PT	Portuguese (Portugal)
0x0046	pa	Punjabi
0x0446	pa-IN	Punjabi (India)
0x006B	quz	Quechua
0x046B	quz-BO	Quechua (Bolivia)
0x086B	quz-EC	Quechua (Ecuador)
0x0C6B	quz-PE	Quechua (Peru)
0x0018	ro	Romanian
0x0418	ro-RO	Romanian (Romania)
0x0017	rm	Romansh
0x0417	rm-CH	Romansh (Switzerland)
0x0019	ru	Russian
0x0419	ru-RU	Russian (Russia)
0x703B	smn	Sami (Inari)
0x7C3B	smj	Sami (Lule)

LCID	Culture Name	Language
0x003B	se	Sami (Northern)
0x743B	sms	Sami (Skolt)
0x783B	sma	Sami (Southern)
0x243B	smn-FI	Sami, Inari (Finland)
0x103B	smj-NO	Sami, Lule (Norway)
0x143B	smj-SE	Sami, Lule (Sweden)
0x0C3B	se-FI	Sami, Northern (Finland)
0x043B	se-NO	Sami, Northern (Norway)
0x083B	se-SE	Sami, Northern (Sweden)
0x203B	sms-FI	Sami, Skolt (Finland)
0x183B	sma-NO	Sami, Southern (Norway)
0x1C3B	sma-SE	Sami, Southern (Sweden)
0x004F	sa	Sanskrit
0x044F	sa-IN	Sanskrit (India)
0x0091	gd	Scottish Gaelic
0x0491	gd-GB	Scottish Gaelic (United Kingdom)
0x7C1A	sr	Serbian
0x6C1A	sr-Cyrl	Serbian (Cyrillic)
0x1C1A	sr-Cyrl-BA	Serbian (Cyrillic, Bosnia and Herzegovina)
0x301A	sr-Cyrl-ME	Serbian (Cyrillic, Montenegro)
0x0C1A	sr-Cyrl-CS	Serbian (Cyrillic, Serbia and Montenegro (Former))
0x281A	sr-Cyrl-RS	Serbian (Cyrillic, Serbia)
0x701A	sr-Latn	Serbian (Latin)
0x181A	sr-Latn-BA	Serbian (Latin, Bosnia and Herzegovina)
0x2C1A	sr-Latn-ME	Serbian (Latin, Montenegro)
0x081A	sr-Latn-CS	Serbian (Latin, Serbia and Montenegro (Former))
0x241A	sr-Latn-RS	Serbian (Latin, Serbia)
0x006C	nso	Sesotho sa Leboa
0x046C	nso-ZA	Sesotho sa Leboa (South Africa)
0x0032	tn	Setswana
0x0432	tn-ZA	Setswana (South Africa)
0x005B	si	Sinhala

LCID	Culture Name	Language
0x045B	si-LK	Sinhala (Sri Lanka)
0x001B	sk	Slovak
0x041B	sk-SK	Slovak (Slovakia)
0x0024	sl	Slovenian
0x0424	sl-SI	Slovenian (Slovenia)
0x000A	es	Spanish
0x2C0A	es-AR	Spanish (Argentina)
0x400A	es-BO	Spanish (Bolivia)
0x340A	es-CL	Spanish (Chile)
0x240A	es-CO	Spanish (Colombia)
0x140A	es-CR	Spanish (Costa Rica)
0x1C0A	es-DO	Spanish (Dominican Republic)
0x300A	es-EC	Spanish (Ecuador)
0x440A	es-SV	Spanish (El Salvador)
0x100A	es-GT	Spanish (Guatemala)
0x480A	es-HN	Spanish (Honduras)
0x080A	es-MX	Spanish (Mexico)
0x4C0A	es-NI	Spanish (Nicaragua)
0x180A	es-PA	Spanish (Panama)
0x3C0A	es-PY	Spanish (Paraguay)
0x280A	es-PE	Spanish (Peru)
0x500A	es-PR	Spanish (Puerto Rico)
0x0C0A	es-ES	Spanish (Spain, International Sort)
0x540A	es-US	Spanish (United States)
0x380A	es-UY	Spanish (Uruguay)
0x200A	es-VE	Spanish (Venezuela)
0x001D	sv	Swedish
0x081D	sv-FI	Swedish (Finland)
0x041D	sv-SE	Swedish (Sweden)
0x005A	syr	Syriac
0x045A	syr-SY	Syriac (Syria)
0x0028	tg	Tajik (Cyrillic)

LCID	Culture Name	Language
0x7C28	tg-Cyrl	Tajik (Cyrillic)
0x0428	tg-Cyrl-TJ	Tajik (Cyrillic, Tajikistan)
0x005F	tzm	Tamazight
0x7C5F	tzm-Latn	Tamazight (Latin)
0x085F	tzm-Latn-DZ	Tamazight (Latin, Algeria)
0x0049	ta	Tamil
0x0449	ta-IN	Tamil (India)
0x0044	tt	Tatar
0x0444	tt-RU	Tatar (Russia)
0x004A	te	Telugu
0x044A	te-IN	Telugu (India)
0x001E	th	Thai
0x041E	th-TH	Thai (Thailand)
0x0051	bo	Tibetan
0x0451	bo-CN	Tibetan (PRC)
0x001F	tr	Turkish
0x041F	tr-TR	Turkish (Turkey)
0x0042	tk	Turkmen
0x0442	tk-TM	Turkmen (Turkmenistan)
0x0022	uk	Ukrainian
0x0422	uk-UA	Ukrainian (Ukraine)
0x002E	hsb	Upper Sorbian
0x042E	hsb-DE	Upper Sorbian (Germany)
0x0020	ur	Urdu
0x0420	ur-PK	Urdu (Islamic Republic of Pakistan)
0x0080	ug	Uyghur
0x0480	ug-CN	Uyghur (PRC)
0x7843	uz-Cyrl	Uzbek (Cyrillic)
0x0843	uz-Cyrl-UZ	Uzbek (Cyrillic, Uzbekistan)
0x0043	uz	Uzbek (Latin)
0x7C43	uz-Latn	Uzbek (Latin)
0x0443	uz-Latn-UZ	Uzbek (Latin, Uzbekistan)

LCID	Culture Name	Language
0x002A	vi	Vietnamese
0x042A	vi-VN	Vietnamese (Vietnam)
0x0052	cy	Welsh
0x0452	cy-GB	Welsh (United Kingdom)
0x0088	wo	Wolof
0x0488	wo-SN	Wolof (Senegal)
0x0085	sah	Yakut
0x0485	sah-RU	Yakut (Russia)
0x0078	ii	Yi
0x0478	ii-CN	Yi (PRC)
0x006A	yo	Yoruba
0x046A	yo-NG	Yoruba (Nigeria)

2.5.9.10 vPanose

The **vPanose** custom structure specifies a string that contains ten integers separated by space. Those integers are used to specify the typefaces of a font.

The meaning of each integer is specified in the following table.

Index	Meaning
0	Family Type
1	Serif Style
2	Weight
3	Proportion
4	Contrast
5	Stroke Variation
6	Arm Style
7	Letterform
8	Midline
9	X-height

2.5.9.11 vThemeString

The **vThemeString** custom structure is a string that specifies a fixed theme property.

It MUST be a value from the following table.

Fixed Theme Property	User Row_Type	Scheme	Value Type
TextColor	msvThemeTextColor	Color	PtgColorRGB
FillColor	msvThemeFillColor	Color	PtgColorRGB
FillColor2	msvThemeFillColor2 (section 2.4.3.26)	Color	PtgColorRGB
LineColor	msvThemeLineColor	Color	PtgColorRGB
ConnectorColor	msvThemeConnectorColor	Color	PtgColorRGB
ShadowColor	msvThemeShadowColor	Color	PtgColorRGB
AccentColor	msvThemeAccentColor	Color	PtgColorRGB
AccentColor2	msvThemeAccentColor2.4.3.6	Color	PtgColorRGB
AccentColor3	msvThemeAccentColor2.4.3.7	Color	PtgColorRGB
AccentColor4	msvThemeAccentColor2.4.3.8	Color	PtgColorRGB
AccentColor5	msvThemeAccentColor2.4.3.9	Color	PtgColorRGB
BackgroundColor	msvThemeBackgroundColor	Color	PtgColorRGB
LatinFont	msvThemeLatinFont	Effect	vFont
AsianFont	msvThemeAsianFont	Effect	vFont
ComplexFont	msvThemeComplexFont	Effect	vFont
LineTransparency	msvThemeLineTransparency	Effect	PtgNumPct
LinePattern	msvThemeLinePattern	Effect	PtgNum
LineWeight	msvThemeLineWeight	Effect	PtgTypPt
LineRounding	msvThemeLineRounding	Effect	PtgNumI
ConnectorTransparency	msvThemeConnectorTransparency	Effect	PtgNumPct
ConnectorPattern	msvThemeConnectorPattern	Effect	PtgNum
ConnectorWeight	msvThemeConnectorWeight	Effect	PtgTypPt
ConnectorRounding	msvThemeConnectorRounding	Effect	PtgNumI
ConnectorBegin	msvThemeConnectorBegin	Effect	PtgNum
ConnectorEnd	msvThemeConnectorEnd	Effect	PtgNum
ConnectorEnd2	msvThemeConnectorEnd2.4.3.18	Effect	PtgNum
ConnectorBeginSize	msvThemeConnectorBeginSize	Effect	PtgNum
ConnectorEndSize	msvThemeConnectorEndSize	Effect	PtgNum
FillTransparency	msvThemeFillTransparency	Effect	PtgNumPct
FillPattern	msvThemeFillPattern	Effect	PtgNum

Fixed Theme Property	User Row_Type	Scheme	Value Type
ShadowTransparency	msvThemeShadowTransparency	Effect	PtgNumPct
ShadowPattern	msvThemeShadowPattern	Effect	PtgNum
ShadowStyle	msvThemeShadowStyle	Effect	PtgNum
ShadowXOffset	msvThemeShadowXOffset	Effect	PtgNumI
ShadowYOffset	msvThemeShadowYOffset	Effect	PtgNumI
ShadowMagnification	msvThemeShadowMagnification	Effect	PtgNumPct
ShadowDirection	msvThemeShadowDirection	Effect	PtgAngDD

2.5.9.12 vDynamicThemeString

The **vDynamicThemeString** custom structure is a string that specifies a dynamic theme property.

It MUST be a value from the following table.

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
0	Specifies the value of the **dk1** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.9 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeDarkColor Row_Type element.	PtgColorRGB	msvThemeDarkColor	#000000
1	Specifies the value of the **lt1** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.22 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeLightColor Row_Type element.	PtgColorRGB	msvThemeLightColor	#FFFFFF
2	Specifies the value of the **accent1** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.1 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor Row_Type element.	PtgColorRGB	msvThemeAccentColor	#C05046
3	Specifies the value of the **accent2** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section	PtgColorRGB	msvThemeAccentColor2.4.3.6	#9DBB61

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
	20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.6 Row_Type element.			
4	Specifies the value of the **accent3** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.7 Row_Type element.	PtgColorRGB	msvThemeAccent Color2.4.3.7	#AB9AC0
5	Specifies the value of the **accent4** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.8 Row_Type element.	PtgColorRGB	msvThemeAccent Color2.4.3.8	#4BACC6
6	Specifies the value of the **accent5** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.9 Row_Type element.	PtgColorRGB	msvThemeAccent Color2.4.3.9	#F59D56
7	Specifies the value of the **accent6** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.5 Row_Type element.	PtgColorRGB	msvThemeAccent Color2.4.3.5	#FFC000
8	Specifies the value of the **bkgnd** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section §A.4.1 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeBackgroundColor Row_Type element.	PtgColorRGB	msvThemeBackgr oundColor	#FFFFFF
"TextColor"	Specifies the value of the structure of a Color Cell_Type child element of a Row_Type element that has a Character Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	#000000

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
"FillColor"	Specifies the value of the structure of a FillForegnd Cell_Type element in a dynamic theme.	PtgColorRGB	Not applicable	#96AFCF
"FillColor2"	Specifies the value of the structure of a FillBkgnd Cell_Type element in a dynamic theme.	PtgColorRGB	Not applicable	#BFCEE1
"LineColor"	Specifies the value of the structure of a LineColor Cell_Type element in a dynamic theme.	PtgColorRGB	Not applicable	#1F477D
"Connector Color"	Specifies the value of the structure of a LineColor Cell_Type element in a dynamic theme.	PtgColorRGB	Not applicable	#1F477D
"ShadowCo lor"	Specifies the value of the structure of a ShdwForegnd Cell_Type element in a dynamic theme.	PtgColorRGB	Not applicable	#464646
"Dark"	Specifies the value of the **dk1** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.9 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeDarkColor Row_Type element.	PtgColorRGB	msvThemeDarkCo lor	#000000
"Light"	Specifies the value of the **lt1** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.22 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeLightColor Row_Type element.	PtgColorRGB	msvThemeLightC olor	#FFFFFF
"AccentCol or"	Specifies the value of the **accent1** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.1.1 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor Row_Type element.	PtgColorRGB	msvThemeAccent Color	#C05046
"AccentCol or2"	Specifies the value of the **accent2** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.6 Row_Type element.	PtgColorRGB	msvThemeAccent Color2.4.3.6	#9DBB61

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
"AccentColor3"	Specifies the value of the **accent3** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.7 Row_Type element.	PtgColorRGB	msvThemeAccentColor2.4.3.7	#AB9AC0
"AccentColor4"	Specifies the value of the **accent4** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.8 Row_Type element.	PtgColorRGB	msvThemeAccentColor2.4.3.8	#4BACC6
"AccentColor5"	Specifies the value of the **accent5** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.9 Row_Type element.	PtgColorRGB	msvThemeAccentColor2.4.3.9	#F59D56
"AccentColor6"	Specifies the value of the **accent6** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section 20.1.4.12 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeAccentColor2.4.3.5 Row_Type element.	PtgColorRGB	msvThemeAccentColor2.4.3.5	#FFC000
"BackgroundColor"	Specifies the value of the **bkgnd** element as specified by the **CT_Color** type specified in [ISO/IEC29500-1:2011] section §A.4.1 in a dynamic theme, or the value of the structure of the Value Cell_Type child element of an msvThemeBackgroundColor Row_Type element.	PtgColorRGB	msvThemeBackgroundColor	#FFFFFF
"LatinFont"	Specifies the value of the structure of a Font Cell_Type element in a dynamic theme.	vFont	Not applicable	Specified by a Font Cell_Type element in the root style sheet.
"AsianFont"	Specifies the value of the structure of an AsianFont Cell_Type element in a dynamic theme.	vFont	Not applicable	Specified by an AsianFont Cell_Type element in the root style

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				sheet.
"ComplexFont"	Specifies the value of the structure of a ComplexScriptFont Cell_Type element in a dynamic theme.	vFont	Not applicable	Specified by a ComplexScriptFont Cell_Type element in the root style sheet.
"LineTransparency"	Specifies the value of the structure of a LineColorTrans Cell_Type element in a dynamic theme.	PtgNumPct	Not applicable	Specified by a LineColorTrans Cell_Type element in the root style sheet.
"LinePattern"	Specifies the value of the structure of a LinePattern Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a LinePattern Cell_Type element in the root style sheet.
"LineWeight"	Specifies the value of the structure of a LineWeight Cell_Type element in a dynamic theme.	PtgTypPt	Not applicable	Specified by a LineWeight Cell_Type element in the root style sheet.
"LineRounding"	Specifies the value of the structure of a Rounding Cell_Type element in a dynamic theme.	PtgNumI	Not applicable	Specified by a Rounding Cell_Type element in the root style sheet.
"LineCap"	Specifies the value of the structure of a LineCap Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a LineCap Cell_Type element in the root style sheet.
"LineCompoundType"	Specifies the value of the structure of a CompoundType Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a CompoundType Cell_Type element in the root style sheet.
"LineGradientEnabled"	Specifies the value of the structure of a LineGradientEnabled Cell_Type element in a dynamic theme.	PtgBool	Not applicable	Specified by a LineGradientEnabled Cell_Type element in the root style sheet.
"LineGradientDir"	Specifies the value of the structure of a LineGradientDir Cell_Type	PtgNum	Not applicable	Specified by a LineGradientDi

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
	element in a dynamic theme.			r Cell_Type element in the root style sheet.
"LineGradientAngle"	Specifies the value of the structure of a LineGradientAngle Cell_Type element in a dynamic theme.	vAngle	Not applicable	Specified by a LineGradientAngle Cell_Type element in the root style sheet.
"Connector Transparency"	Specifies the value of the structure of a LineColorTrans Cell_Type element in a dynamic theme.	PtgNumPct	Not applicable	Specified by a LineColorTrans Cell_Type element in the root style sheet.
"Connector Pattern"	Specifies the value of the structure of a LinePattern Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a LinePattern Cell_Type element in the root style sheet.
"Connector Weight"	Specifies the value of the structure of a LineWeight Cell_Type element in a dynamic theme.	PtgTypPt	Not applicable	Specified by a LineWeight Cell_Type element in the root style sheet.
"Connector Rounding"	Specifies the value of the structure of a Rounding Cell_Type element in a dynamic theme.	PtgNumI	Not applicable	Specified by a Rounding Cell_Type element in the root style sheet.
"Connector Begin"	Specifies the value of the structure of a BeginArrow Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a BeginArrow Cell_Type element in the root style sheet.
"Connector End"	Specifies the value of the structure of a EndArrow Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by an EndArrow Cell_Type element in the root style sheet.
"Connector BeginSize"	Specifies the value of the structure of a BeginArrowSize Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a BeginArrowSize Cell_Type element in the root style sheet.

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
"ConnectorEndSize"	Specifies the value of the structure of a EndArrowSize Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by an EndArrowSize Cell_Type element in the root style sheet.
"FillTransparency"	Specifies the value of the structure of a FillForegndTrans Cell_Type element in a dynamic theme.	PtgNumPct	Not applicable	Specified by a FillForegndTrans Cell_Type element in the root style sheet.
"FillTransparencyBkgnd"	Specifies the value of the structure of a FillBkgndTrans Cell_Type element in a dynamic theme.	PtgNumPct	Not applicable	Specified by a FillBkgndTrans Cell_Type element in the root style sheet.
"FillPattern"	Specifies the value of the structure of a FillPattern Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a FillPattern Cell_Type element in the root style sheet.
"FillGradientEnabled"	Specifies the value of the structure of a FillGradientEnabled Cell_Type element in a dynamic theme.	PtgBool	Not applicable	Specified by a FillGradientEnabled Cell_Type element in the root style sheet.
"FillGradientDir"	Specifies the value of the structure of a FillGradientDir Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a FillGradientDir Cell_Type element in the root style sheet.
"FillGradientAngle"	Specifies the value of the structure of a FillGradientAngle Cell_Type element in a dynamic theme.	vAngle	Not applicable	Specified by a FillGradientAngle Cell_Type element in the root style sheet.
"RotateGradientWithShape"	Specifies the value of the structure of a RotateGradientWithShape Cell_Type element in a dynamic theme.	PtgBool	Not applicable	Specified by a RotateGradientWithShape Cell_Type element in the root style sheet.
"UseGroup	Specifies the value of the structure of a UseGroupGradient Cell_Type	PtgBool	Not applicable	Specified by a UseGroupGrad

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
Gradient"	element in a dynamic theme.			ient Cell_Type element in the root style sheet.
"ShadowTransparency"	Specifies the value of the structure of a ShdwForegndTrans Cell_Type element in a dynamic theme.	PtgNumPct	Not applicable	Specified by a ShdwForegndTrans Cell_Type element in the root style sheet.
"ShadowPattern"	Specifies the value of the structure of a ShdwPattern Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a ShdwPattern Cell_Type element in the root style sheet.
"ShadowStyle"	Specifies the value of the structure of a ShapeShdwType Cell_Type element in a dynamic theme.	PtgByte	Not applicable	Specified by a ShapeShdwType Cell_Type element in the root style sheet.
"ShadowXOffset"	Specifies the value of the structure of a ShapeShdwOffsetX Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a ShapeShdwOffsetX Cell_Type element in the root style sheet.
"ShadowYOffset"	Specifies the value of the structure of a ShapeShdwOffsetY Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a ShapeShdwOffsetY Cell_Type element in the root style sheet.
"ShadowMagnification"	Specifies the value of the structure of a ShapeShdwScaleFactor Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a ShapeShdwScaleFactor Cell_Type element in the root style sheet.
"ShadowDirection"	Specifies the value of the structure of a ShapeShdwObliqueAngle Cell_Type element in a dynamic theme.	vAngle	Not applicable	Specified by a ShapeShdwObliqueAngle Cell_Type element in the root style sheet.
"ShadowBlur"	Specifies the value of the structure of a ShapeShdwBlur Cell_Type element in a dynamic theme.	PtgTypPt	Not applicable	Specified by a ShapeShdwBlur Cell_Type element in the root style

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				sheet.
"BevelTopType"	Specifies the value of the structure of a BevelTopType Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a BevelTopType Cell_Type element in the root style sheet.
"BevelTopWidth"	Specifies the value of the structure of a BevelTopWidth Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a BevelTopWidth Cell_Type element in the root style sheet.
"BevelTopHeight"	Specifies the value of the structure of a BevelTopHeight Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a BevelTopHeight Cell_Type element in the root style sheet.
"BevelContourColor"	Specifies the value of the structure of a BevelContourColor Cell_Type element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a BevelContourColor Cell_Type element in the root style sheet.
"BevelContourSize"	Specifies the value of the structure of a BevelContourSize Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a BevelContourSize Cell_Type element in the root style sheet.
"BevelMaterial"	Specifies the value of the structure of a BevelMaterialType Cell_Type element in a dynamic theme.	PtgByte	Not applicable	Specified by a BevelMaterialType Cell_Type element in the root style sheet.
"BevelLightingType"	Specifies the value of the structure of a BevelLightingType Cell_Type element in a dynamic theme.	PtgByte	Not applicable	Specified by a BevelLightingType Cell_Type element in the root style sheet.
"BevelLightingAngle"	Specifies the value of the structure of a BevelLightingAngle Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a BevelLightingAngle Cell_Type element in the root style sheet.
"GlowSize"	Specifies the value of the structure of a GlowSize Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a GlowSize Cell_Type

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				element in the root style sheet.
"GlowColor"	Specifies the value of the structure of a GlowColor Cell_Type element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GlowColor Cell_Type element in the root style sheet.
"GlowColorTrans"	Specifies the value of the structure of a GlowColorTrans Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a GlowColorTrans Cell_Type element in the root style sheet.
"ReflectionBlur"	Specifies the value of the structure of a ReflectionBlur Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a ReflectionBlur Cell_Type element in the root style sheet.
"ReflectionDist"	Specifies the value of the structure of a ReflectionDist Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a ReflectionDist Cell_Type element in the root style sheet.
"ReflectionSize"	Specifies the value of the structure of a ReflectionSize Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a ReflectionSize Cell_Type element in the root style sheet.
"ReflectionTrans"	Specifies the value of the structure of a ReflectionTrans Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a ReflectionTrans Cell_Type element in the root style sheet.
"SoftEdgesSize"	Specifies the value of the structure of a SoftEdgesSize Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a SoftEdgesSize Cell_Type element in the root style sheet.
"SketchAmount"	Specifies the value of the structure of a SketchAmount Cell_Type element in a dynamic theme.	PtgInt	Not applicable	Specified by a SketchAmount Cell_Type element in the root style sheet.
"SketchEna	Specifies the value of the structure	PtgBool	Not applicable	Specified by a

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
bled"	of a SketchEnabled Cell_Type element in a dynamic theme.			SketchEnabled Cell_Type element in the root style sheet.
"SketchFill Change"	Specifies the value of the structure of a SketchFillChange Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a SketchFillChange Cell_Type element in the root style sheet.
"SketchLineChange"	Specifies the value of the structure of a SketchLineChange Cell_Type element in a dynamic theme.	PtgNum	Not applicable	Specified by a SketchLineChange Cell_Type element in the root style sheet.
"SketchLineWeight"	Specifies the value of the structure of a SketchLineWeight Cell_Type element in a dynamic theme.	vLength	Not applicable	Specified by a SketchLineWeight Cell_Type element in the root style sheet.
"LineStop1 Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a LineGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop1 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a LineGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a LineGradient Section_Type parent element in the root style sheet.

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
"LineStop1 Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a LineGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop2 Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a LineGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop2 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a LineGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop2 Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a LineGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				whose **IX** attribute value is equal to one that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop3 Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop3 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop3 Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a LineGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a LineGradient Section_Type parent element in the

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				root style sheet.
"LineStop4 Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop4 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop4 Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a LineGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop5 Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a LineGradient	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
	Section_Type parent element in a dynamic theme.			element whose **IX** attribute value is equal to 4 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop5 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop5 Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a LineGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop6 Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a LineGradient Section_Type parent

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				element in the root style sheet.
"LineStop6 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop6 Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a LineGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop7 Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop7 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX**	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
	attribute value is equal to 6 that has a LineGradient Section_Type parent element in a dynamic theme.			child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop7 Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a LineGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop8 Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop8 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a LineGradient

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				Section_Type parent element in the root style sheet.
"LineStop8 Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a LineGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop9 Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop9 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop9	Specifies the value of the structure of a GradientStopPosition Cell_Type	vLength	Not applicable	Specified by a GradientStopP

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
Position"	child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a LineGradient Section_Type parent element in a dynamic theme.			osition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop10Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop10Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a LineGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a LineGradient Section_Type parent element in the root style sheet.
"LineStop10Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a LineGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 10

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				that has a LineGradient Section_Type parent element in the root style sheet.
"FillStop1Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a FillGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop1Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a FillGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop1Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a FillGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to zero that has a FillGradient Section_Type parent element in the root style sheet.

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
"FillStop2Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a FillGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop2Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a FillGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop2Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a FillGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to one that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop3Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX**

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				attribute value is equal to two that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop3Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop3Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a FillGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 2 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop4Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a FillGradient Section_Type parent element in the root style

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				sheet.
"FillStop4Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop4Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a FillGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 3 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop5Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop5Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a FillGradient Section_Type	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
	parent element in a dynamic theme.			element whose **IX** attribute value is equal to 4 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop5Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a FillGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 4 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop6Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop6Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a FillGradient Section_Type parent

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				element in the root style sheet.
"FillStop6Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a FillGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 5 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop7Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop7Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop7Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
	equal to 6 that has a FillGradient Section_Type parent element in a dynamic theme.			child element of a Row_Type element whose **IX** attribute value is equal to 6 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop8Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop8Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop8Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a FillGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 7 that has a FillGradient

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
				Section_Type parent element in the root style sheet.
"FillStop9Color"	Specifies the value of the structure of a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgColorRGB	Not applicable	Specified by a GradientStopColor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop9Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop9Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a FillGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 8 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop10	Specifies the value of the structure of a GradientStopColor Cell_Type	PtgColorRGB	Not applicable	Specified by a GradientStopC

Dynamic Theme Property	Description	Value	User Row_Type	No Theme
Color"	child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a FillGradient Section_Type parent element in a dynamic theme.			olor Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop10 Transparency"	Specifies the value of the structure of a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a FillGradient Section_Type parent element in a dynamic theme.	PtgNum	Not applicable	Specified by a GradientStopColorTrans Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a FillGradient Section_Type parent element in the root style sheet.
"FillStop10 Position"	Specifies the value of the structure of a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a FillGradient Section_Type parent element in a dynamic theme.	vLength	Not applicable	Specified by a GradientStopPosition Cell_Type child element of a Row_Type element whose **IX** attribute value is equal to 9 that has a FillGradient Section_Type parent element in the root style sheet.

2.5.9.13 vThemeColor

The **vThemeColor** custom structure is a PtgColorRGB that specifies a fixed theme fixed color scheme.

Fixed theme property values for each of the fixed color schemes are specified in the following table.

Fixed Theme Index	Text Color	Fill Color	Fill Color 2	Line Color	Connector Color	Shadow Color
0	#000000	#96AFCF	#BFCEE1	#1F477D	#1F477D	#464646
1	#000000	#FFFFFF	#EAEAEA	#000000	#000000	#C0C0C0
2	#000000	#9AB3D1	#C4D2E3	#1F497D	#1F497D	#464646
3	#000000	#A0BFD8	#BBD0E1	#69544B	#775F55	#827068
4	#000000	#6AC5DB	#8CD2E3	#464646	#464646	#464646
5	#000000	#6FBBCE	#A0D1DE	#4F271C	#4F271C	#7E6C67
6	#000000	#8FB6C2	#A8C7D0	#3B3B3B	#3B3B3B	#ABABAD
7	#000000	#A5B592	#C0CBB4	#444D26	#444D26	#444D26
8	#000000	#95B998	#ADC9AF	#676A55	#676A55	#868A6F
9	#000000	#CAB86F	#DCCD92	#525056	#5C5A61	#7D7B82
10	#000000	#EEAF51	#F2C179	#4E3B30	#4E3B30	#745847
11	#000000	#F0AD00	#FFC733	#5A6378	#5A6378	#5A6378
12	#000000	#FB944F	#FFAD77	#575F6D	#575F6D	#7A838F
13	#000000	#F89430	#FAB067	#323232	#323232	#626262
14	#000000	#ED744B	#F19272	#696464	#696464	#696464
15	#000000	#D16349	#E19A88	#646B86	#646B86	#464646
16	#000000	#BF8096	#D1A5B5	#653445	#3C1C31	#202020
17	#000000	#FDA4CA	#FFCEE3	#620027	#790031	#606060
18	#000000	#A3ABCD	#B5BACE	#46465D	#46465D	#6F6F84
19	#000000	#9C9DC7	#BCBDDA	#444444	#444444	#444444
20	#000000	#82B4E1	#91C6F7	#000000	#000000	#1C4E5B
21	#000000	#7FD13B	#6DBD2D	#000000	#4E5B6F	#5E646F
22	#000000	#9AB3D1	#C6D3E3	#1F497D	#1F497D	#576980
23	#FFFFFF	#4F81BD	#4172AD	#EEECE1	#EEECE1	#000000
24	#000000	#A0BFD8	#BBD0E1	#69544B	#775F55	#827068
25	#FFFFFF	#94B6D2	#77A2C4	#EBDDC3	#EBDDC3	#1F497D
26	#000000	#6AC5DB	#8CD2E3	#464646	#464646	#464646
27	#FFFFFF	#2DA2BF	#25869E	#DEF5FA	#DEF5FA	#000000
28	#000000	#A5B592	#C0CBB4	#444D26	#444D26	#444D26
29	#FFFFFF	#A5B592	#B9C5AB	#FEFAC9	#FEFAC9	#000000
30	#000000	#95B998	#ADC9AF	#676A55	#676A55	#868A6F

Fixed Theme Index	Text Color	Fill Color	Fill Color 2	Line Color	Connector Color	Shadow Color
31	#FFFFFF	#72A376	#6C886F	#EAEBDE	#EAEBDE	#000000
32	#000000	#ED744B	#F19272	#696464	#696464	#696464
33	#FFFFFF	#D34817	#B6421B	#E9E5DC	#E9E5DC	#000000
34	#000000	#FDA4CA	#FFCEE3	#620027	#790031	#606060
35	#FFFFFF	#FF388C	#FF599F	#D2D2D2	#D2D2D2	#000000
36	#000000	#FFFFFF	#F0F0F0	#404040	#404040	#CDCDCD
37	#000000	#A9A57C	#C2C0A3	#7B7859	#7B7859	#7B7859
38	#FFFFFF	#797B7E	#9B9D9F	#CDD7D9	#CDD7D9	#000000
39	#000000	#93A299	#B3BDB7	#6B766F	#6B766F	#6B766F
40	#000000	#94C600	#C0FF02	#6B9100	#6B9100	#6B9100
41	#000000	#7A7A7A	#A7A7A7	#585858	#585858	#585858
42	#FFFFFF	#6F6F74	#A7A7A7	#E3DCCF	#E3DCCF	#A7A7A7
43	#000000	#98C723	#BBE155	#6E9217	#6E9217	#6E9217
44	#000000	#93A299	#ADB8B1	#6B766F	#6B766F	#6B766F
45	#FFFFFF	#629DD1	#98BEE0	#ACCBF9	#ACCBF9	#000000
46	#FFFFFF	#6076B4	#8A9AC8	#E4E9EF	#E4E9EF	#000000
47	#000000	#C66951	#D89685	#914B39	#914B39	#914B39
48	#FFFFFF	#873624	#CC5840	#ECE9C6	#ECE9C6	#000000
49	#000000	#7E97AD	#9FB1C1	#5B6E7E	#5B6E7E	#5B6E7E
50	#FFFFFF	#AD0101	#FE0707	#DEDEE0	#DEDEE0	#000000
51	#000000	#9E8E5C	#B4A77E	#736741	#736741	#736741
52	#FFFFFF	#838D9B	#A6ADB7	#5F6671	#5F6671	#000000
53	#000000	#FDA023	#FEB858	#975E11	#975E11	#975E11
54	#FFFFFF	#4E67C8	#8091D7	#B4DCFA	#B4DCFA	#000000
55	#FFFFFF	#759AA5	#92AFB8	#DFE6D0	#DFE6D0	#000000
56	#000000	#31B6FD	#68C9FD	#165D83	#165D83	#165D83

Fixed Theme Index	Accent Color	Accent Color 2	Accent Color 3	Accent Color 4	Accent Color 5	Background Color
0	#C05046	#9DBB61	#AB9AC0	#4BACC6	#F59D56	#FFFFFF
1	#DDDDDD	#C0C0C0	#B2B2B2	#969696	#808080	#FFFFFF

Fixed Theme Index	Accent Color	Accent Color 2	Accent Color 3	Accent Color 4	Accent Color 5	Background Color
2	#CE7977	#A5C06B	#AB9AC0	#69BBCF	#F89C52	#FFFFFF
3	#DD8047	#A5AB81	#D8B25C	#7BA79D	#968C8C	#FFFFFF
4	#EB747A	#F08E59	#729CD6	#8A9ECE	#B96F80	#FFFFFF
5	#FDC022	#DD6868	#9FBD62	#D07837	#768AB8	#FFFFFF
6	#EDCB0C	#9F9CB1	#9DAC8A	#ABA085	#979CA4	#FFFFFF
7	#F3A447	#E7BC29	#D092A7	#9C85C0	#809EC2	#FFFFFF
8	#B0CCB0	#A8CDD7	#C0BEAF	#CEC597	#E8B7B7	#FFFFFF
9	#A0B489	#74B7CD	#7592D5	#8573CC	#B78AC4	#FFFFFF
10	#BB846F	#B99288	#C3986D	#A99E81	#D39556	#FFFFFF
11	#60B5CC	#E66C7D	#6BB76D	#E88651	#C64847	#FFFFFF
12	#8CA9DF	#E9725C	#F3D14B	#B5BFD9	#8A8E95	#FFFFFF
13	#CE5977	#73A6C4	#7DAE73	#8D76A5	#BD975E	#FFFFFF
14	#DD6453	#A79C9D	#AF7E6D	#986D6D	#AB9978	#FFFFFF
15	#CCB400	#8CADAE	#8C7B70	#8FB08C	#D19049	#FFFFFF
16	#B678C2	#DE6C36	#F9C18A	#D195C0	#FA9970	#FFFFFF
17	#FF4A93	#E659D5	#C586E3	#69A6F8	#89A4DC	#FFFFFF
18	#9FB8CD	#D2DA7A	#FADA7A	#B88472	#A28A82	#FFFFFF
19	#97C7CC	#BE78C0	#D28759	#B57C57	#7EA9C5	#FFFFFF
20	#09B0F0	#0ED6E0	#12DEA6	#7CC97A	#ABC55F	#FFFFFF
21	#F059A0	#FEB80A	#52C9BA	#37BEE6	#8EA2D2	#FFFFFF
22	#CF7A78	#A6C269	#AB9AC0	#69BBCF	#F89C52	#EEECE1
23	#C0504D	#9BBB59	#8064A2	#4BACC6	#F79646	#1F497D
24	#DD8047	#A5AB81	#D8B25C	#7BA79D	#968C8C	#EBDDC3
25	#DD8047	#A5AB81	#D8B25C	#7BA79D	#968C8C	#775F55
26	#EB747A	#F08E59	#729CD6	#8A9ECE	#B96F80	#DEF5FA
27	#DA1F28	#EB641B	#39639D	#474B78	#7D3C4A	#464646
28	#F3A447	#E7BC29	#D092A7	#9C85C0	#809EC2	#FEFAC9
29	#F3A447	#E7BC29	#D092A7	#9C85C0	#809EC2	#444D26
30	#B0CCB0	#A8CDD7	#C0BEAF	#CEC597	#E8B7B7	#EAEBDE
31	#B0CCB0	#A8CDD7	#C0BEAF	#CEC597	#E8B7B7	#676A55
32	#DD6453	#A79C9D	#AF7E6D	#986D6D	#AB9978	#E9E5DC

Fixed Theme Index	Accent Color	Accent Color 2	Accent Color 3	Accent Color 4	Accent Color 5	Background Color
33	#9B2D1F	#918485	#956251	#855D5D	#A28E6A	#696464
34	#FF4A93	#E659D5	#C586E3	#69A6F8	#89A4DC	#D2D2D2
35	#E40059	#9C007F	#68007F	#005BD3	#00349E	#666666
36	#EA9651	#B3C283	#6B9BC7	#4E66B2	#8976AC	#FFFFFF
37	#9CBEBD	#D2CB6C	#95A39D	#C89F5D	#B1A089	#FFFFFF
38	#F96A1B	#08A1D9	#7C984A	#C2AD8D	#506E94	#434342
39	#CF543F	#B5AE53	#848058	#E8B54D	#786C71	#FFFFFF
40	#71685A	#FF6700	#909465	#956B43	#FEA022	#FFFFFF
41	#F5C201	#526DB0	#989AAC	#DC5924	#B4B392	#FFFFFF
42	#A7B789	#BEAE98	#92A9B9	#9C8265	#8D6974	#000000
43	#59B0B9	#DEAE00	#B77BB4	#E0773C	#A98D63	#FFFFFF
44	#AD8F67	#726056	#4C5A6A	#808DA0	#79463D	#FFFFFF
45	#297FD5	#7F8FA9	#4A66AC	#5AA2AE	#9D90A0	#242852
46	#9C5252	#E68422	#846648	#63891F	#758085	#2F5897
47	#BF974D	#928B70	#87706B	#94734E	#6F777D	#FFFFFF
48	#D6862D	#D0BE40	#877F6C	#972109	#AEB795	#895D1D
49	#CC8E60	#7A6A60	#B4936D	#67787B	#9D936F	#FFFFFF
50	#726056	#AC956E	#808DA9	#424E5B	#730E00	#303030
51	#A09781	#85776D	#AEAFA9	#8D878B	#6B6149	#FFFFFF
52	#D2610C	#80716A	#94147C	#5D5AD2	#6F6C7D	#283138
53	#AA2B1E	#71685C	#64A73B	#EB5605	#B9CA1A	#FFFFFF
54	#5ECCF3	#A7EA52	#5DCEAF	#FF8021	#F14124	#212745
55	#CFC60D	#99987F	#90AC97	#FFAD1C	#B9AB6F	#1D3641
56	#4584D3	#5BD078	#A5D028	#F5C040	#05E0DB	#FFFFFF

2.5.9.14 vThemeEffect

The **vThemeEffect** custom structure is a vNum or vFont that specifies a fixed theme fixed effect scheme.

Fixed theme property values for each of the fixed effect schemes are specified in the following table. Asian Font and Complex Script Font property values vary based on the **Language** property of the web drawing, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML Part, and are specified in the Asian and Complex Font Properties section.

Fixed Theme Index	0	1	2	3	4
Latin Font	Calibri	Arial	Calibri	Arial	Arial
Asian Font	Asian and Complex Font Properties				
Complex Font					
Line Transparency	0	0.35	0	0	0
Line Pattern	1	1	1	1	1
Line Weight	0.003472222	0.000694444	0.003472222	0.016666667	0.006944444
Line Rounding	0	0	0	0.0625	0
Connector Transparency	0	0.25	0	0	0
Connector Pattern	1	1	1	1	1
Connector Weight	0.003472222	0.010416667	0.013888889	0.013888889	0.003472222
Connector Rounding	0	0	0	0.0625	0
Connector Begin	0	0	0	0	0
Connector End	4	4	5	5	4
Connector End 2	4	4	5	5	4
Connector Begin Size	2	1	2	2	2
Connector End Size	2	1	2	4	2
Fill Transparency	0	0.15	0	0	0
Fill Pattern	1	1	28	36	30
Shadow Transparency	0	0	0.5	0.6	0.5
Shadow Pattern	0	0	1	1	1
Shadow Style	0	0	1	1	13
Shadow Offset X	0	0.125	0.02	0.025	0
Shadow Offset Y	0	-0.125	-0.02	-0.025	-0.03
Shadow Magnification	1	1	1	1	1
Shadow Direction	0	0	0	0	0

Fixed Theme Index	5	6	7	8	9
Latin Font	Cambria	Arial Rounded MT Bold	Corbel	Trebuchet MS	Trebuchet MS
Asian Font	Asian and Complex Font Properties				
Complex Font					

Fixed Theme Index	5	6	7	8	9
Line Transparency	0.25	0	0	0	0.8
Line Pattern	1	1	1	1	1
Line Weight	0.013888889	0.022916667	0.010416667	0.03125	0.041666667
Line Rounding	0.0875	0.0173	0.0925	0	0.0625
Connector Transparency	0.25	0	0	0	0
Connector Pattern	1	1	1	1	1
Connector Weight	0.013888889	0.016666667	0.010416667	0.020833333	0.016666667
Connector Rounding	0.0875	0.0173	0.0925	0	0
Connector Begin	0	0	0	0	0
Connector End	5	2	8	4	4
Connector End 2	5	2	8	4	4
Connector Begin Size	0	3	2	2	2
Connector End Size	0	3	2	2	2
Fill Transparency	0	0.35	0.35	0	0
Fill Pattern	29	39	39	37	8
Shadow Transparency	0	0.75	0.7	0.35	0.35
Shadow Pattern	0	1	1	0	1
Shadow Style	0	1	1	0	1
Shadow Offset X	0.125	0.0425	0.03	0.125	0.012
Shadow Offset Y	-0.125	-0.0425	-0.03	-0.125	-0.012
Shadow Magnification	1	0.98	0.99	1	1
Shadow Direction	0	0	0	0	0

Fixed Theme Index	10	11	12	13	14
Latin Font	Arial	Franklin Gothic Demi	Constantia	Arial	Verdana
Asian Font	Asian and Complex Font Properties				
Complex Font					
Line Transparency	0.4	0	0	0.3	0
Line Pattern	1	1	1	1	1
Line Weight	0.024305556	0.013888889	0.010416667	0.016666667	0.020833333

Fixed Theme Index	10	11	12	13	14
Line Rounding	0	0.0625	0	0.0625	0.035
Connector Transparency	0.3	0	0	0	0.15
Connector Pattern	1	23	1	10	1
Connector Weight	0.010416667	0.003472222	0.003472222	0.016666667	0.020833333
Connector Rounding	0	0.0625	0.0625	0.0625	0.035
Connector Begin	0	0	11	0	0
Connector End	13	5	4	11	4
Connector End 2	13	5	4	11	4
Connector Begin Size	0	3	1	0	0
Connector End Size	0	3	2	0	0
Fill Transparency	0.25	0.15	0	0.2	0
Fill Pattern	8	24	20	19	1
Shadow Transparency	0.65	0.5	0.5	0.7	0.65
Shadow Pattern	1	0	0	0	1
Shadow Style	1	1	0	1	2
Shadow Offset X	0.02	0.0625	0.0625	0.0625	0
Shadow Offset Y	-0.02	-0.0625	-0.0625	-0.0625	0
Shadow Magnification	1	1	1	1	0.35
Shadow Direction	0	0	0	0	1.308996939

Fixed Theme Index	15	16
Latin Font	Arial Rounded MT Bold	Calibri
Asian Font	Asian and Complex Font Properties	
Complex Font		
Line Transparency	0	0
Line Pattern	1	1
Line Weight	0.003472222	0.003472222
Line Rounding	0.125	0
Connector Transparency	0	0
Connector Pattern	1	1
Connector Weight	0.010416667	0.013888889

Fixed Theme Index	15	16
Connector Rounding	0.3125	0
Connector Begin	10	0
Connector End	10	5
Connector End 2	10	5
Connector Begin Size	1	2
Connector End Size	3	2
Fill Transparency	0	0
Fill Pattern	28	30
Shadow Transparency	0	0.5
Shadow Pattern	0	1
Shadow Style	1	1
Shadow Offset X	0.0625	0.02
Shadow Offset Y	-0.0625	-0.02
Shadow Magnification	1	1
Shadow Direction	0	0

2.5.9.14.1 Asian and Complex Font Properties

The Asian Font and Complex Script Font property values are determined based on the [ISO-15924] script tag of the **Language** property, specified in [ISO/IEC29500-2:2011] section 11, from the Core XML Part of a web drawing.

To lookup which font is used for a fixed effect scheme, map the script tag from the first column to the fixed theme index in the first row. The values of the cells in this table correspond to font indexes in the following table.

Script Tag	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
HANS	0	0	1	0	0	1	0	2	1	1	0	0	3	0	1	0	1
HANT	0	4	4	4	4	5	4	5	4	4	4	4	5	4	4	4	4
JPAN	0	6	6	6	6	7	8	9	12	12	6	11	10	6	6	8	6
HANG	0	14	15	14	14	15	14	16	18	18	14	16	17	14	14	14	15
ARAB	0	25	21	25	25	21	24	25	24	24	25	24	21	25	24	24	21
HEBR	0	25	26	25	25	21	25	26	30	30	25	29	28	25	31	29	26
THAI	0	33	34	33	33	35	33	37	40	40	33	39	38	33	40	33	34
ETHI	0	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41	41

Script Tag	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
BENG	0	42	42	42	42	42	42	42	42	42	42	42	42	42	42	42	42
GUJR	0	43	43	43	43	43	43	43	43	43	43	43	43	43	43	43	43
KHMR	0	45	45	45	44	45	44	44	45	44	45	45	45	44	44	45	45
KNDA	0	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46
GURU	0	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47
CANS	0	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48
CHER	0	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
YIII	0	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
TIBT	0	51	51	51	51	51	51	51	51	51	51	51	51	51	51	51	51
THAA	0	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52	52
DEVA	0	53	53	53	53	53	53	53	53	53	53	53	53	53	53	53	53
TELU	0	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
TAML	0	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
SYRC	0	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
ORYA	0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
MLYM	0	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
LAOO	0	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23
SINH	0	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27
MONG	0	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
VIET	0	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21
UIGH	0	54	54	54	54	54	54	54	54	54	54	54	54	54	54	54	54
GEOR	0	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55	55

Font Index	Font
0	黑体
1	宋体
2	KaiTi_GB2312
3	FangSong_GB2312
4	Microsoft JhengHei
5	PMingLiU
6	MS PGothic

Font Index	Font
7	MS PMincho
8	HGMaruGothicMPRO
9	HGSoeiKakupoptai
10	HGMinchoE
11	HGSoeiKakugothicUB
12	HGGothicM
13	Gautami
14	Gulim
15	Malgun Gothic
16	Dotum
17	Batang
18	HYGothic-Extra
19	Estrangelo Edessa
20	Kalinga
21	Times New Roman
22	Kartika
23	DokChampa
24	Tahoma
25	Arial
26	Levenim MT
27	Iskoola Pota
28	David
29	Aharoni
30	Miriam
31	Courier New
32	Mongolian Baiti
33	Cordia New
34	Browallia New
35	Angsana New
36	Latha
37	DilleniaUPC
38	EucrosiaUPC

Font Index	Font
39	JasmineUPC
40	FreesiaUPC
41	Nyala
42	Vrinda
43	Shruti
44	MoolBoran
45	DaunPenh
46	Tunga
47	Raavi
48	Euphemia
49	Plantagenet Cherokee
50	Microsoft Yi Baiti
51	Microsoft Himalaya
52	MV Boli
53	Mangal
54	Microsoft Uighur
55	Sylfaen

2.5.9.15 vUnitString

The **vUnitString** custom structure is a string that specifies the unit of measure for a numeric value.

It MUST be a value from the following table.

Value	Unit of Measure
IN	Inches
FT	Feet
YD	Yards
MI	Miles
NM	Nautical miles
MM	Millimeters
CM	Centimeters
M	Meters
KM	Kilometers

Value	Unit of Measure
P	Picas
PT	Points
C	Ciceros
D	Didots
RAD	Radians
DEG	Degrees
°	Degrees, minutes, and seconds
’	Minutes
"	Seconds
DATE	Date and time
%	Percent
IN_F	Fractional inches
MI_F	Fractional miles
FEET/INCH	Feet and inches
PICAPOINTS	Picas and points
CICERO/DIDOT	Ciceros and didots
SQ IN	Square inches
SQ FT	Square feet
SQ MI	Square miles
ACRE	Acres
HA	Hectares
SQ MM	Square millimeters
SQ CM	Square centimeters
SQ M	Square meters
SQ KM	Square kilometers
EW	Elapsed weeks
ED	Elapsed days
EH	Elapsed hours
EM	Elapsed minutes
ES	Elapsed seconds
DE	Elapsed days
DFT	Points

Value	Unit of Measure
DT	Points
DFA	Degrees
DA	Degrees
DFL	Inches
DL	Inches
DFP	The unit of DrawingScale in drawing page
DP	The unit of DrawingScale in drawing page

3 Structure Examples

This section provides examples describing some of the parts in two simple web drawings.

3.1 Document with a Shape on a Page

This section describes some of the contents of the Document XML Part, Pages XML Part, and Page XML Part for a web drawing consisting of one drawing page with a rectangle with blue fill color and the text "Sample Text" on it:

3.1.1 Document XML Part

The following is an example of a Document XML part.

```
<?xml version="1.0"?>
<VisioDocument xmlns="http://schemas.microsoft.com/office/visio/2011/1/core"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xml:space="preserve">
  <FaceNames>
    <FaceName NameU="Calibri"/>
  </FaceNames>
  <StyleSheets>
    <StyleSheet ID="0" NameU="No Style" Name="No Style">
      <Cell N="EnableLineProps" V="1"/>
      <Cell N="EnableFillProps" V="1"/>
      <Cell N="EnableTextProps" V="1"/>
      <Cell N="LineWeight" V="0.01041666666666667"/>
```

```
        <Cell N="LineColor" V="0"/>
        <Cell N="LinePattern" V="1"/>
        <Cell N="FillForegnd" V="1"/>
        <Cell N="FillPattern" V="1"/>
        <Cell N="VerticalAlign" V="1"/>
        <Section N="Character">
          <Row IX="0">
            <Cell N="Font" V="Calibri"/>
            <Cell N="Color" V="0"/>
            <Cell N="FontScale" V="1"/>
            <Cell N="Size" V="0.1666666666666667"/>
          </Row>
        </Section>
        <Section N="Paragraph">
          <Row IX="0">
            <Cell N="IndFirst" V="0"/>
            <Cell N="IndLeft" V="0"/>
            <Cell N="IndRight" V="0"/>
            <Cell N="SpLine" V="-1.2"/>
            <Cell N="HorzAlign" V="1"/>
            <Cell N="Bullet" V="0"/>
            <Cell N="BulletStr" V=""/>
            <Cell N="BulletFont" V="0"/>
            <Cell N="BulletFontSize" V="-1"/>
            <Cell N="TextPosAfterBullet" V="0"/>
            <Cell N="Flags" V="0"/>
          </Row>
        </Section>
        <Section N="Tabs">
          <Row IX="0"/>
        </Section>
      </StyleSheet>
    </StyleSheets>
    <DocumentSheet NameU="TheDoc" Name="TheDoc" LineStyle="0" FillStyle="0" TextStyle="0">
      <Cell N="DocLangID" V="en-US"/>
    </DocumentSheet>
  </VisioDocument>
```

The following table provides more information about element and attribute values found in the preceding sample **Document XML**.

Element	Attributes	Notes
FaceName	NameU="Calibri"	The name of the font is "Calibri".
StyleSheet	ID="0" NameU="No Style" Name="No Style"	The ID attribute of the style sheet is zero. The language-dependent name of the style sheet is "No Style". The language-independent name of the style sheet is "No Style".
Cell	N="EnableLineProps" V="1"	The line property of this style sheet will be inherited by shapes using this style sheet.
Cell	N="EnableFillProps" V="1"	The fill property of this style sheet will be inherited by shapes using this style sheet.
Cell	N="EnableTextProps" V="1"	The text properties of this style sheet will be inherited by shapes using this style sheet.
Cell	N="LineWeight" V="0.01041666666666667"	Unless overridden, the line thickness of shapes using this style sheet is 0.01041666666666667 inches (3/4 pt).

Element	Attributes	Notes
Cell	N="LineColor" V="0"	Unless overridden, the color of the line of shapes using this style sheet is the color at index zero in the color table (black).
Cell	N="LinePattern" V="1"	The line pattern is solid line.
Cell	N="FillForegnd" V="1"	Unless overridden, the color of the fill of shapes using this style sheet is the color at index one in the color table (white).
Cell	N="FillPattern" V="1"	Unless overridden, fills of shapes using this style sheet will have a solid fill color.
Cell	N="VerticalAlign" V="1"	Unless overridden, the text of shapes using this style sheet will have a center vertical alignment.
Section	N="Character"	Cells under this tag relate to character properties.
Row	IX="0"	This row describes character properties starting at index zero.
Cell	N="Font" V="Calibri"	Unless overridden, characters in text runs of shapes using this style sheet use the font named "Calibri".
Cell	N="Color" V="0"	Unless overridden, characters in text runs of shapes using this style sheet have the color at index zero in the color table (black).
Cell	N="FontScale" V="1"	Unless overridden, characters in text runs of shapes using this style sheet do not have altered font width.
Cell	N="Size" V="0.1666666666666667"	Unless overridden, characters in text runs of shapes using this style sheet have a font size of 0.1666666666666667 inches (12 pt).
Section	N="Paragraph"	Cells under this tag relate to paragraph properties.
Row	IX="0"	These properties apply to the paragraph with index zero.
Cell	N="IndFirst" V="0"	Unless overridden, paragraphs on shapes using this style sheet have no indentation on their first lines.
Cell	N="IndLeft" V="0"	Unless overridden, paragraphs on shapes using this style sheet have no left indentation.
Cell	N="IndRight" V="0"	Unless overridden, paragraphs on shapes using this style sheet have no right indentation.
Cell	N="SpLine" V="-1.2"	Unless overridden, each line in paragraphs on shapes using this style sheet has a height equal to 1.2 times the maximum font size of characters in the line.
Cell	N="HorzAlign" V="1"	Unless overridden, paragraphs on shapes using this style sheet have a centered horizontal alignment.
Cell	N="Bullet" V="0"	Unless overridden, paragraphs on shapes using this style sheet have no bullet.

Element	Attributes	Notes
Cell	N="BulletStr" V=""	Unless overridden, paragraphs on shapes using this style sheet do not have a custom bullet string.
Cell	N="BulletFont" V="0"	Unless overridden, paragraphs on shapes using this style sheet use the default bullet font.
Cell	N="BulletFontSize" V="0"	Unless overridden, bullets for paragraphs on shapes using this style sheet have the same font size as the first character in the paragraph's text.
Cell	N="TextPosAfterBullet" V="0"	Unless overridden, no distance will be added between any bullets for paragraphs on shapes using this style sheet and the text after it.
Cell	N="Flags" V="0"	Unless overridden, paragraphs on shapes using this style sheet have left to right text.
Section	N="Tabs"	Rows and cells under this tag define tabs properties.
Row	IX="0"	Unless overridden, shapes using this style sheet have a single tab stop at the default position.
DocumentSheet	NameU="TheDoc" Name="TheDoc" LineStyle="0" FillStyle="0" TextStyle="0"	The language-dependent and language-independent names of the web drawing are each "TheDoc". Unless overridden, shapes in the web drawing inherit the line property, fill property, and text properties of the style sheet with an ID attribute of zero.
Cell	N="DocLangID" V="en-US"	Unless overridden, text in this web drawing is U.S. English.

3.1.2 Pages XML Part

The following is an example of a Pages XML part.

```
<?xml version="1.0"?>
<Pages xmlns="http://schemas.microsoft.com/office/visio/2011/1/core"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xml:space="preserve">
  <Page ID="0" NameU="Page-1" Name="Page-1">
    <PageSheet LineStyle="0" FillStyle="0" TextStyle="0">
      <Cell N="PageWidth" V="8.5"/>
      <Cell N="PageHeight" V="11"/>
      <Cell N="PageScale" V="1"/>
      <Cell N="DrawingScale" V="1"/>
    </PageSheet>
    <Rel r:id="rId1"/>
  </Page>
</Pages>
```

The following table provides more information about element and attribute values found in the preceding sample **Pages XML**.

Element	Attributes	Notes
Page	ID="0" NameU="Page-1" Name="Page-1"	The ID attribute of the drawing page is zero. The language-dependent and language-independent names of the drawing page are both "Page-1".
PageSheet	LineStyle="0" FillStyle="0" TextStyle="0"	Unless overridden, shapes on this drawing page inherit the line property, fill property, and text properties of the style sheet with an ID attribute of zero.
Cell	N="PageWidth" V="8.5"	The width of the drawing page is 8.5 inches.
Cell	N="PageHeight" V="11"	The height of the drawing page is 11 inches.
Cell	N="PageScale" V="1"	The page scale of the drawing page is one inch.
Cell	N="DrawingScale" V="1"	The drawing scale of the drawing page is one inch.

3.1.3 Page XML Part

The following is an example of a Page XML part.

```
<?xml version="1.0"?>
<PageContents xmlns="http://schemas.microsoft.com/office/visio/2011/1/core"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xml:space="preserve">
  <Shapes>
    <Shape ID="1" Type="Shape" LineStyle="0" FillStyle="0" TextStyle="0">
      <Cell N="PinX" V="4"/>
      <Cell N="PinY" V="5.5"/>
      <Cell N="Width" V="2"/>
      <Cell N="Height" V="1"/>
      <Cell N="LocPinX" V="1"/>
      <Cell N="LocPinY" V="0.5"/>
      <Cell N="Angle" V="0"/>
      <Cell N="FillForegnd" V="#0070c0"/>
      <Section N="Geometry" IX="0">
        <Row T="RelMoveTo" IX="1">
          <Cell N="X" V="0"/>
          <Cell N="Y" V="0"/>
        </Row>
        <Row T="RelLineTo" IX="2">
          <Cell N="X" V="1"/>
          <Cell N="Y" V="0"/>
        </Row>
        <Row T="RelLineTo" IX="3">
          <Cell N="X" V="1"/>
          <Cell N="Y" V="1"/>
        </Row>
        <Row T="RelLineTo" IX="4">
          <Cell N="X" V="0"/>
          <Cell N="Y" V="1"/>
        </Row>
        <Row T="RelLineTo" IX="5">
          <Cell N="X" V="0"/>
          <Cell N="Y" V="0"/>
```

```
          </Row>
        </Section>
        <Text>Sample text</Text>
      </Shape>
    </Shapes>
  </PageContents>
```

The following table provides more information about element and attribute values and element contents found in the preceding sample **Page XML**.

Element	Attributes/contents	Notes
Shape	ID="1" Type="Shape" LineStyle="0" FillStyle="0" TextStyle="0"	The ID attribute of the shape is one. The shape will be displayed and has no subshapes or foreign data. The shape inherits the line property, fill property, and text properties of the style sheet with an ID attribute of zero.
Cell	N="PinX" V="4"	The center of rotation of the shape is 4 inches right from the left edge of the drawing page.
Cell	N="PinY" V="5.5"	The center of rotation of the shape is 5.5 inches up from the bottom edge of the drawing page.
Cell	N="Width" V="2"	The shape is two inches wide.
Cell	N="Height" V="1"	The shape is one inch tall.
Cell	N="LocPinX" V="1"	The center of rotation of the shape is one inch right from the left edge of the shape.
Cell	N="LocPinY" V="0.5"	The center of rotation of the shape is .5 inches up from the bottom edge of the shape.
Cell	N="Angle" V="0"	The shape has no rotation.
Cell	N="FillForegnd" V="#0070c0"	The color of the fill of the shape is equivalent to RGB(0, 112, 192). Note that this overrides the definition of this cell inherited from the style sheet.
Section	N="Geometry" IX="0"	Rows and cells under this element define geometry properties of the shape. This geometry section has an index of zero.
Row	T="RelMoveTo" IX="1"	This row defines the starting point of the geometry path in relative coordinates. This row has an index of one.
Cell	N="X" V="0"	The x-coordinate of the starting point of the geometry path is the same as that of the left edge of the shape.
Cell	N="Y" V="0"	The y-coordinate of the starting point of the geometry path is the same as that of the bottom edge of the shape.
Row	T="RelLineTo"	This row defines a line segment in the geometry path

Element	Attributes/contents	Notes
	IX="2"	from the point defined in the previous row to the point defined in this row's cells in relative coordinates. This row has an index of 2.
Cell	N="X" V="1"	The x-coordinate of the endpoint of the line segment defined by this row is the same as that of the right edge of the shape.
Cell	N="Y" V="0"	The y-coordinate of the endpoint of the line segment defined by this row is the same as that of the bottom edge of the shape.
Row	T="RelLineTo" IX="3"	This row defines a line segment in the geometry path from the point defined in the previous row to the point defined in this row's cells in relative coordinates. This row has an index of 3.
Cell	N="X" V="1"	The x-coordinate of the endpoint of the line segment defined by this row is the same as that of the right edge of the shape.
Cell	N="Y" V="1"	The y-coordinate of the endpoint of the line segment defined by this row is the same as that of the top edge of the shape.
Row	T="RelLineTo" IX="4"	This row defines a line segment in the geometry path from the point defined in the previous row to the point defined in this row's cells in relative coordinates. This row has an index of 4.
Cell	N="X" V="0"	The x-coordinate of the endpoint of the line segment defined by this row is the same as that of the left edge of the shape.
Cell	N="Y" V="1"	The y-coordinate of the endpoint of the line segment defined by this row is the same as that of the top edge of the shape.
Row	T="RelLineTo" IX="5"	This row defines a line segment in the geometry path from the point defined in the previous row to the point defined in this row's cells in relative coordinates. This row has an index of 5.
Cell	N="X" V="0"	The x-coordinate of the endpoint of the line segment defined by this row is the same as that of the left edge of the shape.
Cell	N="Y" V="0"	The y-coordinate of the endpoint of the line segment defined by this row is the same as that of the bottom edge of the shape.
Text	Sample text	The shape has text reading 'Sample text'.

3.2 Document with Master Inheritance

This section describes some of the contents of the Masters XML Part, Master XML Part, and Page XML Part for a web drawing consisting of one master containing a rectangle with blue fill color and the text "Sample Text" on it, and one shape inheriting from that master via master-to-shape inheritance:

Note that the document displayed is identical to that in the Document with a Shape on a Page example. Only the location of the data has changed.

3.2.1 Masters XML Part

The following is an example of a Masters XML part.

```
<?xml version="1.0" encoding="utf-8"?>
<Masters xmlns="http://schemas.microsoft.com/office/visio/2011/1/core"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xml:space="preserve">
<Master ID="2" UniqueID="{0020640D-0002-0000-8E40-00608CF305B2}" BaseID="{8FEBAF3E-100E-46A7-
AFB4-6F1C4D75A7A7}" PatternFlags="0">
    <PageSheet LineStyle="0" FillStyle="0" TextStyle="0">
      <Cell N="PageWidth" V="2"/>
      <Cell N="PageHeight" V="1"/>
      <Cell N="PageScale" V="1"/>
      <Cell N="DrawingScale" V="1"/>
    </PageSheet>
    <Rel r:id="rId1"/>
</Master>
</Masters>
```

The following table provides more information about element and attribute values and element contents found in the preceding sample **Masters XML**.

Element	Attributes	Notes
Master	ID="2" UniqueID="{0020640D-0002-0000-8E40-00608CF305B2}" BaseID="{8FEBAF3E-100E-46A7-AFB4-6F1C4D75A7A7}" PatternFlags="0"	The ID attribute of the master is two. The UniqueID of the master is the GUID {0020640D-0002-0000-8E40-00608CF305B2}. The BaseID of the master is the GUID {8FEBAF3E-100E-46A7-AFB4-6F1C4D75A7A7}. The master is not used as a custom pattern master.
Cell	N="PageWidth" V="2"	The width of the drawing page of the master is two inches.
Cell	N="PageHeight" V="1"	The height of the drawing page of the master is one inch.
Cell	N="PageScale" V="1"	The page scale of the drawing page of the master is one inch.
Cell	N="DrawingScale" V="1"	The drawing scale of the drawing page of the master is one inch.
Rel	r:id="rId1"	The shapes on this master are located in the XML part referenced by rId1 in the master.xml.rels XML part.

3.2.2 Master XML Part

The following is an example of a Master XML part.

```
<?xml version="1.0" encoding="utf-8"?>
<MasterContents xmlns="http://schemas.microsoft.com/office/visio/2011/1/core"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xml:space="preserve">
  <Shapes>
    <Shape ID="5" Type="Shape" LineStyle="3" FillStyle="3" TextStyle="3">
      <Cell N="PinX" V="1"/>
      <Cell N="PinY" V="0.5"/>
      <Cell N="Width" V="2"/>
      <Cell N="Height" V="1"/>
      <Cell N="LocPinX" V="1"/>
      <Cell N="LocPinY" V="0.5"/>
      <Cell N="Angle" V="0"/>
      <Cell N="FillForegnd" V="#0070c0"/>
      <Cell N="FillBkgnd" V="#000000"/>
      <Section N="Geometry" IX="0">
        <Cell N="NoFill" V="0"/>
        <Cell N="NoLine" V="0"/>
        <Cell N="NoShow" V="0"/>
        <Row T="RelMoveTo" IX="1">
          <Cell N="X" V="0"/>
          <Cell N="Y" V="0"/>
        </Row>
        <Row T="RelLineTo" IX="2">
          <Cell N="X" V="1"/>
          <Cell N="Y" V="0"/>
        </Row>
        <Row T="RelLineTo" IX="3">

          <Cell N="X" V="1"/>
          <Cell N="Y" V="1"/>
```

```
          </Row>
          <Row T="RelLineTo" IX="4">
            <Cell N="X" V="0"/>
            <Cell N="Y" V="1"/>
          </Row>
          <Row T="RelLineTo" IX="5">
            <Cell N="X" V="0"/>
            <Cell N="Y" V="0"/>
          </Row>
        </Section>
        <Text>Sample text</Text>
      </Shape>
    </Shapes>
  </MasterContents>
```

The following table provides more information about element and attribute values and element contents found in the preceding sample **Master XML**.

Element	Attributes/contents	Notes
Shape	ID="1" Type="Shape" LineStyle="3" FillStyle="3" TextStyle="3">	The ID attribute of the shape on the master is one. The shape inherits the line property, fill property, and text properties of the style sheet with an ID attribute of three.
Cell	N="PinX" V="1"	The center of rotation of the shape on the master is one inch right from the left edge of the drawing page of the master.
Cell	N="PinY" V="0.5"	The center of rotation of the shape on the master is 0.5 inches up from the bottom edge of the drawing page of the master.
Cell	N="Width" V="2"	The shape on the master is two inches wide.
Cell	N="Height" V="1"	The shape on the master is one inch tall.
Cell	N="LocPinX" V="1"	The center of rotation of the shape on the master is one inch right from the left edge of the shape.
Cell	N="LocPinY" V="0.5"	The center of rotation of the shape on the master is .5 inches up from the bottom edge of the shape.
Cell	N="Angle" V="0"	The shape on the master has no rotation.
Cell	N="FillForegnd" V="#0070c0"	The color of the fill of the shape on the master is equivalent to RGB(0, 112, 192).
Section	N="Geometry" IX="0"	Rows and cells under this element define geometry properties of the shape on the master. This geometry section has an index of zero.
Row	T="RelMoveTo" IX="1"	This row defines the starting point of the geometry path in relative coordinates. This row has an index of one.
Cell	N="X"	The x-coordinate of the starting point of the geometry

Element	Attributes/contents	Notes
	V="0"	path is the same as that of the left edge of the shape.
Cell	N="Y" V="0"	The y-coordinate of the starting point of the geometry path is the same as that of the bottom edge of the shape.
Row	T="RelLineTo" IX="2"	This row defines a line segment in the geometry path from the point defined in the previous row to the point defined in this row's cells in relative coordinates. This row has an index of 2.
Cell	N="X" V="1"	The x-coordinate of the endpoint of the line segment defined by this row is the same as that of the right edge of the shape.
Cell	N="Y" V="0"	The y-coordinate of the endpoint of the line segment defined by this row is the same as that of the bottom edge of the shape.
Row	T="RelLineTo" IX="3"	This row defines a line segment in the geometry path from the point defined in the previous row to the point defined in this row's cells in relative coordinates. This row has an index of 3.
Cell	N="X" V="1"	The x-coordinate of the endpoint of the line segment defined by this row is the same as that of the right edge of the shape.
Cell	N="Y" V="1"	The y-coordinate of the endpoint of the line segment defined by this row is the same as that of the top edge of the shape.
Row	T="RelLineTo" IX="4"	This row defines a line segment in the geometry path from the point defined in the previous row to the point defined in this row's cells in relative coordinates. This row has an index of 4.
Cell	N="X" V="0"	The x-coordinate of the endpoint of the line segment defined by this row is the same as that of the left edge of the shape.
Cell	N="Y" V="1"	The y-coordinate of the endpoint of the line segment defined by this row is the same as that of the top edge of the shape.
Row	T="RelLineTo" IX="5"	This row defines a line segment in the geometry path from the point defined in the previous row to the point defined in this row's cells in relative coordinates. This row has an index of 5.
Cell	N="X" V="0"	The x-coordinate of the endpoint of the line segment defined by this row is the same as that of the left edge of the shape.
Cell	N="Y" V="0"	The y-coordinate of the endpoint of the line segment defined by this row is the same as that of the bottom edge of the shape.
Text	Sample text	The shape on the master has text reading 'Sample text'.

3.2.3 Page XML Part

The following is an example of a Page XML part.

```
<?xml version="1.0" encoding="utf-8"?>
<PageContents xmlns="http://schemas.microsoft.com/office/visio/2011/1/core"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xml:space="preserve">
  <Shapes>
    <Shape ID="1" Type="Shape" Master="2">
      <Cell N="PinX" V="4"/>
      <Cell N="PinY" V="5.5"/>
    </Shape>
</Shapes>
</PageContents>
```

The following table provides more information about element and attribute values found in the preceding sample **Page XML**.

Element	Attributes	Notes
Shape	ID="1" Type="Shape" Master="2"	The ID attribute of the shape is one. The shape will be displayed and has no subshapes or foreign data. The shape inherits the values of the master with an ID attribute of two via master-to-shape inheritance.
Cell	N="PinX" V="4"	The center of rotation of the shape is 4 inches right from the left edge of the drawing page. Note that this overrides the PinX value in the Master XML Part.
Cell	N="PinY" V="5.5"	The center of rotation of the shape is 5.5 inches up from the bottom edge of the drawing page. Note that this overrides the PinY value in the Master XML Part.

4 Security

4.1 Security Considerations for Implementers

None.

4.2 Index of Security Fields

None.

5 Appendix A: Full XML Schema

For ease of implementation, the following is the full XML schema for this protocol.

```
<!--
  XML for Visio Schema
  http://schemas.microsoft.com/office/visio/2011/1/core
  Copyright (C) 2000-2002 Microsoft Corporation. All rights reserved.
-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://schemas.microsoft.com/office/visio/2011/1/core"
            xmlns="http://schemas.microsoft.com/office/visio/2011/1/core"
            xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
            elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xsd:import namespace="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
              schemaLocation="oRel.xsd"/>

  <xsd:annotation>
    <xsd:documentation>
      Permission to copy, display and distribute the contents of this document (the
      "Specification"), in any medium for any purpose without fee or royalty is
      hereby granted, provided that you include the following notice on ALL copies of
      the Specification, or portions thereof, that you make:
      Copyright (c) Microsoft Corporation.  All rights reserved.  Permission to copy,
      display and distribute this document is available at:
      http://msdn.microsoft.com/library/en-
us/odcXMLRef/html/odcXMLRefLegalNotice.asp?frame=true.
      No right to create modifications or derivatives of this Specification is
      granted herein. There is a separate patent license available to parties
      interested in implementing software programs that can read and write files that
      conform to the Specification.  This patent license is available at this
      location:  http://www.microsoft.com/mscorp/ip/format/xmlpatentlicense.asp.
      THE SPECIFICATION IS PROVIDED "AS IS" AND MICROSOFT MAKES NO REPRESENTATIONS OR
      WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF
      MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE;
      THAT THE CONTENTS OF THE SPECIFICATION ARE SUITABLE FOR ANY PURPOSE; NOR THAT
      THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS,
      COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. MICROSOFT WILL NOT BE LIABLE FOR ANY
      DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF
      OR RELATING TO ANY USE OR DISTRIBUTION OF THE SPECIFICATION.
      The name and trademarks of Microsoft may NOT be used in any manner, including
      advertising or publicity pertaining to the Specification or its contents
      without specific, written prior permission. Title to copyright in the
      Specification will at all times remain with Microsoft. No other rights are
      granted by implication, estoppel or otherwise.
    </xsd:documentation>
  </xsd:annotation>

  <!--
    Root-level elements for different parts
  -->
  <!--document.xml-->
  <xsd:element name="VisioDocument" type="VisioDocument_Type" />
  <!--masters/masters.xml-->
  <xsd:element name="Masters" type="Masters_Type" />
  <!--masters/master#.xml-->
  <xsd:element name="MasterContents" type="PageContents_Type" />
  <!--pages/pages.xml-->
  <xsd:element name="Pages" type="Pages_Type" />
  <!--pages/page#.xml-->
  <xsd:element name="PageContents" type="PageContents_Type" />
  <!--data/connections.xml-->
  <xsd:element name="DataConnections" type="DataConnections_Type" />
  <!--data/recordsets.xml-->
  <xsd:element name="DataRecordSets" type="DataRecordSets_Type" />
  <!--comments.xml-->
```

```
<xsd:element name="Comments" type="Comments_Type" />
<!--extensions.xml-->
<xsd:element name="Extensions" type="Extensions_Type" />
<!--
  Complex types
-->
<xsd:complexType name="VisioDocument_Type">
  <xsd:sequence>
    <xsd:element name="DocumentSettings" type="DocumentSettings_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="Colors" type="Colors_Type" minOccurs="0" maxOccurs="1" />
    <xsd:element name="FaceNames" type="FaceNames_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="StyleSheets" type="StyleSheets_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="DocumentSheet" type="DocumentSheet_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="EventList" type="EventList_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="HeaderFooter" type="HeaderFooter_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="PublishSettings" type="PublishSettings_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:any minOccurs="0" maxOccurs="unbounded" namespace="##other"
             processContents="lax" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<!--Sheet abstract base class-->
<xsd:complexType name="Sheet_Type" abstract="true">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Cell" type="Cell_Type" minOccurs="0"
                 maxOccurs="unbounded" />
    <xsd:element name="Trigger" type="Trigger_Type" minOccurs="0"
                 maxOccurs="unbounded" />
    <xsd:element name="Section" type="Section_Type" minOccurs="0"
                 maxOccurs="unbounded" />

    <!--Sheets can hold and roundtrip arbitrary, unknown sub-XML-->
    <xsd:any minOccurs="0" maxOccurs="unbounded" namespace="##other"
             processContents="lax" />
  </xsd:sequence>

  <!--Style sheet IDs for inheritance-->
  <xsd:attribute name="LineStyle" type="xsd:unsignedInt" />
  <xsd:attribute name="FillStyle" type="xsd:unsignedInt" />
  <xsd:attribute name="TextStyle" type="xsd:unsignedInt" />

  <!--Sheets can hold and roundtrip arbitrary, unknown attributes-->
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<!--Section base types-->
<xsd:complexType name="Section_Type">
  <xsd:sequence>
    <!--Cells only show up directly under the Geometry section-->
    <xsd:element name="Cell" type="Cell_Type" minOccurs="0"
                 maxOccurs="unbounded" />
    <xsd:element name="Trigger" type="Trigger_Type" minOccurs="0"
                 maxOccurs="unbounded" />

    <!--All Sections have child rows-->
    <xsd:element name="Row" type="Row_Type" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>

  <xsd:attribute name="N" type="xsd:string" use="required" />
  <xsd:attribute name="Del" type="xsd:boolean" />
```

```
  <!--Only Geometry sections have indexes as there can be several in a Sheet-->
  <xsd:attribute name="IX" type="xsd:unsignedInt" />
</xsd:complexType>

<!--Row base types-->
<xsd:complexType name="Row Type">
  <xsd:sequence>
    <xsd:element name="Cell" type="Cell_Type" minOccurs="0"
                 maxOccurs="unbounded" />
    <xsd:element name="Trigger" type="Trigger_Type" minOccurs="0"
                 maxOccurs="unbounded" />
  </xsd:sequence>

  <!--The name attributes are only used on named rows-->
  <xsd:attribute name="N" type="xsd:string" />
  <xsd:attribute name="LocalName" type="xsd:string" />

  <!--The index attribute is only used on indexed rows-->
  <xsd:attribute name="IX" type="xsd:unsignedInt" />

  <!--The type attribute is only used on Rows in the Geometry section-->
  <xsd:attribute name="T" type="xsd:string" />

  <!--Any row can be locally deleted-->
  <xsd:attribute name="Del" type="xsd:boolean" />
</xsd:complexType>

<!--Cell base type-->
<xsd:complexType name="Cell Type" mixed="true">
  <xsd:sequence>
    <xsd:element name="RefBy" type="RefBy Type" minOccurs="0"
                 maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="N" type="xsd:string" use="required" />
  <xsd:attribute name="U" type="xsd:string" />
  <xsd:attribute name="E" type="xsd:string" />
  <xsd:attribute name="F" type="xsd:string" />
  <xsd:attribute name="V" type="xsd:string" />
</xsd:complexType>

<!--Trigger base type-->
<xsd:complexType name="Trigger Type" mixed="true">
  <xsd:sequence>
    <xsd:element name="RefBy" type="RefBy_Type" minOccurs="0"
                 maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="N" type="xsd:string" use="required" />
</xsd:complexType>

<xsd:complexType name="DocumentSheet_Type">
  <xsd:complexContent>
    <xsd:extension base="Sheet Type">
      <xsd:attribute name="Name" type="xsd:string" />
      <xsd:attribute name="NameU" type="xsd:string" />
      <xsd:attribute name="IsCustomName" type="xsd:boolean" />
      <xsd:attribute name="IsCustomNameU" type="xsd:boolean" />
      <xsd:attribute name="UniqueID" type="xsd:string" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="StyleSheet_Type">
  <xsd:complexContent>
    <xsd:extension base="Sheet_Type">
      <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
      <xsd:attribute name="Name" type="xsd:string" />
      <xsd:attribute name="NameU" type="xsd:string" />
      <xsd:attribute name="IsCustomName" type="xsd:boolean" />
      <xsd:attribute name="IsCustomNameU" type="xsd:boolean" />
```

```
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="PageSheet_Type">
  <xsd:complexContent>
    <xsd:extension base="Sheet_Type">
      <xsd:attribute name="UniqueID" type="xsd:string" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ShapeSheet_Type">
  <xsd:complexContent>
    <xsd:extension base="Sheet_Type">
      <xsd:sequence>
        <!--Shape Text-->
        <xsd:element name="Text" type="Text_Type" minOccurs="0" maxOccurs="1" />

        <!--Legacy data fields-->
        <xsd:element name="Data1" type="Data_Type" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Data2" type="Data_Type" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Data3" type="Data_Type" minOccurs="0" maxOccurs="1" />

        <!--Foreign object data for images, OLE, ActiveX, etc.-->
        <xsd:element name="ForeignData" type="ForeignData_Type" minOccurs="0"
                     maxOccurs="1" />

        <!--Sub-shapes if this is a group-->
        <xsd:element name="Shapes" type="Shapes_Type" minOccurs="0"
                     maxOccurs="1" />
      </xsd:sequence>

      <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
      <xsd:attribute name="OriginalID" type="xsd:unsignedInt" />
      <xsd:attribute name="Del" type="xsd:boolean" />
      <xsd:attribute name="MasterShape" type="xsd:unsignedInt" />
      <xsd:attribute name="UniqueID" type="xsd:string" />
      <xsd:attribute name="Name" type="xsd:string" />
      <xsd:attribute name="NameU" type="xsd:string" />
      <xsd:attribute name="IsCustomName" type="xsd:boolean" />
      <xsd:attribute name="IsCustomNameU" type="xsd:boolean" />
      <xsd:attribute name="Master" type="xsd:unsignedInt" />
      <xsd:attribute name="Type" type="xsd:token" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="Text_Type" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="cp" type="cp_Type" minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="pp" type="pp_Type" minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="tp" type="tp_Type" minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="fld" type="fld_Type" minOccurs="0" maxOccurs="unbounded" />
  </xsd:choice>
</xsd:complexType>

<!--Text fields-->
<xsd:complexType name="cp_Type">
  <xsd:attribute name="IX" type="xsd:unsignedInt" use="required" />
</xsd:complexType>
<xsd:complexType name="pp_Type">
  <xsd:attribute name="IX" type="xsd:unsignedInt" use="required" />
</xsd:complexType>
<xsd:complexType name="tp_Type">
  <xsd:attribute name="IX" type="xsd:unsignedInt" use="required" />
</xsd:complexType>
<xsd:complexType name="fld_Type">
  <xsd:simpleContent>
```

```
    <xsd:extension base="xsd:string">
      <xsd:attribute name="IX" type="xsd:unsignedInt" use="required" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="Data_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string" />
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="ForeignData_Type" mixed="true">
  <xsd:sequence>
    <xsd:element name="Rel" type="Rel_Type" minOccurs="1" maxOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="ForeignType" type="xsd:token" use="required" />
  <xsd:attribute name="ObjectType" type="xsd:unsignedInt" />
  <xsd:attribute name="ShowAsIcon" type="xsd:boolean" />
  <xsd:attribute name="ObjectWidth" type="xsd:double" />
  <xsd:attribute name="ObjectHeight" type="xsd:double" />
  <xsd:attribute name="MappingMode" type="xsd:unsignedShort" />
  <xsd:attribute name="ExtentX" type="xsd:double" />
  <xsd:attribute name="ExtentY" type="xsd:double" />
  <xsd:attribute name="CompressionType" type="xsd:token" />
  <xsd:attribute name="CompressionLevel" type="xsd:double" />
</xsd:complexType>

<xsd:complexType name="DocumentSettings Type">
  <xsd:all>
    <xsd:element name="GlueSettings" type="GlueSettings Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="SnapSettings" type="SnapSettings_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="SnapExtensions" type="SnapExtensions_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="SnapAngles" type="SnapAngles_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="DynamicGridEnabled" type="DynamicGridEnabled_Type"
                 minOccurs="0" maxOccurs="1" />
    <xsd:element name="ProtectStyles" type="ProtectStyles_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="ProtectShapes" type="ProtectShapes_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="ProtectMasters" type="ProtectMasters_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="ProtectBkgnds" type="ProtectBkgnds_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="CustomMenusFile" type="CustomMenusFile_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="CustomToolbarsFile" type="CustomToolbarsFile_Type"
                 minOccurs="0" maxOccurs="1" />
    <xsd:element name="AttachedToolbars" type="AttachedToolbars_Type" minOccurs="0"
                 maxOccurs="1" />
  </xsd:all>
  <xsd:attribute name="TopPage" type="xsd:unsignedInt" />
  <xsd:attribute name="DefaultTextStyle" type="xsd:unsignedInt" />
  <xsd:attribute name="DefaultLineStyle" type="xsd:unsignedInt" />
  <xsd:attribute name="DefaultFillStyle" type="xsd:unsignedInt" />
  <xsd:attribute name="DefaultGuideStyle" type="xsd:unsignedInt" />
</xsd:complexType>
<xsd:complexType name="GlueSettings_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:int" />
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="SnapSettings_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:int" />
```

```
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="SnapExtensions_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:int" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="SnapAngles_Type">
    <xsd:sequence>
      <xsd:element name="SnapAngle" type="SnapAngle_Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="SnapAngle_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:double" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="DynamicGridEnabled Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:boolean" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="ProtectStyles_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:boolean" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="ProtectShapes Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:boolean" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="ProtectMasters Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:boolean" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="ProtectBkgnds_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:boolean" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="CustomMenusFile_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="CustomToolbarsFile_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="AttachedToolbars Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:base64Binary" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="Colors Type">
    <xsd:sequence>
      <xsd:element name="ColorEntry" type="ColorEntry_Type" minOccurs="1"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ColorEntry Type">
    <xsd:attribute name="IX" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="RGB" type="xsd:string" use="required" />
  </xsd:complexType>
  <xsd:complexType name="FaceNames_Type">
```

```
    <xsd:sequence>
      <xsd:element name="FaceName" type="FaceName_Type" minOccurs="1"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="FaceName_Type">
    <xsd:attribute name="NameU" type="xsd:string" use="required" />
    <xsd:attribute name="UnicodeRanges" type="xsd:string" />
    <xsd:attribute name="CharSets" type="xsd:string" />
    <xsd:attribute name="Panos" type="xsd:string" />
    <xsd:attribute name="Panose" type="xsd:string" />
    <xsd:attribute name="Flags" type="xsd:unsignedInt" />
  </xsd:complexType>
  <xsd:complexType name="StyleSheets_Type">
    <xsd:sequence>
      <xsd:element name="StyleSheet" type="StyleSheet_Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Masters_Type">
    <xsd:sequence>
      <xsd:element name="Master" type="Master_Type" minOccurs="0"
                   maxOccurs="unbounded" />
      <xsd:element name="MasterShortcut" type="MasterShortcut_Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Master_Type">
    <xsd:all>
      <xsd:element name="PageSheet" type="PageSheet_Type" minOccurs="0"
                   maxOccurs="1" />
      <xsd:element name="Rel" type="Rel_Type" minOccurs="1" maxOccurs="1" />
      <xsd:element name="Icon" type="Icon_Type" minOccurs="0" maxOccurs="1" />
    </xsd:all>
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="BaseID" type="xsd:string" />
    <xsd:attribute name="UniqueID" type="xsd:string" />
    <xsd:attribute name="MatchByName" type="xsd:boolean" />
    <xsd:attribute name="Name" type="xsd:string" />
    <xsd:attribute name="NameU" type="xsd:string" />
    <xsd:attribute name="IsCustomName" type="xsd:boolean" />
    <xsd:attribute name="IsCustomNameU" type="xsd:boolean" />
    <xsd:attribute name="IconSize" type="xsd:unsignedShort" />
    <xsd:attribute name="PatternFlags" type="xsd:unsignedShort" />
    <xsd:attribute name="Prompt" type="xsd:string" />
    <xsd:attribute name="Hidden" type="xsd:boolean" />
    <xsd:attribute name="IconUpdate" type="xsd:boolean" />
    <xsd:attribute name="AlignName" type="xsd:unsignedShort" />
    <xsd:attribute name="MasterType" type="xsd:unsignedShort" />
  </xsd:complexType>
  <xsd:complexType name="Icon_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:base64Binary" />
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="MasterShortcut_Type">
    <xsd:all>
      <xsd:element name="Icon" type="Icon_Type" minOccurs="0" maxOccurs="1" />
    </xsd:all>
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="Name" type="xsd:string" />
    <xsd:attribute name="NameU" type="xsd:string" />
    <xsd:attribute name="IsCustomName" type="xsd:boolean" />
    <xsd:attribute name="IsCustomNameU" type="xsd:boolean" />
    <xsd:attribute name="IconSize" type="xsd:unsignedShort" />
    <xsd:attribute name="PatternFlags" type="xsd:unsignedShort" />
    <xsd:attribute name="Prompt" type="xsd:string" />
    <xsd:attribute name="ShortcutURL" type="xsd:string" />
    <xsd:attribute name="ShortcutHelp" type="xsd:string" />
```

```
    <xsd:attribute name="AlignName" type="xsd:unsignedShort" />
    <xsd:attribute name="MasterType" type="xsd:unsignedShort" />
  </xsd:complexType>
  <xsd:complexType name="PageContents_Type">
    <xsd:sequence>
      <xsd:element name="Shapes" type="Shapes Type" minOccurs="0" maxOccurs="1" />
      <xsd:element name="Connects" type="Connects_Type" minOccurs="0"
                   maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Shapes Type">
    <xsd:sequence>
      <xsd:element name="Shape" type="ShapeSheet_Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Connects Type">
    <xsd:sequence>
      <xsd:element name="Connect" type="Connect Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Connect Type">
    <xsd:attribute name="FromSheet" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="FromCell" type="xsd:string" />
    <xsd:attribute name="FromPart" type="xsd:int" />
    <xsd:attribute name="ToSheet" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="ToCell" type="xsd:string" />
    <xsd:attribute name="ToPart" type="xsd:int" />
  </xsd:complexType>
  <xsd:complexType name="Pages Type">
    <xsd:sequence>
      <xsd:element name="Page" type="Page_Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Page_Type">
    <xsd:all>
      <xsd:element name="PageSheet" type="PageSheet_Type" minOccurs="0"
                   maxOccurs="1" />
      <xsd:element name="Rel" type="Rel_Type" minOccurs="1" maxOccurs="1" />
    </xsd:all>
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="Name" type="xsd:string" />
    <xsd:attribute name="NameU" type="xsd:string" />
    <xsd:attribute name="IsCustomName" type="xsd:boolean" />
    <xsd:attribute name="IsCustomNameU" type="xsd:boolean" />
    <xsd:attribute name="Background" type="xsd:boolean" />
    <xsd:attribute name="BackPage" type="xsd:unsignedInt" />
    <xsd:attribute name="ViewScale" type="xsd:double" />
    <xsd:attribute name="ViewCenterX" type="xsd:double" />
    <xsd:attribute name="ViewCenterY" type="xsd:double" />
    <xsd:attribute name="ReviewerID" type="xsd:unsignedInt" />
    <xsd:attribute name="AssociatedPage" type="xsd:unsignedInt" />
  </xsd:complexType>
  <xsd:complexType name="EventList_Type">
    <xsd:sequence>
      <xsd:element name="EventItem" type="EventItem Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="EventItem_Type">
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="Action" type="xsd:unsignedShort" use="required" />
    <xsd:attribute name="EventCode" type="xsd:unsignedShort" use="required" />
    <xsd:attribute name="Enabled" type="xsd:boolean" />
    <xsd:attribute name="Target" type="xsd:string" use="required" />
    <xsd:attribute name="TargetArgs" type="xsd:string" use="required" />
  </xsd:complexType>
```

```
<xsd:complexType name="HeaderFooter_Type">
  <xsd:all>
    <xsd:element name="HeaderMargin" type="HeaderMargin_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="FooterMargin" type="FooterMargin_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="HeaderLeft" type="HeaderLeft_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="HeaderCenter" type="HeaderCenter_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="HeaderRight" type="HeaderRight Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="FooterLeft" type="FooterLeft_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="FooterCenter" type="FooterCenter_Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="FooterRight" type="FooterRight Type" minOccurs="0"
                 maxOccurs="1" />
    <xsd:element name="HeaderFooterFont" type="HeaderFooterFont Type" minOccurs="0"
                 maxOccurs="1" />
  </xsd:all>
  <xsd:attribute name="HeaderFooterColor" type="xsd:string" />
</xsd:complexType>
<xsd:complexType name="HeaderMargin_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:double">
      <xsd:attribute name="Unit" type="xsd:string" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="FooterMargin Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:double">
      <xsd:attribute name="Unit" type="xsd:string" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="HeaderLeft_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string" />
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="HeaderCenter_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string" />
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="HeaderRight Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string" />
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="FooterLeft_Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string" />
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="FooterCenter Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string" />
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="FooterRight Type">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string" />
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="HeaderFooterFont_Type">
  <xsd:attribute name="Height" type="xsd:int" />
```

```
    <xsd:attribute name="Width" type="xsd:int" />
    <xsd:attribute name="Escapement" type="xsd:int" />
    <xsd:attribute name="Orientation" type="xsd:int" />
    <xsd:attribute name="Weight" type="xsd:int" />
    <xsd:attribute name="Italic" type="xsd:unsignedByte" />
    <xsd:attribute name="Underline" type="xsd:unsignedByte" />
    <xsd:attribute name="StrikeOut" type="xsd:unsignedByte" />
    <xsd:attribute name="CharSet" type="xsd:unsignedByte" />
    <xsd:attribute name="OutPrecision" type="xsd:unsignedByte" />
    <xsd:attribute name="ClipPrecision" type="xsd:unsignedByte" />
    <xsd:attribute name="Quality" type="xsd:unsignedByte" />
    <xsd:attribute name="PitchAndFamily" type="xsd:unsignedByte" />
    <xsd:attribute name="FaceName" type="xsd:string" />
  </xsd:complexType>
  <xsd:complexType name="DataConnections_Type">
    <xsd:sequence>
      <xsd:element name="DataConnection" type="DataConnection Type" minOccurs="1"
                   maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="NextID" type="xsd:unsignedInt" use="required" />
  </xsd:complexType>
  <xsd:complexType name="DataConnection_Type">
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="FileName" type="xsd:string" use="required" />
    <xsd:attribute name="ConnectionString" type="xsd:string" />
    <xsd:attribute name="Command" type="xsd:string" />
    <xsd:attribute name="FriendlyName" type="xsd:string" />
    <xsd:attribute name="Timeout" type="xsd:unsignedInt" />
    <xsd:attribute name="AlwaysUseConnectionFile" type="xsd:boolean" />
  </xsd:complexType>
  <xsd:complexType name="DataRecordSets Type">
    <xsd:sequence>
      <xsd:element name="DataRecordSet" type="DataRecordSet_Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="NextID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="ActiveRecordsetID" type="xsd:unsignedInt" />
    <xsd:attribute name="DataWindowOrder" type="xsd:string" />
  </xsd:complexType>
  <xsd:complexType name="DataRecordSet Type">
    <xsd:sequence>
      <xsd:element name="Rel" type="Rel Type" minOccurs="1" maxOccurs="1" />
      <xsd:element name="DataColumns" type="DataColumns_Type" minOccurs="1"
                   maxOccurs="1" />
      <xsd:element name="PrimaryKey" type="PrimaryKey_Type" minOccurs="0"
                   maxOccurs="unbounded" />
      <xsd:element name="RowMap" type="RowMap_Type" minOccurs="0"
                   maxOccurs="unbounded" />
      <xsd:element name="RefreshConflict" type="RefreshConflict_Type" minOccurs="0"
                   maxOccurs="unbounded" />
      <xsd:element name="AutoLinkComparison" type="AutoLinkComparison_Type"
                   minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="ConnectionID" type="xsd:unsignedInt" />
    <xsd:attribute name="Command" type="xsd:string" />
    <xsd:attribute name="Options" type="xsd:unsignedInt" />
    <xsd:attribute name="TimeRefreshed" type="xsd:dateTime" />
    <xsd:attribute name="NextRowID" type="xsd:unsignedInt" />
    <xsd:attribute name="Name" type="xsd:string" />
    <xsd:attribute name="RowOrder" type="xsd:boolean" />
    <xsd:attribute name="RefreshOverwriteAll" type="xsd:boolean" />
    <xsd:attribute name="RefreshNoReconciliationUI" type="xsd:boolean" />
    <xsd:attribute name="RefreshInterval" type="xsd:unsignedInt" />
    <xsd:attribute name="ReplaceLinks" type="xsd:unsignedInt" />
    <xsd:attribute name="Checksum" type="xsd:unsignedInt" />
  </xsd:complexType>
  <xsd:complexType name="DataColumns_Type">
    <xsd:sequence>
```

```
      <xsd:element name="DataColumn" type="DataColumn_Type" minOccurs="1"
                   maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="SortColumn" type="xsd:string" />
    <xsd:attribute name="SortAsc" type="xsd:boolean" />
  </xsd:complexType>
  <xsd:complexType name="DataColumn_Type">
    <xsd:attribute name="ColumnNameID" type="xsd:string" use="required" />
    <xsd:attribute name="Name" type="xsd:string" use="required" />
    <xsd:attribute name="Label" type="xsd:string" use="required" />
    <xsd:attribute name="OrigLabel" type="xsd:string" />
    <xsd:attribute name="LangID" type="xsd:unsignedInt" />
    <xsd:attribute name="Calendar" type="xsd:unsignedShort" />
    <xsd:attribute name="DataType" type="xsd:unsignedShort" />
    <xsd:attribute name="UnitType" type="xsd:string" />
    <xsd:attribute name="Currency" type="xsd:unsignedShort" />
    <xsd:attribute name="Degree" type="xsd:unsignedInt" />
    <xsd:attribute name="DisplayWidth" type="xsd:unsignedInt" />
    <xsd:attribute name="DisplayOrder" type="xsd:unsignedInt" />
    <xsd:attribute name="Mapped" type="xsd:boolean" />
    <xsd:attribute name="Hyperlink" type="xsd:boolean" />
  </xsd:complexType>
  <xsd:complexType name="PrimaryKey Type">
    <xsd:sequence>
      <xsd:element name="RowKeyValue" type="RowKeyValue_Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="ColumnNameID" type="xsd:string" use="required" />
  </xsd:complexType>
  <xsd:complexType name="RowKeyValue_Type">
    <xsd:attribute name="RowID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="Value" type="xsd:string" use="required" />
  </xsd:complexType>
  <xsd:complexType name="RowMap Type">
    <xsd:attribute name="RowID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="PageID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="ShapeID" type="xsd:unsignedInt" use="required" />
  </xsd:complexType>
  <xsd:complexType name="RefreshConflict_Type">
    <xsd:attribute name="RowID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="ShapeID" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="PageID" type="xsd:unsignedInt" use="required" />
  </xsd:complexType>
  <xsd:complexType name="AutoLinkComparison_Type">
    <xsd:attribute name="ColumnName" type="xsd:string" use="required" />
    <xsd:attribute name="ContextType" type="xsd:unsignedInt" use="required" />
    <xsd:attribute name="ContextTypeLabel" type="xsd:string" />
  </xsd:complexType>
  <xsd:complexType name="PublishSettings_Type">
    <xsd:sequence>
      <xsd:element name="PublishedPage" type="PublishedPage_Type" minOccurs="0"
                   maxOccurs="unbounded" />
      <xsd:element name="RefreshableData" type="RefreshableData_Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Comments Type">
    <xsd:sequence>
      <xsd:element name="AuthorList" type="AuthorList Type" minOccurs="0"
                   maxOccurs="1" />
      <xsd:element name="CommentList" type="CommentList_Type" minOccurs="0"
                   maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="ShowCommentTags" type="xsd:boolean" />
  </xsd:complexType>
  <xsd:complexType name="AuthorList_Type">
    <xsd:sequence>
      <xsd:element name="AuthorEntry" type="AuthorEntry_Type" minOccurs="0"
                   maxOccurs="unbounded" />
```

```
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="AuthorEntry_Type">
    <xsd:attribute name="Name" type="xsd:string" />
    <xsd:attribute name="Initials" type="xsd:string" />
    <xsd:attribute name="ResolutionID" type="xsd:string" />
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
  </xsd:complexType>
  <xsd:complexType name="CommentList_Type">
    <xsd:sequence>
      <xsd:element name="CommentEntry" type="CommentEntry Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CommentEntry_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="AuthorID" type="xsd:unsignedInt" use="required" />
        <xsd:attribute name="PageID" type="xsd:unsignedInt" use="required" />
        <xsd:attribute name="ShapeID" type="xsd:unsignedInt" />
        <xsd:attribute name="Date" type="xsd:dateTime" use="required" />
        <xsd:attribute name="EditDate" type="xsd:dateTime" />
        <xsd:attribute name="Done" type="xsd:boolean" />
        <xsd:attribute name="CommentID" type="xsd:unsignedInt" use="required" />
        <xsd:attribute name="AutoCommentType" type="xsd:unsignedInt" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="PublishedPage Type">
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
  </xsd:complexType>
  <xsd:complexType name="RefreshableData_Type">
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
  </xsd:complexType>
  <xsd:complexType name="RefBy_Type">
    <xsd:attribute name="T" type="xsd:string" use="required" />
    <xsd:attribute name="ID" type="xsd:unsignedInt" use="required" />
  </xsd:complexType>
  <xsd:complexType name="Extensions_Type">
    <xsd:sequence>
      <xsd:element name="CellDef" type="CellDef_Type" minOccurs="0"
                   maxOccurs="unbounded" />
      <xsd:element name="FunctionDef" type="FunctionDef_Type" minOccurs="0"
                   maxOccurs="unbounded" />
      <xsd:element name="SectionDef" type="SectionDef_Type" minOccurs="0"
                   maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CellDef_Type">
    <xsd:attribute name="N" type="xsd:string" use="required" />
    <xsd:attribute name="T" type="xsd:token" use="required" />
    <xsd:attribute name="F" type="xsd:string" />
    <xsd:attribute name="IX" type="xsd:unsignedByte" />
    <xsd:attribute name="S" type="xsd:unsignedByte" />
  </xsd:complexType>
  <xsd:complexType name="FunctionDef_Type">
    <xsd:attribute name="N" type="xsd:string" use="required" />
  </xsd:complexType>
  <xsd:complexType name="SectionDef Type">
    <xsd:sequence>
      <xsd:element name="CellDef" type="CellDef_Type" minOccurs="0"
                   maxOccurs="unbounded" />
      <xsd:element name="RowDef" type="RowDef Type" minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="N" type="xsd:string" use="required" />
    <xsd:attribute name="T" type="xsd:string" />
    <xsd:attribute name="S" type="xsd:unsignedByte" />
  </xsd:complexType>
  <xsd:complexType name="RowDef_Type">
```

```
      <xsd:sequence>
        <xsd:element name="CellDef" type="CellDef_Type" minOccurs="0"
                     maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Rel Type">
      <xsd:attribute ref="r:id" use="required"/>
    </xsd:complexType>
  </xsd:schema>
```

6 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

- Microsoft SharePoint Server 2013
- Microsoft Visio 2013
- Microsoft Visio 2016

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
6 Appendix B: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

8 Index

E

F

G

H

I

K

L

R

S

T

U

V

W

X

Y