ShapeSheets

ShapeSheet sections

ShapeSheet formulas

Alphabetical list of ShapeSheet cells

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
Click a letter to go to that section.

A

B

C

Understanding ShapeSheets

Related Topics

You can display information about a shape in the ShapeSheet window, which displays a spreadsheet representation of a shape. For example, the ShapeSheet window displays a shape's dimensions, its angle and center of rotation, and the attributes that determine the shape's appearance. The ShapeSheet window also displays the formulas that define the way a shape behaves when it is moved or sized.

How shapes look in the ShapeSheet window

When you draw a shape, Visio records the shape as a collection of horizontal and vertical locations. These locations (called vertices) are displayed in the ShapeSheet window and are measured from a point of origin on the page and connected with line segments.

Visio records each shape vertex as a set of x- and y-coordinates. When you move the shape or change its size, Visio automatically records changes in the ShapeSheet window.

Every page, shape, group, master, object from another program, guide, and guide point has a ShapeSheet representation that you can view. A group, and each shape or object in the group, has its own ShapeSheet representation.

The ShapeSheet window

When the ShapeSheet window is active, the menu bar changes to display menus and commands you use to work in this view. When a ShapeSheet cell is selected, a formula bar appears, which you use to enter and edit ShapeSheet formulas.

Sections, cells, and formulas

TheShapeSheet window is divided into sections that control a particular aspect of a shape's behavior or appearance—for example, its geometry or its formatting. Each section consists of cells that contain formulas. These can be default formulas created by Visio or formulas you create. A formula can be simple and always evaluate to the same value for the cell, or more complex, built from functions, operators, and other elements that evaluate to different values depending on the circumstances.

Sections overview

Related Topics

The ShapeSheet window is divided into sections that control a particular aspect of a shape's behavior or appearance—for example, its geometry or its formatting.

The sections that appear in the ShapeSheet window depend on the shape itself and the sections you choose to display. Most shapes have a Shape Transform section, which contains general positioning information for the shape. (Guides and guide points have a Guide Info section.) Visio adds other sections required for the shape as it is drawn.

A particular shape may not have every possible section. For example, a 2-D shape doesn't have a 1-D Endpoints section, and only an object from another program has an Image Info section.

Cells and formulas

Each section consists of cells that contain formulas, which can be default formulas created by Visio or formulas you create. You change a shape's behavior by editing formulas in cells. Editing ShapeSheet formulas provides a way to change a shape with greater precision than is possible in the drawing window or to define interdependencies among shape attributes.

Definable sections

Not all ShapeSheet sections correspond to shape attributes. You can use the User-Defined Cells and the Scratch section for formulas that contain intermediate calculations or for other purposes.

Formulas overview

Related Topics

Visio formulas are similar to typical spreadsheet formulas in many ways. Like most spreadsheet programs, Visio regards anything in a cell as a formula, even if it's a numeric value or simple cell reference.

A formula can include standard mathematical and comparison operators as well as a full set of mathematical, trigonometric, statistical, and date and time functions and many unique functions. Formulas can contain references to other cells in the shape and to cells in other shapes.

A formula in a cell can be the default formula, one that is inherited from a master, or one you create by entering it in a ShapeSheet cell. Editing a cell's formula changes the way the value of the cell is calculated, so it changes a particular behavior of the shape.

A formula always starts with an equal sign, which Visio inserts automatically.

Default formulas

When you create a shape, Visio automatically creates default formulas. To see what the default formulas are, draw a simple shape (such as a rectangle, ellipse, or straight line) and open its ShapeSheet window.

Automatic updates

Visio automatically updates certain cells whenever you change a shape in a drawing. This means that under some circumstances formulas you enter can be replaced with formulas from Visio. For example, if you drag a corner handle to resize the shape, Visio resets formulas in the PinX, PinY, Width, and Height cells.

If necessary, you can protect formulas against changes using the GUARD function.

Elements of a formula

A formula can contain numbers, coordinates, units of measure, cell references, functions, operators and parentheses, inherited formulas, and local formulas.

Units of measure

Related Topics

When you insert fields into text or build ShapeSheet formulas, you'll often specify units of measure for the values you type.

Visio evaluates the result of a formula differently depending on the cell you enter it in. In general, cells that represent shape position, a dimension, or an angle require a number-unit pair that consists of a number and the qualifying units needed to interpret the number. Many other cells are unitless and evaluate to a string, to TRUE or FALSE, or to an index. For example, the same formula in the FillForegnd cell means color 5 from the drawing's color palette, and in the LockWidth cell it means TRUE (and locks the shape's width).

Always specify a unit of measure when you enter a formula in a cell that expects a dimensional value. If you don't specify a unit of measure, Visio uses the default unit for that cell, which can be page units, drawing units, or angular units.

You can type units of measure in full or use these abbreviations:

Unit	**Valid abbreviation**
centimeters	centimeters, cm., or cm
ciceros	ciceros, cicero, ci., ci, or c
degrees	degrees, degree, deg., deg, or °
didots	didots, didot, di., di, or d
feet	feet, foot, ft., ft, f, or '
inches	inches, inch, in., in, i, or "
kilometers	kilometers, km., or km
meters	meters, meter, metres, metre, m., or m
miles	miles, mile, mi., or mi
millimeters	millimeters, mm., or mm
minutes	minutes, minute, min., min, or '
nautical miles	nm.,n.m.,nm
percent	%
picas	picas, pica, or p
points	points, point, pt., or pt
radians	radians, radian, rad., or rad
seconds	seconds, second, sec., sec, or "
yards	yards, yard, yds., yds, yd., or yd

Implicit units of measure

Related Topics

When Visio parses and stores a number-unit pair, it can use explicit units or implicit units. A number expressed in explicit units is always displayed in the units of measure originally entered. A number expressed in implicit units always converts to the equivalent value in the drawing, page, or angular units appropriate for the cell.

For example, suppose you enter the equivalent of 1 inch in cell A using explicit units and in cell B using implicit units, and that both cell A and cell B use drawing units. Next, you change the default units for the page to centimeters. Cell A still displays 1 in., because it uses explicit units that don't change with the defaults. Cell B now displays 2.54 cm, the equivalent value in the default units.

To enter units implicitly, use this syntax:

```
number [unit, flag]
```

where:

number	Equals the original value, such as 3.7, 1.7E-4, or 5 1/2
unit	Equals the units in which **number** is originally expressed
flag	Equals one of the following letters (either uppercase or lowercase) indicating the measurement system that should be used when the implicit-value unit is displayed:

d	Drawing measurements
p	Page measurements
ed	Duration measurements
t	Type measurements
a	Angular measurements

Valid flags:

Flag	Measurements	Example
d, D	Drawing	=5[in,D]
p, P	Page	=5[in,P]
ed, ED	Duration	=5[h,ED]
t, T	Type	=5[pt,T]
a, A	Angular	=5[deg,A]

Compound units of measure

Related Topics

In formulas, you can express units of measure for compound numbers using the abbreviations in the table below. Visio simplifies the results and displays them in the compound units.

For example, if you enter 45.635 dms, Visio displays the equivalent value as 45° 38' 6".

To specify units	You can type an abbreviation
Ciceros and didots	cicero/didot
Degrees, minutes, and seconds	dms, degrees, deg, or °
Feet and inches	feet/inch
Picas and points	picapoints

Universal strings

Related Topics

In localized versions of Visio, the set of recognized strings changes with the language. If you want your program to work with multiple languages, use the universal strings for units of measure.

For	**Use**
Centimeters	CM
Ciceros	C
Degrees	DEG or °
Didots	D
Feet	FT
Inches	IN
Kilometers	KM
Meters	M
Miles	MI
Millimeters	MM
Minutes	`
Percent	%
Picas	P
Points	PT
Seconds	"
Yards	YD

Operators

You can use operators in expressions in formulas to perform arithmetic operations or logical comparisons. You can also alter the order of evaluation in a formula by enclosing expressions in parentheses. Use the ampersand operator to combine character strings.

Ampersand operator

Related Topics

The ampersand operator returns a new character string. You can create compound words and phrases.

Syntax "string1" & "string2"

Example

```
"dog" & "house"
```

Returns "doghouse".

Arithmetic operators

Related Topics

Arithmetic operators perform operations on numbers. The + and - operators can be used alone as unary operators to establish the sign of a number. The % operator is also a unary operator that identifies the number as a percentage.

Operator	**Action**	**Example**	**Result**
+	Unary plus	+37	+37
-	Unary minus	-37	-37
%	Unary percentage	37%	37%
^	Exponentiation	5^2	25
*	Multiplication	5*2	10
/	Division	5/2	2.5
+	Addition	5+2	7
-	Subtraction	5-2	3

Comparison operators

Related Topics

Comparison operators are used to construct logical expressions. A logical expression evaluates to either TRUE or FALSE.

Operator	**Alternative**	**Action**	**Example**	**Result**
>	_GT_	Greater than	5 > 2	TRUE
<	_LT_	Less than	5 < 2	FALSE
>=	_GE_	Greater than or equal to	5 >= 2	TRUE
<=	_LE_	Less than or equal to	5 <= 2	FALSE
=	_EQ_	Equal to	5 = 2	FALSE
<>	_NE_	Not equal to	5 <> 2	TRUE

The symbolic comparison operators (>, <, and so forth) are the best choice for most comparisons. The alternative operators (_GT_, _LT_, and so forth) perform an exact comparison to the full 15 digits of precision that Visio uses to store values internally.

When you compare rounded or calculated values by using the alternative operators, Visio might return FALSE when for all practical purposes the expression should evaluate to TRUE.

When you compare text strings, Visio converts the string to a value. Text strings that cannot be converted return a value of 0; therefore ,comparisons vary and may not produce the results you expect.

Order of evaluation

Related Topics

When a formula contains more than one expression, the expressions are evaluated in order according to the operation being performed. In Visio, the order of evaluation of operators is:

Order	**Action**	**Operator**
First	Positive	+ (unary)
	Negative	- (unary)
	Percent	%
Second	Exponentiation	^
Third	Multiplication	*
	Division	/
Fourth	Addition	+
	Subtraction	-
Fifth	String concatenation	&
Sixth	Greater than	>
	Greater than or equal to	=>
	Less than	<
	Less than or equal to	<=
Seventh	Equal	=
	Not equal	<>

You can change the order of evaluation by enclosing expressions in parentheses. Visio evaluates expressions within parentheses first, from left to right. For example:

4+5*6 =	4+30 = 34
	(4+5)*6 = 9*6 = 54

If expressions in parentheses are nested, Visio evaluates the expression in the innermost set of parentheses first.

Cell reference syntax

In a cell's formula, you can refer to a ShapeSheet cell of either the current shape or another shape on the same page so that Visio calculates a value for one cell based on another cell's value.

Cell references can include shape IDs or names. You can always refer to any shape on the page by its ID, whether the shape is named or not. If a shape hasn't been named, its default name is Sheet.n, where n is the shape ID. The ID is assigned by Visio when the shape is created and does not change unless you move the shape to another page or document.

If more than one shape on a page has the same name, you must include the ID assigned by Visio. To display the ID or enter a name, use the Special command.

Whether you can refer to a shape by name, and the exact syntax you'll use, depends on where the shape is in relation to the shape whose formula contains the reference. These general rules apply:

- If a shape is a peer of the shape whose formula you are editing, you can refer to that shape by name. If the peer shape is a group, you can refer by name to the group, but not its members. Neither can you refer by name to a shape's parent or its parent's peers.
- You can use Sheet.ID syntax to refer to any shape on the page, whether the shape is in a group or is a parent of the shape.

ShapeSheet	Cell reference	Example
Same shape	Cellname	Width
Named shape, group, or guide	Shapename!Cellname	Star!Angle
Named shape, group, or guide in which more than one shape at the same level has the same name	Shapename.ID!Cellname	Executive.2!Height
Any shape on the page	Sheet.ID!Cellname	Sheet.8!FillForegnd
A cell in an unnamed column with indexed rows	Section.ColumnIndex	Scratch.A5
A cell in a named column with indexed rows	Section.Column[index]	Char.Font[3]
A ShapeSheet cell of the page	ThePage!Cellreference	ThePage!User.Vanishing_Point

Sections, columns, and rows

Visio identifies rows with numbers and columns with letters. Unlike a conventional spreadsheet, however, the first two columns are X and Y. Visio identifies columns with A starting with the third column through W. Visio skips X and Y. Columns after Z are identified as in a conventional spreadsheet: AA, AB, through AZ, and so forth, but these cells can be accessed only programmatically, not through the ShapeSheet window.

Local and inherited formulas

Related Topics

The formula you see in a cell can be one that is inherited from a master or a style, or a local formula that you enter in the cell. Visio uses inherited formulas whenever possible.

Rather than make a local copy of every ShapeSheet formula in the instance, an instance inherits formulas from its master on the local stencil and inherits formatting from the style definition stored with the drawing. This behavior results in smaller files, but also allows changes to the master's formulas or the style definition to be propagated to all instances.

Cells

Black text in a cell indicates an inherited formula. Blue text indicates a local formula, either the result of editing the formula in the ShapeSheet window or some change to the shape that caused Visio to reset the formula for that cell.

Local formulas

When you write a local formula for an instance, you are replacing the inherited formula in the cell, which then no longer inherits its value from the master. An exception is when you apply a style, which always writes new values into the ShapeSheet cells and so can overwrite even locally edited cells.

Error values

Related Topics

Visio displays error values in cells that have incorrect formulas for that cell.

If a formula references a cell that contains an error value, that formula also displays an error value. You can use the functions ISERR, ISERRNA, ISERROR, or ISERRVALUE to look for error values.

Error values

If the cell displays	The formula includes	Example
#DIV/0!	Division by 0	10/0
#VALUE!	An argument or operand of the wrong type	5 + "House"
#REF!	A reference to a cell that does not exist	A cell that refers to another cell that no longer exists
#NUM!	An invalid number	Square root of a negative number
#N/A!	Not available value	NA( ) function

Formatting strings in formulas

Related Topics

To format string output, such as in a prompt cell, a custom property value, or a text field, you specify a format picture. Output can be formatted as a number-unit pair, string, date-time, duration, or currency.

For example, the format picture "0 #/10 uu" formats the number-unit pair 10.9cm as "10 9/10 centimeters".

You can use format pictures in the Format cell of the Custom Properties section and as an argument to the FORMAT or FORMATEX functions. When you insert a text field, format pictures appear in the list of formats in the Field dialog box.

Using functions to format strings

In any formula that resolves to a string, including custom text field formulas, you can use the FORMAT or FORMATEX functions. The FORMAT function returns a string of the formatted output. The FORMATEX function converts untyped input to the units you choose for the formatted result.

Displaying formatted custom properties

You can format the displayed value of a custom property by entering a format picture in the Format cell.

For example, a project timeline shape can have a custom property that measures the cost of a process. By default, a custom property value is a string. To format the string "1200", you can enter "$###,###.00" in the Format cell so that the user sees a currency.

Visio uses the current Regional Settings in the Windows Control Panel to determine the currency symbol and thousands separator to display.

To convert a string to a currency value so that you can perform calculations with the value, use the CY function.

Expressing dates, time, and duration

Related Topics

You can perform operations in ShapeSheet formulas using date, time, and duration values.

Date and time expressions

In Visio, a date and time expression can be evaluated as a single value. A date and time expression is any expression commonly recognized as a date and/or time or a reference to a cell containing a date and/or time. This includes strings and numbers that look like a date and time, and date and time values returned from functions.

For example, the following are valid dates:

"2/28"	"2/29/97"	"2/29/1997"
"2-28"	"2-29-97"	"2-29/1997"
"6 Mar 97"	"1 Jan"	"1 Jan 97"
"1 January 97"	"Jan 1, 97"	"Jan 1, 1997"
"Jan 97"	"January, 1997"	"Jan 1, 00"

The following are valid times:

"3:45"	"3:45:27"	"7a"
"7 am"	"7 p"	"7:30 PM"

Date and time values in Visio are stored internally as a 64 bit floating point number. The value to the left of the decimal represents the number of days since December 31, 1899. The value to the right of the decimal represents the fraction of a day since midnight. Midday is represented by .5.

Functions that return a date and time value

To use dates and times within an expression (rather than as a single constant), you must use the appropriate ShapeSheet function to identify them as date and time values. For example, if you intend to perform calculations with custom properties, text field values, or other ShapeSheet cells that contain a date and time, you can use one of the following functions:

DATE	Converts numbers to a date value.
DATETIME	Converts a string to a date and time value.
DATEVALUE	Converts a string to a date value.
NOW	Returns the current system date as a date and time value.
TIME	Converts numbers to a time value.
TIMEVALUE	Converts a string to a time value.

Calculating duration

You can perform operations that calculate duration or elapsed time. Duration is internally stored as days and the fraction of a day. For example, 1 elapsed week, 7 elapsed days, and 168 elapsed hours all are stored internally as 7.0, but displayed with the appropriate units.

Visio recognizes the following units of duration:

Unit	Abbreviation	Universal abbreviation
elapsed day	eday, ed.	ed
elapsed hour	ehour, eh.	eh

elapsed minute	eminute, em.	em
elapsed second	esecond, es.	es
elapsed week	eweek, ew.	ew

You can add a date and time to a duration to calculate a new date and time. You can perform the following operations using dates, times, and durations:

Input	**Result**
Date-time +/- duration	Date and time value
Duration +/- date-time	Date and time value
Duration +/- duration	Duration value
Date-time + date-time	Date and time value
Date-time - date-time	Duration value

Other date and time functions

To return a portion of a date or time as an integer, you can use the following functions:

DAY	Returns the day component in a date and time expression
DAYOFYEAR	Returns the sequential day of the year in a date and time expression
HOUR	Returns the hours component in a date and time expression
MINUTE	Returns the minutes component in a date and time expression
MONTH	Returns the month component in a date and time expression
SECOND	Returns the seconds component in a date and time expression
WEEKDAY	Returns the number of the weekday in a date and time expression
YEAR	Returns the year component in a date and time expression

Shape Transform section

The Shape Transform section contains general positioning information about a shape: for example, its width, height, angle, and center of rotation (pin); whether the shape has been flipped; and how the shape should behave when resized within a group.

The Shape Transform section includes these cells:

Angle
FlipX
FlipY
Height
LocPinX

LocPinY
PinX
PinY
ResizeMode
Width

Width (Shape Transform section)

Contains the width of the selected shape in drawing units. The default formula for determining the width of a 1-D shape is:

=SQRT((EndX-BeginX)^2+(EndY-BeginY)^2)

Height (Shape Transform section)

The height of the shape in drawing units.

Angle (Shape Transform section)

The shape's current angle of rotation in relation to its parent. The default formula for determining the rotation angle of a 1-D shape is:

=ATAN2(EndY-BeginY,EndX-BeginX)

FlipX (Shape Transform section)

Indicates whether the shape has been flipped horizontally.

non-0	The shape has been flipped.
0	The shape has not been flipped.

FlipY (Shape Transform section)

Indicates whether the shape has been flipped vertically.

non-0	The shape has been flipped.
0	The shape has not been flipped.

PinX (Shape Transform section)

The x-coordinate of the shape's pin (center of rotation) in relation to the origin of its parent.

PinY (Shape Transform section)

The y-coordinate of the shape's pin (center of rotation) in relation to the origin of its parent.

LocPinX (Shape Transform section)

The x-coordinate of the shape's pin (center of rotation) in relation to the origin of the shape. The default formula for determining LocPinX is:

```
=Width * 0.5
```

LocPinY (Shape Transform section)

The y-coordinate of the shape's pin (center of rotation) in relation to the origin of the shape. The default formula for determining LocPinY is:

```
=Height * 0.5
```

ResizeMode (Shape Transform section)

Shows the current resize behavior setting for the shape. For details, see the Behavior command.

0	Use Group's Setting
1	Reposition Only
2	Scale With Group

Geometry section

The Geometry section lists the coordinates of the vertices for the lines and arcs that make up the shape. If the shape has more than one path, there is a Geometry section for each path. To add this section, use the Section command.

The Geometry section can include these cells:

ArcTo
EllipticalArcTo
LineTo
NoFill
NoShow
SplineKnot
SplineStart
Start

Start (Geometry section)

The x- and y-coordinates of the first vertex for the shape. Usually this is the first vertex placed when the shape was drawn, and it does not necessarily correspond to the begin point of a 1-D shape.

LineTo (Geometry section)

The x-and y-coordinates of the ending vertex of a straight line segment.

EllipticalArcTo (Geometry section)

The x- and y-coordinates, control points, angle of eccentricity, and ratio of major and minor axes of an elliptical arc.

X The x-coordinate of the ending vertex on the arc.

Y The y-coordinate of the ending vertex on the arc.

A The x-coordinate of the arc's control point−the point through which the curve of the arc passes. The control point is best located between the beginning and ending vertices of the arc. Otherwise, the arc may grow to an extreme size in order to pass through the control point, with unpredictable results.

B The y-coordinate of the arc's control point.

C The angle of the arc's major axis relative to the drawing page. If the arc has not been rotated, this value is 0.

D The ratio of the arc's major axis to its minor axis. Ordinarily this value is greater than 1, representing a major axis that is longer than the minor axis. If this value is less than 1, the major axis is shorter than the minor axis. Setting this cell to a value less than or equal to 0 or greater than 1000 can lead to unpredictable results.

ArcTo (Geometry section)

The x- and y-coordinates and bow of a circular arc. When you first draw an arc in Visio, it is an elliptical arc, even if it is based on a circle. In the Geometry section, the arc is indicated by an EllipticalArcTo row. The eccentricity handles of an elliptical arc are based on its height and width, so the arc changes shape when you drag an eccentricity handle.

To convert an elliptical arc to a circular arc, change the EllipticalArcTo to an ArcTo row using the Change Row Type command. When you drag the beginning or ending vertex of a circular arc, the arc changes size, but always remains the same portion of a circle.

X The x-coordinate of the ending vertex of the arc.

Y The y-coordinate of the ending vertex of the arc.

A The distance from the arc's bisector to the bow point, which is the point on the arc halfway between the arc's beginning vertex and ending vertex.

NoFill (Geometry section)

The third cell in the Start row of a Geometry section. The NoFill cell indicates whether a shape can be filled. If TRUE, the shape's paths are open, so it cannot be filled. If FALSE, its paths are closed, so it can be filled.

The cell that appears to be GeometryN.A1 is actually GeometryN.NoFill (formerly GeometryN.X0). For compatibility with earlier versions of Visio, you can refer to this cell by either name (GeometryN.NoFill or GeometryN.X0).

NoShow (Geometry section)

The fourth cell in the Start row of a Geometry section. The NoShow cell indicates whether a shape is displayed on the drawing page. If TRUE, the shape is hidden. If FALSE, the shape is shown.

The cell that appears to be GeometryN.B1 is actually GeometryN.NoShow (formerly GeometryN.A0). For compatibility with earlier versions of Visio, you can refer to this cell by either name (GeometryN.NoShow or GeometryN.A0).

SplineStart (Geometry section)

Related Topics

The x- and y-coordinates of a spline's second control point.

Visio displays the definition of a spline in a Geometry section that contains a SplineStart row followed by one or more SplineKnot rows. The SplineStart row must be preceded by another kind of row, such as a Start row, to indicate the first control point of the spline. The preceding row can be a LineTo, ArcTo, or EllipticalArcTo row if the spline follows a segment of that type.

X The *x*-coordinate of the spline's second control point

Y The *y*-coordinate of the spline's second control point

A The position of the second knot on the spline

B The position of the first knot on the spline

C The position of the last knot on the spline

D The degree of the spline (an integer from 1 to 9)

SplineKnot (Geometry section)

Related Topics

The x- and y-coordinates of the spline knots.

X The x-coordinate of a control point

Y The y-coordinate of a control point

A The location of the third or greater spline knot

Protection section

Related Topics

The Protection section shows the current lock settings set with the Protection command plus a few additional locks that can be set only in the ShapeSheet window. Locking helps prevent inadvertent changes to the shape but does not prevent Visio from resetting values in other circumstances. It also does not protect against changes made in the ShapeSheet window.

To protect a formula from being changed by Visio under any circumstances, use the The Protection section includes these cells:

LockAspect
LockBegin
LockCalcWH
LockCrop
LockDelete
LockEnd
LockFormat
LockGroup
LockHeight
LockMoveX
LockMoveY
LockRotate
LockSelect
LockTextEdit
LockVtxEdit
LockWidth

LockWidth (Protection section)

Locks the width of the shape so that its width remains unchanged when the shape is sized.

non-0	Width is locked.
0	Width is not locked.

LockHeight (Protection section)

Locks the height of the shape so that its height remains unchanged when the shape is sized.

non-0	Height is locked.
0	Height is not locked.

LockMoveX (Protection section)

Locks the horizontal position of the shape so it cannot be moved horizontally.

non-0	Horizontal position is locked.
0	Horizontal position is not locked.

LockMoveY (Protection section)

Locks the vertical position of the shape so it cannot be moved vertically.

non-0	Vertical position is locked.
0	Vertical position is not locked.

LockAspect (Protection section)

Locks the aspect ratio of the shape so the shape can be sized only proportionally; it cannot be sized in a single dimension.

non-0	Aspect ratio is locked.
0	Aspect ratio is not locked.

LockDelete (Protection section)

Locks the shape against being deleted.

non-0	Shape cannot be deleted.
0	Shape can be deleted.

LockBegin (Protection section)

Locks the beginning point (BeginX and BeginY) of a 1-D shape to a specific location.

non-0	Beginning point is locked.
0	Beginning is not locked.

LockEnd (Protection section)

Locks the ending point (EndX and EndY) of a 1-D shape to a specific location.

non-0	Ending point is locked.
0	Ending point is not locked.

LockRotate (Protection section)

Locks the shape against being rotated with the rotation tool or the Rotate Left or Rotate Right command.

non-0	Shape cannot be rotated.
0	Shape can be rotated.

LockRotate does not prevent a 1-D shape from being rotated by dragging an endpoint. To lock a 1-D shape against rotation, set LockWidth to a nonzero value.

LockCrop (Protection section)

Locks an object from another program against being cropped with the crop tool.

non-0	Shape cannot be cropped.
0	Shape can be cropped.

LockVtxEdit (Protection section)

Locks the vertices of a shape so they cannot be edited with any tools on the toolbar.

non-0	Vertices cannot be edited.
0	Vertices can be edited.

LockTextEdit (Protection section)

Locks the text of a shape so it cannot be edited. However, the text may still be formatted by applying a style using the Font style list.

non-0	Text cannot be edited.
0	Text can be edited.

LockFormat (Protection section)

Locks the formatting of a shape so it cannot be changed.

non-0	Formatting cannot be changed.
0	Formatting can be changed.

LockGroup (Protection section)

Locks a group against editing.

non-0	Group cannot be edited.
0	Group can be edited.

LockCalcWH (Protection section)

Locks a shape's selection rectangle so it cannot be recalculated when a vertex is edited or a row type is changed in the Geometry section.

non-0	Width and height cannot be recalculated.
0	Width and height can be recalculated.

LockSelect (Protection section)

Prevents a shape from being selected. In order for LockSelect to take effect, the Shapes option must be checked in the Protect Document dialog box.

non-0	Shape cannot be selected.
0	Shape can be selected..

Line Format section

The Line Format section shows the current line formatting attributes for the shape, including pattern, weight, and color; whether the line ends are formatted (for example, with an arrowhead); the size of line end formats; the radius of the rounding circle applied to the line; and line cap style (round or square). Line formats can be set with the Line command, by applying a line style, or by entering a formula in a Line Format cell.

The Line Format section includes these cells:

ArrowSize
BeginArrow
EndArrow
LineCap
LineColor
LinePattern
LineWeight
Rounding

LineCap (Line Format section)

Indicates whether a line has rounded or square line caps. To set the line caps, type one of these numbers or use the Line command.

0	Rounded ends
1	Square ends

LinePattern (Line Format section)

The line pattern of the shape. To set the line pattern, type one of these numbers or use the Line command.

0	No line pattern
1	Solid
2-24	Assorted line patterns that correspond to indexed entries in the Line dialog box.

TIP To specify a custom line pattern, use the USE function in this cell.

LineWeight (Line Format section)

The line weight of the shape. To set the line weight, enter a number with valid units of measure or use the Line command. If the unit of measure is not entered, then the unit of measure for text specified in the Options dialog box is used. The line weight is independent of the scale of the drawing. If the drawing is scaled, the line weight remains the same.

LineColor (Line Format section)

The line color of the shape, identified by the number assigned to that color in the Color Palette dialog box. To set the line color, enter a number from 0 to 23 or use the Line command. To switch to a different color palette, use the Color Palette command.

A value of 24 or higher indicates a custom color that Visio specifies as a formula containing the RGB function.

BeginArrow (Line Format section)

Indicates whether a line has an arrowhead or other line end format at its first vertex. Enter a number from 0 to 46, or use the Line command. The size of the arrowhead is set in the ArrowSize cell.

0	No arrowhead
1 - 46	Assorted arrowhead styles that correspond to indexed entries in the Line dialog box

TIP To specify a custom line end, use the USE function in this cell.

EndArrow (Line Format section)

Indicates whether a line has an arrowhead or other line end format at its last vertex. Enter a number from 0 to 46, or use the Line command. The size of the arrowhead is set in the ArrowSize cell.

0	No arrowhead
1 - 46	Assorted arrowhead styles that correspond to indexed entries in the Line dialog box

TIP To specify a custom line end, use the USE function in this cell.

ArrowSize (Line Format section)

Indicates the size of the arrowhead set in BeginArrow and EndArrow. Enter a number from 0 to 6, or use the Line command.

0	Very small
1	Small
2	Medium
3	Large
4	Extra large
5	Jumbo
6	Colossal

Rounding (Line Format section)

Indicates the radius of the rounding circle applied where two straight segments of a shape meet. For example, rounding can be used to give a rectangle rounded corners. To set rounding, enter a value with units of measure, or use the Line command.

Fill Format section

The Fill Format section shows the current fill formatting attributes for the shape and the shape's drop shadow, including pattern, foreground color, and background color. Fill formats can be set with the Fill command, by applying a fill style, or by making an entry in a Fill Format cell.

The Fill Format section includes these cells:

FillBkgnd
FillForegnd
FillPattern
ShdwBkgnd
ShdwForegnd
ShdwPattern

FillPattern (Fill Format section)

The fill pattern for the shape. To set the fill pattern, type one of these numbers or use the Fill command.

0	None (transparent fill)
1	Solid foreground color
2 - 40	Assorted fill patterns that correspond to indexed entries in the Line dialog box

TIP To specify a custom fill pattern, use the USE function in this cell.

FillForegnd (Fill Format section)

The color used for the foreground (stroke) of the shape's fill pattern. To set the color, enter a number from 0 to 23. The number corresponds to a color in the current color palette, which is set with the Color Palette command.

A value of 24 or higher indicates a custom color that Visio specifies as a formula containing the RGB function.

FillBkgnd (Fill Format section)

The color used for the background (fill) of the shape's fill pattern. To set the color, enter a number from 0 to 23. The number corresponds to a color in the current color palette, which is set with the Color Palette command.

A value of 24 or higher indicates a custom color that Visio specifies as a formula containing the RGB function.

ShdwPattern (Fill Format section)

The fill pattern for the shape's drop shadow. To set the fill pattern, type one of these numbers or use the Fill command.

0	None (transparent fill)
1	Solid foreground color
2 - 24	Assorted patterns

TIP To specify a custom pattern, use the USE function in this cell.

ShdwForegnd (Fill Format section)

The color used for the foreground (stroke) of the shape's drop shadow fill pattern. To set the color, enter a number from 0 to 23. The number corresponds to a color in the current color palette, which is set with the Color Palette command.

A value of 24 or higher indicates a custom color that Visio specifies as a formula containing the RGB function.

ShdwBkgnd (Fill Format section)

The color used for the background (fill) of the shape's drop shadow fill pattern. To set the color, enter a number from 0 to 23. The number corresponds to a color in the current color palette, which is set with the Color Palette command.

A value of 24 or higher indicates a custom color that Visio specifies as a formula containing the RGB function.

Character section

The Character section shows the formatting attributes for the shape's text, including font, color, text style, case, position relative to the baseline, and point size. Text formats can be set with the Text command, by applying a text style, or by making an entry in a Character section cell.

The Character section includes these cells:

Case
Color
Font
Pos
Size
Style

Font (Character section)

The number of the font used to format the text. Font numbers vary according to the fonts installed on your system. The number 0 represents the default font, which is Arial, unless you change the default font in the VISIO.INI file.

Color (Character section)

The color used for the shape's text. To set the color, enter a number from 0 to 23. The number corresponds to a color in the current color palette, which is set with the Color Palette command.

Style (Character section)

Shows the character formatting applied to the text in the shape's text block. The value represents a binary number in which each bit indicates a character style.

Bit:	**8**	**4**	**2**	**1**
Style:	Small caps	Underline	Italic	Bold

For example, a value of 3 represents text formatted in both italic and bold. If the value of Style is 0, the text is plain, or unformatted.

To test for a particular format, use the Boolean functions BITAND( ), BITNOT( ), BITOR( ),or BITXOR( ).

Case (Character section)

The case of a shape's text. All capitals (1) and initial capitals (2) do not change the appearance of text that was entered in all capital letters. The text must be entered in lowercase letters for these options to show an effect.

0	Normal case
1	All capital (uppercase) letters
2	Initial capital letters only

Pos (Character section)

The position of the shape's text relative to the baseline.

0	Normal position
1	Superscript
2	Subscript

Size (Character section)

The size of the text in the shape's text block, in the default units for text set in the Options dialog box. The text's size is independent of the scale of the drawing. If the drawing is scaled, the text size remains the same.

Paragraph section

The Paragraph section shows the paragraph formatting attributes for the shape's text, including indents, line spacing, and horizontal alignment of paragraphs. Paragraph formats can be set with the Text command, by applying a text style, or by making an entry in a Paragraph section cell.

The Paragraph section includes these cells:

HAlign
IndFirst
IndLeft
IndRight
SpAfter
SpBefore
SpLine

The Paragraph section does not include cells for tab settings. To set tabs for a shape's text, use the Text command.

IndFirst (Paragraph section)

The distance the first line of each paragraph in the shape's text block is indented from the left indent of the paragraph. This value is independent of the scale of the drawing. If the drawing is scaled, the first line indent remains the same.

IndLeft (Paragraph section)

The distance all lines of text in each paragraph are indented from the left margin of the text block. This value is independent of the scale of the drawing. If the drawing is scaled, the left indent remains the same.

IndRight (Paragraph section)

The distance all lines of text in each paragraph are indented from the right margin of the text block. This value is independent of the scale of the drawing. If the drawing is scaled, the right indent remains the same.

SpLine (Paragraph section)

The distance between one line of text and the next, expressed as a percentage, where 100% is the height of a text line. SpLine can have these values:

>0	Absolute spacing, regardless of type size
=0	Set solid (spacing = 100% of type size)
<0	A percentage of type size (for example, -120% yields 120% spacing)

If SpLine is less than 100%, lines of text overlap. This value is independent of the scale of the drawing. If the drawing is scaled, the line spacing remains the same.

SpBefore (Paragraph section)

The amount of space inserted before each paragraph in the shape's text block, in addition to any space from SpLine and, if it is the first paragraph in a text block, TopMargin. This value is independent of the scale of the drawing. If the drawing is scaled, the Space Before setting remains the same.

SpAfter (Paragraph section)

The amount of space inserted after each paragraph in the shape's text block, in addition to any space from SpLine and, if it is the first paragraph in a text block, BottomMargin. This value is independent of the scale of the drawing. If the drawing is scaled, the Space After setting remains the same.

HAlign (Paragraph section)

The horizontal alignment of text in the shape's text block. This cell can also be referred to as Para.HorzAlign. If you create a cell reference in a formula by clicking this cell, Visio inserts the name Para.HorzAlign. HAlign can have these values:

0	Left align
1	Center
2	Right align
3	Justify
4	Force justify

Justify adds space between words in every line except the last line of the paragraph to make both the left and right sides of text flush with the margins. Force justify justifies every line in the paragraph, including the last.

Text Block Format section

The Text Block Format section shows the alignment and margins of text in a shape's text block and includes these cells:

BottomMargin
LeftMargin
RightMargin
TextBkgnd
TopMargin
VerticalAlign

VerticalAlign (Text Block Format section)

The vertical alignment of text within the text block. VerticalAlign can have these values:

0	Top
1	Middle
2	Bottom

TopMargin (Text Block Format section)

The distance between the top border of the text block and the first line of text it contains. The default is 0.1 inch. This value is independent of the scale of the drawing. If the drawing is scaled, the top margin remains the same.

BottomMargin (Text Block Format section)

The distance between the bottom border of the text block and the last line of text it contains. The default is 0.1 inch. This value is independent of the scale of the drawing. If the drawing is scaled, the bottom margin remains the same.

LeftMargin (Text Block Format section)

The distance between the left border of the text block and the text it contains. The default is 0.1 inch. This value is independent of the scale of the drawing. If the drawing is scaled, the left margin remains the same.

RightMargin (Text Block Format section)

The distance between the right border of the text block and the text it contains. The default is 0.1 inch. This value is independent of the scale of the drawing. If the drawing is scaled, the right margin remains the same.

TextBkgnd (Text Block Format Section)

The text background color for a shape. TextBkgnd can have any value between 0 and 24, or 255 (0 and 255 indicate a transparent text background). To select a color value, add 1 to the value displayed in the Color Palette dialog box.

Text Fields section

Related Topics

The Text Fields section displays custom formulas inserted in the shape's text using the Field command. Formulas appear in the Text Fields section in the order they were created, not in the order they appear in the text. To add this section, use the Section command.

To refer to the value of a Text Fields cell in a formula, use the name Fields.Value[**n**] in your cell reference, where **n** refers to the row number. For example, Fields.Value[2].

Text Transform section

The Text Transform section contains positioning information about a shape's text block. To add this section, use the Section command.

The Text Transform section includes these cells:

TxtAngle
TxtHeight
TxtLocPinX
TxtLocPinY
TxtPinX
TxtPinY
TxtWidth

TxtWidth (Text Transform section)

The width of the text block. The default formula is:

```
= Width*1
```

which evaluates to the width of the shape.

TxtHeight (Text Transform section)

The height of the text block. The default formula is:

```
= Height*1
```

which evaluates to the height of the shape.

TxtAngle (Text Transform section)

The text block's current angle of rotation in relation to the x-axis of the shape. The default is 0 degrees.

TxtPinX (Text Transform section)

Related Topics

The x-coordinate of the text block's center of rotation in relation to the origin of the shape. The default formula is:

```
= Width*0.5
```

which evaluates to the horizontal center of the shape.

TxtPinY (Text Transform section)

Related Topics

The y-coordinate of the text block's center of rotation in relation to the origin of the shape. The default formula is:

```
= Height*0.5
```

which evaluates to the vertical center of the shape.

TxtLocPinX (Text Transform section)

Related Topics

The x-coordinate of the text block's center of rotation in relation to the origin of the text block. The default formula is:

```
= TxtWidth*0.5
```

which evaluates to the horizontal center of the text block.

TxtLocPinY (Text Transform section)

Related Topics

The y-coordinate of the text block's center of rotation relative to the origin of the text block. The default formula is:

```
= TxtHeight*0.5
```

which evaluates to the vertical center of the text block.

Scratch section

The Scratch section is a work area for entering and testing formulas that are referred to by other cells. To add this section, use the Section command.

Cells in the Scratch section use units in two different ways. X and Y cells use drawing units; A through D cells are unitless. (In C programmers' jargon, X and Y cells are "typed," and cells A through D are "void.") The Scratch X and Y cells are often used for deriving x- and y-coordinates, such as PinX and PinY, or the X and Y cells found in a Geometry section cell. Scratch cells A through D can display whatever units you specify.

A further difference is the way these cells store point values. A point in Visio is a single data package for an (x,y) coordinate. When a formula returns a point value, that value is interpreted in one of three ways, depending on the ShapeSheet cell the formula is in. Cells that relate to x-coordinates (for example, PinX, or cells in the X column of a Geometry section) extract just the x-coordinate part of a point value. Cells that relate to y-coordinates extract just the y-coordinate part of a point value. A cell that is unitless extracts the distance from (0,0) to the point.

For example, Visio extracts the formula PNT(3,4) in these three ways:

Cell	If you enter	Visio treats it like	Result
X	PNT(3,4)	PNTX(PNT(3,4))	3
Y	PNT(3,4)	PNTY(PNT(3,4))	4
A-D	PNT(3,4)	SQRT(3^2+4^2)	5

Connection Points section

The Connection Points section contains cells for each connection point defined for the shape. To add this section, use the Section command.

You can change the row type of a connection point in this section to display additional cells that you can use for scratch calculations. To do this, right-click the row index, then choose Change Row Type.

The Connection Points section includes this cell:

Connections.Row

Connections.Row (Connection Points section)

The x- and y-coordinates of each connection point defined for the shape. Coordinates of connection points are measured from the origin of the shape.

You can rename this cell to create a more meaningful reference for the value contained in the X or Y cell of the same row. The cell name you enter must be unique within the section. When you create a name for one cell in this section, Visio names all the cells in the section with the default name, Connections.Row_*n*. If no rows in the section are named, the name cell is blank.

For example, to rename the cell for the first row, type *Custom* in the formula bar to create the cell name Connections.Custom. Visio creates the name Connections.Row_2 for the cell in the second row. To refer to the X cell of the first row, use Connections.Custom.X or Connections.X1. To refer to the Y cell of the first row, use Connections.Custom.Y1 or Connections.Y1. To refer to the X cell of the second row, use Connections.Row_2.X or Connections.X2, and to refer to its Y cell, use Connections.Row_2.Y or Connections.Y2.

NOTE Named connection point rows are not compatible with versions of Visio earlier than 5.0. When saving a Visio 5.0 drawing file to an older format, references to named connection point rows are converted to indexed references, and the row names are lost.

Controls section

The Controls section contains cells for the x- and y-coordinates of each control handle defined for a shape and cells that specify the way the control handle should behave.

The Controls section includes these cells:

Can Glue
X
X Behavior
X Dynamics
Y
Y Behavior
Y Dynamics
Tip

X (Controls section)

The x-coordinate that indicates the location of a shape's control handle.

Y (Controls section)

The y-coordinate that indicates the location of a shape's control handle.

X Dynamics (Controls section)

Related Topics

The x-coordinate for a control handle's anchor point. The anchor point is used for rubber-banding during dynamics.

Y Dynamics (Controls section)

J Related Topics

The y-coordinate for a control handle's anchor point. The anchor point is used for rubber-banding during dynamics.

X Behavior (Controls section)

Related Topics

Controls the type of behavior the x-coordinate of the control handle will exhibit after the handle is moved. These formulas are available:

Value	Behavior	Definition
0	Proportional	The control handle can be moved, and it also moves in proportion with the shape when it is stretched.
1	Proportional locked	The control handle moves in proportion with the shape but the control handle itself cannot be moved.
2	Offset from left edge	The control handle is offset a constant distance from the left side of the shape.
3	Offset from center	The control handle is offset a constant distance from the center of the shape.
4	Offset from right edge	The control handle is offset a constant distance from the right side of the shape.
5	Proportional, hidden	Same as 0, but the control handle is not visible.
6	Proportional locked, hidden	Same as 1, but the control handle is not visible.
7	Offset from left edge, hidden	Same as 2, but the control handle is not visible.
8	Offset from center, hidden	Same as 3, but the control handle is not visible.
9	Offset from right edge, hidden	Same as 4, but the control handle is not visible.

Y Behavior (Controls section)

Related Topics

Controls the type of behavior the y-coordinate of the control handle will exhibit after the handle is moved. These formulas are available:

Value	Behavior	Definition
0	Proportional	The control handle can be moved, and it also moves in proportion with the shape when it is stretched.
1	Proportional locked	The control handle moves in proportion with the shape, but the control handle itself cannot be moved.
2	Offset from bottom edge	The control handle is offset a constant distance from the bottom of the shape.
3	Offset from center	The control handle is offset a constant distance from the center of the shape.
4	Offset from top edge	The control handle is offset a constant distance from the top of the shape.

5	Proportional, hidden	Same as 0, but the control handle is not visible.
6	Proportional locked, hidden	Same as 1, but the control handle is not visible.
7	Offset from left edge, hidden	Same as 2, but the control handle is not visible.
8	Offset from center, hidden	Same as 3, but the control handle is not visible.
9	Offset from right edge, hidden	Same as 4, but the control handle is not visible.

Can Glue (Controls section)

Controls whether a control handle can be glued to other shapes.

non-0	Control handle can be glued.
0	Control handle cannot be glued.

Tip (Controls section)

A descriptive text string that appears as a tool tip when a user pauses the pointer over a shape's control handle. Visio automatically encloses the tip string in quotation marks in the cell, but the quotation marks are not displayed in the tool tip.

1-D Endpoints section

The 1-D Endpoints section contains x- and y-coordinates of the begin point and end point of a 1-D shape. This section appears for 1-D shapes only.

This section contains these cells:

BeginX
BeginY
EndX
EndY

BeginX (1-D Endpoints section)

Related Topics

The x-coordinate of the begin point of the 1-D shape, in relation to the origin of its parent.

BeginY (1-D Endpoints section)

Related Topics

The y-coordinate of the begin point of the 1-D shape, in relation to the origin of its parent.

EndX (1-D Endpoints section)

The x-coordinate of the end point of the 1-D shape, in relation to the origin of its parent.

EndY (1-D Endpoints section)

The y-coordinate of the end point of the 1-D shape, in relation to the origin of its parent.

Alignment section

The Alignment section indicates the alignment of a shape with respect to the guide or guide point to which the shape is glued. The Alignment section appears only for shapes that are glued to guides.

Formulas in this section refer to the Guide Info section of the guide to which the shape is glued.

The Alignment section includes these cells:

AlignBottom
AlignCenter
AlignLeft
AlignMiddle
AlignRight
AlignTop

AlignLeft (Alignment section)

The horizontal position, relative to the origin of its parent, of a vertical guide or guide point to which the shape's left border is aligned.

AlignCenter (Alignment section)

The horizontal position, relative to the origin of its parent, of a vertical guide or guide point to which the shape's horizontal center is aligned.

AlignRight (Alignment section)

The horizontal position, relative to the origin of its parent, of a vertical guide or guide point to which the shape's right border is aligned.

AlignTop (Alignment section)

The vertical position, relative to the origin of its parent, of a horizontal guide or guide point to which the shape's top border is aligned.

AlignMiddle (Alignment section)

The vertical position, relative to the origin of its parent, of a horizontal guide or guide point to which the shape's vertical center is aligned.

AlignBottom (Alignment section)

The vertical position, relative to the origin of its parent, of a horizontal guide or guide point to which the shape's bottom border is aligned.

Guide Info section

The Guide Info section contains the x- and y-coordinates of a guide or guide point. Shapes that are aligned to the guide or guide point refer to its Guide Info section for their values.

The Guide Info section includes these cells:

Angles
PinX
PinY

PinX (Guide Info section)

The x-coordinate of a vertical guide or guide point relative to the origin of the page. For a horizontal guide, PinX is 0.

The pin position is the guide's center of rotation.

To change the value of this cell, use the Size & Position command, or type a value in this cell.

PinY (Guide Info section)

The y-coordinate of a horizontal guide or guide point relative to the origin of the page. For a vertical guide, PinY is 0.

The pin position is the guide's center of rotation.

To change the value of this cell, use the Size & Position command, or type a value in this cell.

Angle (Guide Info section)

A guide's angle of rotation with respect to the bottom and left edges of the page. The center of rotation is PinY for a horizontal guide and PinX for a vertical guide.

To change the value of this cell, use the Size & Position command, or type a value in this cell.

Image Info section

The Image Info section contains the width and height of an object from another program used in a Visio drawing and appears only in the ShapeSheet window for such objects. It also indicates the distance the object's image is offset within its borders.

The Image Info section includes these cells:

ImgHeight
ImgOffsetX
ImgOffsetY
ImgWidth

ImgWidth (Image Info section)

The width of the object's image within its border. The default formula is:

```
= Width*1.
```

Cropping the object changes the factor by which Width is multiplied.

ImgHeight (Image Info section)

The height of the object's image within its border. The default formula is:

```
= Height*1.
```

Cropping the object changes the factor by which Height is multiplied.

ImgOffsetX (Image Info section)

The distance the object is offset horizontally from the origin of the object's border. The default is 0. Panning the object with the crop tool changes this value.

ImgOffsetY (Image Info section)

The distance the object is offset vertically from the origin of the object's border. The default is 0. Panning the object with the crop tool changes this value.

Events section

The Events section contains formulas that control shape events. Event cells are evaluated only when the event occurs, not upon formula entry.

The Events section includes these cells:

EventDblClick
EventDrop
EventXFMod
TheData
TheText

TheText (Events section)

Related Topics

An event cell associated with changes to a shape's text. The cell refers to the current shape unless it is preceded by a reference to another shape on the page.

TheText is reevaluated whenever the shape's text changes and can be used to trigger recalculations (for example, to recalculate the text width and height with the TEXTWIDTH( ) and TEXTHEIGHT( ) functions).

NOTE Event cells are evaluated only when the event occurs, not upon formula entry.

EventDblClick (Events section)

Related Topics

An event cell associated with a shape being double-clicked. The cell refers to the current shape unless it is preceded by a reference to another shape on the page.

NOTE Event cells are evaluated only when the event occurs, not upon formula entry.

EventXFMod (Events section)

Related Topics

An event cell associated with a transformation ("XF") in a shape's position or orientation on the page. The cell refers to the current shape unless it is preceded by a reference to another shape on the page.

NOTE Event cells are evaluated only when the event occurs, not upon formula entry.

EventDrop (Events section)

Related Topics

An event cell associated with a shape being dropped on the drawing page, either as an instance or when a shape is duplicated or pasted. The cell refers to the current shape unless it is preceded by a reference to another shape on the page.

NOTE Event cells are evaluated only when the event occurs, not upon formula entry.

TheData (Events section)

Related Topics

TheData is reserved for future use.

Miscellaneous section

The Miscellaneous section contains object properties for the selected shape, including selection highlighting and visibility.

The Miscellaneous section includes these cells:

DynFeedback
HideText
NoAlignBox
NoCtlHandles
NoObjHandles
NonPrinting
ObjBehavior
ObjInteract
ObjType
UpdateAlignBox

DynFeedback (Miscellaneous section)

J Related Topics

Changes the type of visual feedback provided to users when they drag a connector. When the mouse button is released, the resulting connector shape is not affected by this setting. This setting does not apply to routable connectors.

0	Remains straight (no legs)
1	Shows three legs when dragged
2	Shows five legs when dragged

NoObjHandles (Miscellaneous section)

Related Topics

Toggles the display of selection handles on and off for the selected shape. Upon setting this cell to a nonzero value (TRUE), the selection handles will not be displayed when the shape is selected.

non-0	Selection handles not displayed
0	Selection handles displayed

NoCtlHandles (Miscellaneous section)

Related Topics

Toggles the display of control handles on and off for the selected shape. Upon setting this cell to a nonzero value (TRUE), the control handles will not be displayed when the shape is selected.

non-0	Control handles not displayed
0	Control handles displayed

NoAlignBox (Miscellaneous section)

Related Topics

Toggles the display of the selection rectangle on and off for the selected shape. Upon setting this cell to a nonzero value (TRUE), the selection rectangle will not be displayed when the shape is selected.

non-0	Selection rectangle not displayed
0	Selection rectangle displayed

NonPrinting (Miscellaneous section)

J Related Topics

Toggles printing on and off for the selected shape. Upon setting this cell to a nonzero value (TRUE), the shape will be displayed in the drawing window but will not print.

non-0	Printing disabled
0	Printing enabled

UpdateAlignBox (Miscellaneous section)

Related Topics

Recalculates the selection rectangle whenever a control handle is moved.

HideText (Miscellaneous section)

Related Topics

Hides the text for a shape. Upon setting this cell to a nonzero value (TRUE), the text does not appear on the screen and does not print. You can view text, edit properties, and apply styles to the text in the text block, although the changes will not appear until you reset HideText to 0 (FALSE).

non-0	Text is hidden.
0	Text is not hidden.

ObjBehavior (Miscellaneous section)

Related Topics

The behavior of a placeable or routable shape in a connected diagram. To set the behavior, type one of these numbers:

0	Use behavior set for the page.
1	Create right angle connections.
2	Create straight connections.
5	Create a flowchart with north-south orientation.
6	Create a flowchart with west-east orientation.
7	Create a tree diagram with north-south orientation.
8	Create a tree diagram with west-east orientation.

ObjInteract (Miscellaneous section)

J Related Topics

The interaction style of a placeable or routable shape in a connected diagram. To set the interaction style, type one of these numbers.

0	Unassigned

ObjType (Miscellaneous section)

Related Topics

In a diagram that uses automatic layout, the object type: placeable or routable.

By default, ObjType is set to No Formula for a shape, which evaluates to 0, meaning that Visio determines whether the shape can be placeable depending on its context. For example, if you draw a simple rectangle, the value of its ObjType cell is 0. If you then use the Connect Shapes command or the Dynamic Connector tool to connect the rectangle to another shape, Visio resets the value of the rectangle's ObjType cell to 1 (placeable).

To set the type, use the Behavior command or type one of the following numbers.

0	Visio decides based on the drawing context.
1	Shape is placeable.
2	Shape is routable.
3	Unassigned.
4	Shape is not placeable, not routable.

Glue Info section

The Glue Info section contains formulas generated for a 1-D shape by Visio when the 1-D shape is glued to other shapes. This section contains these cells:

BegTrigger
EndTrigger
GlueType
WalkPreference

GlueType (Glue Info section)

Related Topics

Determines whether a 1-D shape uses static (point-to-point) or dynamic (shape-to-shape) glue when it is glued to another shape.

0	Uses static glue.
1	Unassigned.
2	Unassigned.
3	Uses dynamic glue. Visio writes default formulas in the BegTrigger and EndTrigger cells.

WalkPreference (Glue Info section)

Determines whether an endpoint of a 1-D shape moves to a horizontal or vertical connection point on the shape it is glued to, using dynamic glue, when the shape is moved to an ambiguous position.

0	Both endpoints of the 1-D shape move to horizontal connection points (side-to-side connections).
1	The begin point of the 1-D shape moves to a vertical connection point, and the end point moves to a horizontal connection point (top-to-side or bottom-to-side connections).
2	The begin point of the 1-D shape moves to a horizontal connection point, and the end point moves to a vertical connection point (side-to-top or side-to-bottom connections).
3	Both endpoints of the 1-D shape move to vertical connection points (top-to-bottom connections).

BegTrigger (Glue Info section)

Related Topics

Contains a trigger formula generated by Visio that determines whether to move the begin point of a 1-D shape to maintain its connection to another shape.

When you glue a 1-D shape to another shape using dynamic glue, Visio generates a formula that refers to the EventXFMod cell of the other shape. When that shape is changed, Visio recalculates any formula that refers to its EventXFMod cell, including the formula in the BegTrigger cell. Other ShapeSheet formulas for the 1-D shape refer to the BegTrigger cell and move the begin point of the 1-D shape or alter the shape as needed.

EndTrigger (Glue Info section)

Related Topics

Contains a trigger formula generated by Visio that determines whether to move the end point of a 1-D shape to maintain its connection to another shape.

When you glue a 1-D shape to another shape using dynamic glue, Visio generates a formula that refers to the EventXFMod cell of the other shape. When that shape is changed, Visio recalculates any formula that refers to its EventXFMod cell, including the formula in the EndTrigger cell. Other ShapeSheet formulas of the 1-D shape refer to the EndTrigger cell and move the end point of the 1-D shape or alter the shape as needed.

Actions section

The Actions section defines custom command names that appear on a shape's or page's shortcut menu and specifies the actions that the commands take.

This section contains these cells:

Action
Checked
Disabled
Menu
Prompt

Action (Actions section)

J Related Topics

Contains the formula to be executed when a user chooses the command name defined in the corresponding Menu cell.

NOTE An Action cell is evaluated only when the action occurs, not upon formula entry.

Menu (Actions section)

Related Topics

Defines the name of the command that appears on a shape's or page's shortcut menu. Prefix the command name with an underscore character (_) to display a divider bar above the command in the menu. Prefix the name with a percent character (%) to display the command at the bottom of the shortcut menu.

Prompt (Actions section)

Related Topics

Specifies a descriptive prompt that appears in the status bar when a user selects the corresponding shortcut command.

Checked (Actions section)

J Related Topics

Toggles the display of a checkmark beside the command name on the shortcut menu.

non-0	Checkmark is displayed.
0	Checkmark is not displayed.

Disabled (Actions section)

Related Topics

Toggles the display of the command name on the shortcut menu, highlighting the name when FALSE and dimming it when TRUE.

non-0	Command name is disabled (dimmed).
0	Command name is not disabled (highlighted).

Custom Properties section

The Custom Properties section contains cells for associating data with a shape.

Custom properties can be defined and edited by using the Custom Properties Editor or the cells of the Custom Properties section. The value of a shape's custom properties can be set in the Custom Properties dialog box or in the Value cell.

This section contains these cells:

Ask
Format
Invisible
Label
Prompt
Prop.Row
SortKey
Type
Value

Ask (Custom Properties section)

Related Topics

Determines whether a user is queried to enter custom property information for a shape when an instance is created or the shape is duplicated or copied. If true, Visio displays the Custom Properties dialog box every time any of these events is triggered.

non-0	Ask user to enter custom property data.
0	Do not ask user to enter data.

Format (Custom Properties section)

Related Topics

Specifies the formatting of a custom property that is a string (Type = 0), a fixed list (Type = 1), a number (Type = 2), a variable list (Type = 4), a date or time (Type = 5), a duration (Type = 6), or a currency (Type = 7).

"picture"	If the value of the Type cell is 0, 2, 5, 6, or 7 (string, number, date or time, duration, or currency), specify a format picture appropriate for the data type. For example, the format picture "# #/4 UU" formats the number 12.43 in. as 12 2/4 INCHES.
"item;...;item"	If the value of the Type cell is 1 or 4 (fixed or variable list), specify the items to appear in the list, separated by semicolons. For example, "Engineering;Human Resources;Sales;Marketing".

Date values (Type = 5) are displayed in the short date format. Currency values (Type = 7) are displayed using the user's current Regional Settings for currency.

A number (Type = 2) can represent a dimension, scalar, angle, date, time, or currency. To ensure that an input number is always evaluated as a date, time, or currency, use the DATETIME or CY function in the Format cell instead of a format picture.

Invisible (Custom Properties section)

Related Topics

Specifies whether the custom property is visible in the Custom Properties dialog box.

non-0	Custom property is not visible.
0	Custom property is visible.

Label (Custom Properties section)

Related Topics

Specifies the label that appears to users in the Custom Properties dialog box. A label consists of alphanumeric characters, including the underscore (_) character.

Visio automatically encloses the Label string in quotation marks in the cell, but the quotation marks are not displayed in the dialog box.

If no label text is found, Visio displays the row name (Prop.Row) in the dialog box.

Prompt (Custom Properties section)

Related Topics

Specifies descriptive or instructional text that appears to users in the Custom Properties dialog box when the property is selected.

Prop.Row (Custom Properties section)

J Related Topics

Specifies the row label, which appears in the ShapeSheet window in red text. For example, type "Price" in the formula bar to create the custom property Prop.Price.

You can use this name in cell references to refer to the value in the Value cell of the corresponding row. You can also use this name to reference other cells in the same row by including their column name, for example, Prop.Price.Prompt.

SortKey (Custom Properties section)

Related Topics

Specifies a key by which items in the Custom Properties dialog box are listed. The sort is locale specific, case insensitive, and descending. The sort key is a string, which Visio automatically encloses in quotation marks.

For example, enter the following sort keys to display the custom properties in the dialog box in the order Item Number, Quantity, Price:

Row	Label	SortKey
Prop.Item	Item Number	1
Prop.Price	Price	3
Prop.Quan	Quantity	2

Type (Custom Properties section)

Related Topics

Specifies a data type for the custom property value.

0	String. This is the default.
1	Fixed list. Displays the list items in a drop-down combo box in the Custom Properties dialog box. Specify the list items in the Format cell. Users can select only one item from the list.
2	Number. Includes date, time, duration, and currency values as well as scalars, dimensions, and angles. Specify a format picture in the Format cell.
3	Boolean. Displays FALSE and TRUE as items users can select from a drop-down list box in the Custom Properties dialog box.
4	Variable list. Displays the list items in a drop-down combo box in the Custom Properties dialog box. Specify the list items in the Format cell. Users can select a list item or enter a new item that is added to the current list in the Format cell.
5	Date or time value. Displays days, months, and years, or seconds, minutes, and hours, or a combined date and time value. Specify a format picture in the Format cell.
6	Duration value. Displays elapsed time. Specify a format picture in the Format cell.
7	Currency value. Uses the system's current Regional Settings. Specify a format picture in the Format cell.

Value (Custom Properties section)

Related Topics

Contains the property's value as entered in the Custom Properties dialog box.

You can enter a formula in this cell, but its value will be overwritten by the value entered in the Custom Properties dialog box, even if you use the GUARD function to protect the formula.

Layer Membership section

Related Topics

The Layer Membership section contains one row that lists each layer to which the shape is assigned.

The layer assignment is shown as an index to the list of layers on the page. The layer index corresponds to a layer name in the Layer Properties dialog box. The first name in the dialog box is layer 0, the second is layer 1, and so forth.

If a shape is assigned to more than one layer, each layer index appears in the Layer Membership cell separated by a semicolon.

To refer to the value of the Layer Membership cell in a formula, use the name LayerMember in your cell reference.

Layers section

The Layers section shows all layers defined for the page and the properties set for each layer. This section is a page property that can be viewed only in the ShapeSheet window for a page. It contains the following cells, which correspond to the options in the Layer Properties dialog box:

Active
Color
Glue
Lock
Print
Snap
Visible

Active (Layers section)

Related Topics

Specifies whether the layer is the default active layer. Shapes without preassigned layers are assigned to the active layer when dropped on the drawing page. Corresponds to the Active option in the Layer Properties dialog box.

non-0	Active layer
0	Not an active layer

Color (Layers section)

Related Topics

Specifies the index of the color used to display the layer. Corresponds to the Layer Color option in the Layer Properties dialog box.

Glue (Layers section)

Related Topics

Specifies whether shapes belonging to the layer can be glued. Corresponds to the Glue option in the Layer Properties dialog box.

non-0	Glue is enabled.
0	Glue is disabled.

Lock (Layers section)

Related Topics

Specifies whether shapes belonging to the layer are locked against being selected or edited. Corresponds to the Lock option in the Layer Properties dialog box.

non-0	Shapes are locked.
0	Shapes are not locked.

Print (Layers section)

Related Topics

Specifies whether shapes belonging to the layer can be printed. Corresponds to the Print option in the Layer Properties dialog box.

non-0	Shapes can be printed.
0	Shapes cannot be printed.

Snap (Layers section)

Related Topics

Specifies whether shapes belonging to the layer snap to the grid. Corresponds to the Snap option in the Layer Properties dialog box.

non-0	Shapes snap to grid.
0	Shapes do not snap to grid.

Visible (Layers section)

Related Topics

Specifies whether shapes belonging to the layer are visible on the drawing page. Corresponds to the Visible option in the Layer Properties dialog box.

non-0	Shapes are visible.
0	Shapes are hidden.

Ruler & Grid section

The Ruler & Grid section shows the current settings of the page's rulers and grid. This section is a page property that can be viewed only in the ShapeSheet window for a page. It contains the following cells, which correspond to the settings in the Ruler & Grid dialog box:

XGridDensity
XGridOrigin
XGridSpacing
XRulerDensity
XRulerOrigin
YGridDensity
YGridOrigin
YGridSpacing
YRulerDensity
YRulerOrigin

XGridDensity (Ruler & Grid section)

J Related Topics

Specifies the type of horizontal grid to use. Corresponds to the Horizontal Grid Spacing option in the Ruler & Grid dialog box.

0	Fixed
2	Coarse
4	Normal (default)
8	Fine

NOTE If an invalid value is entered, it specifies the same type as 4.

XGridSpacing (Ruler & Grid section)

J Related Topics

Specifies the distance between horizontal lines in a fixed grid (XGridDensity = 0). Corresponds to the Horizontal Minimum Spacing option in the Ruler & Grid dialog box.

XGridOrigin (Ruler & Grid section)

Related Topics

Specifies the horizontal coordinate of the grid origin. Corresponds to the Horizontal Grid Origin option in the Ruler & Grid dialog box.

XRulerDensity (Ruler & Grid section)

Related Topics

Specifies the horizontal subdivisions on the ruler for the page. Corresponds to the Horizontal Subdivision option in the Ruler & Grid dialog box.

8	Coarse
16	Normal
32	Fine

NOTE If an invalid value or no value is entered, it specifies the same subdivisions as 16.

XRulerOrigin (Ruler & Grid section)

J Related Topics

Specifies the zero point on the x-axis ruler for the page. Corresponds to the Horizontal Ruler Zero option in the Ruler & Grid dialog box.

YGridDensity (Ruler & Grid section)

J Related Topics

Specifies the type of vertical grid to use. Corresponds to the Vertical Grid Spacing option in the Ruler & Grid dialog box.

0	Fixed
2	Coarse
4	Normal
8	Fine

NOTE If an invalid value is entered, it specifies the same type as 4.

YGridOrigin (Ruler & Grid section)

Related Topics

Specifies the vertical origin of the grid. Corresponds to the Vertical Grid option in the Ruler & Grid dialog box.

YGridSpacing (Ruler & Grid section)

Related Topics

Specifies the distance between vertical lines in a fixed grid (YGridDensity = 0). Corresponds to the Vertical Minimum Spacing option in the Ruler & Grid dialog box.

YRulerDensity (Ruler & Grid section)

Related Topics

Specifies the vertical subdivisions on the ruler for the page. Corresponds to the Vertical Subdivision option in the Ruler & Grid dialog box.

8	Coarse
16	Normal
32	Fine

NOTE If an invalid value or no value is entered, it specifies the same subdivisions as 16.

YRulerOrigin (Ruler & Grid section)

Related Topics

Specifies the zero point on the y-axis ruler for the page. Corresponds to the Vertical Ruler Zero option in the Ruler & Grid dialog box.

Page Properties section

The Page Properties section describes page attributes.

This section contains these cells:

DrawingScale
DrawingScaleType
DrawingSizeType
PageHeight
PageScale
PageWidth
ShdwOffsetX
ShdwOffsetY

DrawingScale (Page Properties section)

Related Topics

The value of the drawing unit in the current drawing scale. The drawing scale for the page is the ratio of the page unit shown in the PageScale cell to the drawing unit. This cell corresponds to the settings in the Page Setup dialog box.

PageHeight (Page Properties section)

Related Topics

The height of the printed page in drawing units. To set page height, use the Page Setup command or manually resize the page with the mouse.

PageScale (Page Properties section)

Related Topics

The value of the page unit in the current drawing scale. The drawing scale for the page is the ratio of the page unit to the drawing unit shown in the DrawingScale cell. This cell corresponds to the settings in the Page Setup dialog box.

PageWidth (Page Properties section)

Related Topics

The width of the printed page drawing units. To set page width, use the Page Setup command or manually resize the page with the mouse.

ShdwOffsetX (Page Properties section)

Related Topics

The distance in page units that a shape's drop shadow is offset horizontally from the shape. This value is set in the Page Setup dialog box.

ShdwOffsetY (Page Properties section)

J Related Topics

The distance in page units that a shape's drop shadow is offset vertically from the shape. This value is set in the Page Setup dialog box.

DrawingSizeType (Page Properties section)

Related Topics

The drawing size. To set drawing size, use the Page Setup command or manually resize the page with the mouse.

DrawingSizeType can have these values:

0	Same as printer
1	Fit page to drawing contents
2	Standard
3	Custom page size
4	Custom scaled drawing size
5	Metric (ISO)
6	ANSI Engineering
7	ANSI Architectural

DrawingScaleType (Page Properties section)

Related Topics

The drawing scale selected in the Page Setup dialog box. DrawingScaleType can have these values:

0	No Scale
1	Architectural Scale
2	Civil Engineering Scale
3	Custom Scale
4	Metric
5	Mechanical Engineering Scale

User-Defined Cells section

The User-Defined Cells section is a work area for entering and testing formulas that are referred to by other cells and add-on tools. The section contains the following cells:

Prompt
User.Row
Value

Value (User-Defined Cells section)

Related Topics

Specifies a value for the corresponding user-defined cell. To refer to this value in another cell, specify the user-defined name entered in the row label User.Row.

Prompt (User-Defined Cells section)

Related Topics

Specifies a descriptive prompt or comment for the user-defined cell. Visio automatically encloses the prompt text in quotation marks (" ") to indicate that it is a text string.

If you type an equal sign (=) and omit the quotation marks, you can enter a formula in this cell that Visio evaluates.

User.Row (User-Defined Cells section)

Related Topics

Specifies the user-defined cell name for the value contained in the Value cell of the same row. The cell name you enter must be unique within the section. For example, type Offset in the formula bar to create the cell name User.Offset.

Hyperlink section

The Hyperlink section contains cells for creating a jump between a shape or drawing page and another drawing page, another file, or a World Wide Web site.

To define a hyperlink, you can enter values in the cells of this section, use the Hyperlink dialog box.

The Hyperlink section can be added to a shape, group, page, or foreign object in the ShapeSheet window.

The Hyperlink section includes these cells:

Address
Description
ExtraInfo
Frame
NewWindow
SubAddress

Description (Hyperlink section)

Related Topics

A descriptive text string for a hyperlink. Use this cell to store comments about the hyperlink; for example, "Link to our pricing Web site."

To set the value of this cell, you can also use the Hyperlink dialog box.

ExtraInfo (Hyperlink section)

Related Topics

A string that passes information to be used in resolving a URL, such as the coordinates of an image map.

For example, in the ExtraInfo cell, "x=41&y=7" specifies the coordinates of an image map.

Frame (Hyperlink section)

Related Topics

The name of a frame to target when Visio is open as an ActiveX document in an ActiveX container, such as Microsoft Internet Explorer 3.0 or later. The default is an empty string.

SubAddress (Hyperlink section)

Related Topics

Specifies a lcoation within the target document to link to.

For example, if the Address cell is "Drawing1.VSD", the Location cell can specify a page name such as "Page-3". If the Address cell is the Microsoft Excel file "Samples.xls", the value of this cell can be a worksheet or range within a worksheet, such as "Worksheet Functions" or "Sheet1!A1:D10". If the Address cell is "http://www.visio.com/devweb/", the value of this cell can be a named anchor within the document, such as "solutions".

To set the value of this cell, you can also use the Hyperlink dialog box.

NewWindow (Hyperlink section)

Related Topics

Specifies whether to open the hyperlink in a new window. If TRUE, opens the linked page, document, or Web site in a new window. The default is FALSE.

Address (Hyperlink section)

Related Topics

A URL address, DOS file name, or UNC to jump to.

You can specify Address as a relative path based on the base path defined for the document in the Properties dialog box. If the document has no base path, Visio navigates based on the document path. If the document hasn't been saved, the hyperlink is undefined.

To set the value of this cell, you can also use the Hyperlink dialog box.

Categorized list of functions

Alphabetical list of functions

Mathematical functions

ABS(number)
CEILING(number,multiple)
FLOOR(number,multiple)
INT(number)
INTUP(number)
LN(number)
LOG10(number)
MOD(number,divisor)
MODULUS(number,divisor)
PI( )
POW(number,exponent)
RAND( )
ROUND(number,numberofdigits)
SIGN(number,fuzz)
SQRT(number)
SUM(number1,number2...number14)
TRUNC(number,numberofdigits)

Trigonometric functions

ACOS(number)
ANG360(angle)
ASIN(number)
ATAN(number)
ATAN2(y,x)
COS(angle)
COSH(angle)
DEG(angle)
LOC(point)
PAR(point)
PNT(x,y)
PNTX(point)
PNTY(point)
RAD(angle)
SIN(angle)
SINH(angle)
TAN(angle)
TANH(angle)

Geometric functions

GRAVITY(angle,limit1,limit2)
RECTSECT(width,height,x,y,option)

Event functions

CALLTHIS("procedure",["project"],[arg1,arg2,…])
DEFAULTEVENT( )

DEPENDSON(cellref[,cellref2,...])
DOOLEVERB("verb")
RUNADDON("string")
RUNADDONWARGS("filename","arguments")
SETF("cell",formula)

Window management functions

GOTOPAGE("pagename")
OPENFILE("filename")
OPENGROUPWIN( )
OPENPAGE("pagename")
OPENSHEETWIN( )
OPENTEXTWIN( )

Logical functions

AND(logicalexpression1,logicalexpression2,...)
BITAND(binarynumber1,binarynumber2)
BITNOT(binarynumber)
BITOR(binarynumber1,binarynumber2)
BITXOR(binarynumber1,binarynumber2)
IF(logicalexpression,valueiftrue,valueiffalse)
NOT(logicalexpression)
OR(logicalexpression1,logicalexpression2,...)

Error handling functions

ISERR
ISERRNA
ISERROR
ISERRVALUE
NA( )
REF( )

Date and time functions

DATE(year,month,day)
DATETIME("datetime"|expression[,lcid])
DATEVALUE("datetime"|expression[,lcid])
DAY("datetime"|expression[,lcid])
DAYOFYEAR("datetime"|expression[,lcid])
HOUR("datetime"|expression[,lcid])
MINUTE("datetime"|expression[,lcid])
MONTH("datetime"|expression[,lcid])
NOW( )
SECOND("datetime"|expression[,lcid])
TIME(hour,minute,second)
TIMEVALUE("datetime"|expression[,lcid])
WEEKDAY("datetime"|expression[,lcid])
YEAR("datetime"|expression[,lcid])

Statistical functions

MAX(number1,number2,...)
MIN(number1,number2,...)

Text functions

CHAR(number)
EVALTEXT(shapename!TheText)
FORMAT(expression,"formatpicture")
FORMATEX(expression,"formatpicture",[inputunit,][outputunit])
LOWER("string")
TEXTHEIGHT(theText,maximumwidth)
TEXTWIDTH(theText,maximumwidth)
UPPER("string")

Color and pattern management functions

BLUE(expression)
GREEN(expression)
HSL(hue,saturation,luminosity)
HUE(expression)
LUM(expression)
RED(expression)
RGB(red,green,blue)
SAT(expression)
USE("name")

Miscellaneous functions

CY(value [,currency|code])
GUARD(expression)
HELP("filename.hlp!keyword")
HYPERLINK("address"[,"subaddress","extrainfo",window,"frame"])
INDEX(index,list[,[delimiter][,[errorvalue]]])
INTERSECTX(pinx,piny1,angle1,pinx2,piny2,angle2)
INTERSECTY(pinx,piny1,angle1,pinx2,piny2,angle2)
LOOKUP(key,list[,delimiter])
LOTUSNOTES("field")
MAGNITUDE(constantA,A,constantB,B)
PLAYSOUND("filename"|"alias",isAlias,doBeepOnFail,synch)
USERUI(state,defaultexpression,userexpression)

Alphabetical list of functions

J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
J
Y
J
Click a letter to go to that section.

Functions listed by type

A

ABS(number)
ACOS(number)
AND(logicalexpression1,logicalexpression2,...)
ANG360(angle)
ASIN(number)
ATAN(number)
ATAN2(y,x)

B

BITAND(binarynumber1,binarynumber2)
BITNOT(binarynumber)
BITOR(binarynumber1,binarynumber2)
BITXOR(binarynumber1,binarynumber2)
BLUE(expression)

C

CALLTHIS("procedure",["project"],[arg1,arg2,…])
CEILING(number,multiple)
CHAR(number)
COS(angle)
COSH(angle)
CY(value [,currency|code])

D

DATE(year,month,day)
DATETIME("datetime"|expression[,lcid])
DATEVALUE("datetime"|expression[,lcid])
DAY("datetime"|expression[,lcid])
DAYOFYEAR("datetime"|expression[,lcid])
DEFAULTEVENT( )
DEG(angle)
DEPENDSON(cellref,cellref2,...)
DOOLEVERB("verb")

E

EVALTEXT(shape!TheText)

F

FLOOR(number,multiple)
FORMAT(expression,"format")
FORMATEX(expression,"formatstring"[,inputunits][,outputunits])

G

GOTOPAGE("pagename")
GRAVITY(angle,limit1,limit2)
GREEN(expression)
GUARD(expression)

H

HELP("filename.hlp!keyword")
HOUR("datetime"|expression[,lcid])
HSL(hue,saturation,luminosity)
HUE(expression)
HYPERLINK("address"[,"subaddress","extrainfo",window,"frame"])

I - K

IF(logicalexpression,valueiftrue,valueiffalse)
INDEX(index,list[,[delimiter][,[errorvalue]]])
INT(number)
INTERSECTX(pinx,piny1,angle1,pinx2,piny2,angle2)
INTERSECTY(pinx,piny1,angle1,pinx2,piny2,angle2)
INTUP(number)
ISERR(cellref)
ISERRNA(cellref)
ISERROR(cellref)
ISERRVALUE(cellref)

L

LN(number)
LOC(point)
LOG10(number)
LOOKUP(key,list[,delimiter])
LOTUSNOTES("field")

LOWER("string")
LUM(expression)

M

MAGNITUDE(constantA,A,constantB,B)
MAX(number1,number2,...)
MIN(number1,number2,...)
MINUTE("datetime"|expression[,lcid])
MOD(number,divisor)
MODULUS(number,divisor)
MONTH("datetime"|expression[,lcid])

N

NA( )
NOT(logicalexpression)
NOW( )

O

OPENFILE("filename")
OPENGROUPWIN( )
OPENPAGE("pagename")
OPENSHEETWIN( )
OPENTEXTWIN( )
OR(logicalexpression1,logicalexpression2,...)

P - Q

PAR(point)
PI( )
PLAYSOUND("filename"|"alias",isAlias,beep,synch)
PNT(x,y)
PNTX(point)
PNTY(point)
POW(number,exponent)

R

RAD(angle)
RAND( )
RECTSECT(width,height,x,y,option)
RED(expression)
REF( )
RGB(red,green,blue)
ROUND(number,numberofdigits)
RUNADDON("string")
RUNADDONWARGS("filename","arguments")

S

SAT(expression)
SECOND("datetime"|expression[,lcid])
SETF("cell",formula)
SIGN(number,fuzz)
SIN(angle)

ABS

Related Topics

Syntax ABS(number)

Returns the absolute value of **number**.

Example

ABS(-1.24)

Returns 1.24.

AND

Related Topics

Syntax AND(logicalexpression1,logicalexpression2,...,logical expression14)

Returns 1 if all of the logical expressions supplied are true. (Any expression that evaluates to a nonzero value is considered to be true.) If any of the logical expressions are false or equal 0, AND returns 0.

Example

AND(Height > 1,PinX > 1)

Returns 1 if both expressions are true. Returns 0 if either expression is false.

ANG360

Related Topics

Syntax ANG360(angle)

Normalizes an angle's range to be 0 <= result < 2PI radians (0 <= result < 360 deg.). If **angle** is not specified using angular units, it is interpreted as radians. If **angle** cannot be converted to a value, a #VALUE! error is returned.

Examples

ANG360(395 deg.)

Equals 35 deg.

ANG360(-9.8 rad.)

Equals 2.7664 rad.

ANG360(45)

Equals 58.31 deg. (1.0177 rad.)

ACOS

Related Topics

Syntax ACOS(number)

Returns the arccosine of **number**; for example, the angle whose cosine is **number**. The input value must be in the range -1 <= number <= 1, or a #NUM! error is returned. The resulting angle is in the range 0 <= angle <= PI radians (0 <= angle <= 180 degrees).

Example

ACOS(0)

Equals 90 deg.

ASIN

Related Topics

Syntax ASIN(number)

Returns the arcsine of **number**; for example, the angle whose sine is **number**. The input value must be in the range -1 <= number <= 1, or a #NUM! error is returned. The resulting angle is in the range -PI/2 <= angle <= PI/2 radians (-90 <= angle <= 90 degrees).

Example

ASIN(1)

Equals 90 deg.

ATAN

Related Topics

Syntax ATAN(number)

Returns the arctangent of **number**; for example, the angle whose tangent is **number**. The resulting angle is in the range -PI/2 <= angle <= PI/2 radians (-90 <= angle <= 90 degrees).

Example

ATAN(1)

Equals 45 deg.

ATAN2

Related Topics

Syntax ATAN2(y,x)

Returns the arctangent of the point represented by **x** and **y**. The result is a number in the current unit of measure for angles.

The arctangent is the angle measured counterclockwise from the positive **x-**axis to a line that intersects the origin (0,0) and the point represented by **x** and **y**. In Visio, ATAN2(0,0) returns 0. To force the result of ATAN2 into a different angular measurement, use DEG or RAD.

The ATAN2 function is the antifunction of the TAN function. ATAN2 returns the angle whose angle is equal to **y** divided by **x**. If ATAN2(y,x) represents an angle in a right triangle, then **y** is the proverbial "opposite side" and **x** is the "adjacent side," so the function could be written as ATAN2(opposite,adjacent).

Examples

ATAN2(1.25,2.25)

Returns 29.0456 degrees.

ATAN2(1,SQRT(3))

Returns 30 degrees.

ATAN2(1,1)

Returns 45 degrees.

BITAND

Related Topics

Syntax BITAND(binarynumber1,binarynumber2)

Returns a binary number in which each bit is set to 1 only if the corresponding bit in both **binarynumber1** and **binarynumber2** is 1. Otherwise the bit is set to 0. This function can be used to test and change properties of a shape that are stored as bit masks, for example, the shape's text format.

Example

BITAND(12,6)

Returns 4. The 12 = 0...01100. The 6 = 0...00110. Therefore BITAND(12,6) = 0...00100.

BITNOT

Related Topics

Syntax BITNOT(binarynumber)

Returns a binary number in which each bit is set to 1 only if the corresponding bit in **binarynumber** is 0; otherwise the bit is set to 0.

Example

BITNOT(6)

Returns 65529. The 6 = 0...00110. Therefore BITNOT(6) = 1...11001.

BITOR

Related Topics

Syntax BITOR(binarynumber1,binarynumber2)

Returns a binary number in which each bit is set to 1 if the corresponding bit in either **binarynumber1** or **binarynumber2** is 1. The bit is set to 0 only if the corresponding bit is 0 in both **binarynumber1** and **binarynumber2**.

Example

BITOR(12,6)

Returns 14. The 12 = 0...01100. The 6 = 0...00110. Therefore BITOR(12,6) = 0...01110.

BITXOR

Related Topics

Syntax BITXOR(binarynumber1,binarynumber2)

Returns a binary number in which each bit is set to 1 if the corresponding bit in either but not both **binarynumber1** and **binarynumber2** is 1. Otherwise the bit is set to 0.

Example

BITXOR(12,6)

Returns 10. The 12 = 0...01100. The 6 = 0...00110. Therefore BITXOR(12,6) = 0...01010.

BLUE

Related Topics

Syntax BLUE(expression)

Returns the value of a color's blue component in **expression**. The return value is a number in the range 0 to 255, inclusive, or a cell reference that resolves to an index. If **expression** is invalid, returns 0 (black).

Example

BLUE(Sheet.4!FillForegnd)

Returns the value of the fill foreground color's blue component for the shape whose ID is Sheet.4.

BLUE(13)

Returns 128 if the document uses the default Visio color palette, where cyan is the color at index 13.

CALLTHIS

Related Topics

Syntax CALLTHIS("procedure",["project"],[arg1,arg2,…])

Calls a **procedure** in a VBA **project**.

Procedure is a required string that specifies the name of the **procedure** to call in the specified **project** or, if **project** is omitted, in the project of the document that contains the CALLTHIS formula being evaluated. In the VBA project, **procedure** is defined as follows:

```
procedure( shpObj As Visio.shape [arg1 As type, arg2 As type...] )
```

where *shpObj* is a reference to the Shape object that contains the CALLTHIS formula being evaluated, and **arg1**, **arg2**... are the arguments specified in that formula.

Notice that shpObj is very much like the "this" argument passed to a C++ member procedure; hence, the name "CALLTHIS." In effect, a cell that contains a formula with CALLTHIS is saying, "Call procedure and pass it a reference to my shape."

Project is an optional string that identifies the project containing **procedure**. If **project** is specified, Visio scans all open documents for the one containing **project** and calls **procedure** in that project. If **project** is omitted or null (""), **procedure** is assumed to be in the VBA project of the document that contains the CALLTHIS formula being evaluated.

Arg1, arg2… can be zero or more arguments of any type, including number, string, date, and currency values, that are passed as parameters to procedure. Numbers are passed in external units. For example, if you pass the value of the Height cell from a 3-cm-tall shape, 3 is passed. To pass different units with a number, use the FORMATEX function or explicitly coerce units by adding a null number-unit pair, for example, 0 ft + Height.

Notes

CALLTHIS always evaluates to 0, and the call to **procedure** occurs during idle time after the recalculation process finishes. **Procedure** can return a value, but Visio ignores it. **Procedure** can simulate returning a value that Visio can recognize by setting the formula or result of another cell in the document, but not the cell that called **procedure** unless you want to overwrite the CALLTHIS formula.

CALLTHIS differs from the RUNADDON function in that a document's project need not reference another project in order to call into that project.

Examples

CALLTHIS("p",,FORMATEX(Height,"#.00 u",,"cm"))

Calls the procedure named p and passes the value of Height in centimeters, such as 7.62 cm.

CALLTHIS("q",,0 cm.+Height,Sheet.2!Width)

Calls the procedure named q and passes the cell's height in centimeters and Sheet.2's width in external units.

CEILING

Related Topics

Syntax CEILING(number,multiple)

Rounds **number** away from 0 (zero) or to the next instance of **multiple**. If **multiple** is not specified, the number rounds away from 0 to the next integer.

Number and **multiple** must have the same signs, or a #NUM! error is returned. If either **number** or **multiple** cannot be converted to a value, a #VALUE! error is returned. If either **number** or **multiple** is 0, the result is 0.

Examples

CEILING(3.7)

Equals 4.

CEILING(-3.7)

Equals -4.

CEILING(3.7, 0.5)

Equals 4.

CHAR

Related Topics

Syntax CHAR(number)

Returns the ANSI character for **number**.

The resulting string is one character in length. The **number** must be an integer between 1 and 255 (inclusive), or the function returns an error.

Example

CHAR(9)

Returns the tab character.

COS

Related Topics

Syntax COS(angle)

Returns the cosine of **angle**. Uses radians unless a unit of measure is specified with **angle**.

Example

COS(45 degrees)

Returns 0.7071.

COSH

Related Topics

Syntax COSH(angle)

Returns the hyperbolic cosine of **angle**. Uses radians unless a unit of measure is specified with **angle**.

Example

COSH(45 degrees)

Returns 1.3246.

CY

Related Topics

Syntax CY(value[,currencyID|"abbreviation"])

Returns a currency value. The input **value** can be a number or a string that includes currency-specific formatting. If the optional argument is not specified, the currency value is formatted according to the currency style in the system's Regional Settings.

To specify a different currency, you must include a valid **currencyID** or **abbreviation** string. For a list, see Currency constants.

If **value** is incompatible with the designated currency type, or an invalid argument such as "not a number" is specified, a #VALUE! error is returned. If **value** is greater than 922,337,203,685,477.5807 or less than -922,337,203,685,477.5808, a #VALUE! error is returned.

For better precision with very large currency values that include fractions of a unit, such as 3.6 trillion, use string arguments for **value**.

Specifying an invalid **currencyID** or **abbreviation** returns an error.

Examples

If the user's Regional Setting specifies United States $:

CY(999998.993)

Returns $999,998.99

CY(12345678.993, "USD")

Returns $12,345,678.99

CY(12345678.993, "DEM")

Returns 12,345,678.99 DEM

CY(12345678.7832, "XXX")

Returns 12,345,678.78

Currency constants

To format a number as currency, you can use the CY function and pass an optional constant to specify which country's currency to use. The currency constants can be specified as the ID number that corresponds to a country or as a string (enclosed in quotation marks) that corresponds to an ISO 4217 3 abbreviation.

NOTE If you request to show currency symbols for nonlocal currencies (which you can by setting the appropriate flage in the Visio.ini file), and the system does not know the symbol for a given currency, Visio displays the universal currency symbol (¤). You can also use the ¤ symbol as input anywhere a currency symbol is accepted. For example, CY("-¤123,456.78").

The following are valid IDs and abbreviations.

ID	Abbreviation	Currency
0	SYS	Uses system settings
1	XXX	Formats as a number
2 - 10		Reserved
11	USD	United States dollar
12	ATS	Austrian schilling
13	AUD	Australian dollar
14	BEF	Belgian franc
15	CAD	Canadian dollar
16	CHF	Swiss franc
17	CNY	Chinese renminbi yuan
18	DEM	German mark
19	DKK	Danish krone
20	ESP	Spanish peseta
21	FIM	Finnish markka
22	FRF	French franc
23	GBP	British pound sterling
24	GRD	Greek drachma
25	HKD	Hong Kong dollar
26	HUF	Hungarian forinth
27	IDR	Indonesian rupiah
28	IEP	Irish punt
29	ILS	Israeli shekel
30	ITL	Italian lira
31	JPY	Japanese yen
32	KRW	South Korean won
33	LUF	Luxembourgian franc

34	MXN	Mexican peso (new)
35	MYR	Malaysian ringgit
36	NLG	Dutch guilder
37	NOK	Norwegian krone
38	NZD	New Zealand dollar
39	PHP	Philippine peso
40	PLZ	Polish zloty
41	PTE	Portuguese escudo
42	ROL	Romanian leu
43	RUR	Russian ruble
44	SEK	Swedish kroner
45	SGD	Singapore dollar
46	THB	Thai baht
47	TWD	Taiwan dollar
48	XEU	European currency unit
49	YUN	Yugoslavian dinar
50	ZAR	South African rand
51 - 55		Reserved
56	ARS	Argentinian peso
57	BMD	Bermudian dollar
58	BOB	Bolivian boliviano
59	BRR	Brazilian cruzeiro real
60	BSD	Bahamanian dollar
61	CLP	Chilean peso
62	COP	Colombian peso
63	CRC	Costa Rican colon
64	CZK	Czech koruna
65	DOP	Dominican peso
66	ECS	Ecuadorean sucre
67	EGP	Egyptian pound
68	HNL	Honduran lempira
69	INR	Indian rupee
70	JMD	Jamaican dollar
71	JOD	Jordanian dinar
72	KWD	Kuwaiti dinar
73	MOP	Macanese pataca

74	NIO	Nicaraguan cordoba oro
75	PAB	Panamanian balboa
76	PEN	Peruvian nuevo sol
77	PKR	Pakistani rupee
78	PYG	Paraguayan guarani
79	SAR	Saudi riyal
80	SIT	Slovenian tolar
81	SKK	Slovakian koruna
82	SVC	El Salvadorean colon
83	TRL	Turkish lira
84	TTD	Trinidad and Tobago dollar
85	UYU	Uruguayan peso uruguayo
86	VEB	Venezuelan bolivar
87	VND	Vietnamese dong

DATE

Related Topics

Syntax DATE(year,month,day)

Returns the date represented by **year**, **month**, and **day** formatted according to the short date style in the system's Regional Settings.

Example

DATE(1997,6,7)

Returns the value representing 6/7/97.

DATE(1997,6,7) + 4 ed.

Returns the value representing 6/11/97.

FORMAT(DATE(97,10,14),"C")

Returns the value representing Tuesday, October 14, 1997.

DATETIME

Related Topics

Syntax DATETIME("datetime"|expression[,lcid])

Returns the date and time value represented by **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

Notes

If **datetime** is missing or cannot be interpreted as a valid date or time, returns a #VALUE! error.

The returned value is formatted according to the short date style and time style set by the system's current Regional Settings.

DATETIME also accepts a single number value for **expression** where the integer portion of the result represents the number of days since December 31, 1899, and the decimal portion represents the fraction of a day since midnight.

Examples

DATETIME("May 30, 1997")+5 ed.

Returns the value representing 6/4/97.

FORMAT(DATETIME("5/20/97 14:30:45"),"C")

Returns the value representing Tuesday, May 20, 1997 2:30:45 PM

DATETIME("1:30 PM, July 19")

Returns the value representing 7/19/97 1:30 PM (assuming the current year is 1997).

DATETIME(35580.6337)

Returns the value representing 5/30/97 3:12:32 PM.

DATEVALUE

Related Topics

Syntax DATEVALUE("datetime"|expression[,lcid])

Returns the date value represented by **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. Any time component in **datetime** or **expression** is discarded.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

Notes

If **datetime** is missing or cannot be converted to a valid result, returns a #VALUE! error.

The returned value is displayed using the short date style set by the system's current Regional Settings.

DATEVALUE also accepts a single number value for **expression** where the integer portion of the result represents the days since December 31, 1899.

Examples

DATEVALUE(NOW( ))+5 ed.

Returns the date five days from now.

DATEVALUE("7/19/95 12:30")

Returns the date.

DATEVALUE("May 33, 1997")

Returns a #VALUE! error.

DATEVALUE(35580.6337)

Returns 5/30/97.

DAY

Related Topics

Syntax DAY("datetime"|expression[,lcid])

Returns an integer, 1 to 31, representing the day in **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. Any time component in **datetime** or **expression** is discarded.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

Notes

No rounding is done. If **datetime** is missing or cannot be converted to a valid result, returns an error.

DAY also accepts a single number value for **expression** where the integer portion of the result represents the number of days since December 31, 1899.

Example

DAY("May 30, 1997 3:45:24 PM")

Returns 30.

DAY(DATEVALUE("May 30, 1997")+7 ed.)

Returns 6.

DAY(35580.6337)

Returns 30.

DAYOFYEAR

Related Topics

Syntax DAYOFYEAR("datetime"|expression[,lcid])

Returns an integer, 1 to 366, representing the sequential day of the year in **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. Any time component in **datetime** or **expression** is discarded.

The result corresponds to January 1 to December 31. No rounding is done. If **datetime** is missing or cannot be interpreted as a valid date or time, returns an error.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

Notes

DAYOFYEAR also accepts a single number value for **expression** where the integer portion of the result represents the number of days since December 31, 1899.

Example

DAYOFYEAR("May 30, 1997 3:45:24 PM")

Returns 150.

DAYOFYEAR(DATEVALUE("May 30, 1997")+7 ed.)

Returns 157.

DAYOFYEAR(35580.6337)

Returns 150.

DEFAULTEVENT

Related Topics

Syntax DEFAULTEVENT( )

Performs the default event associated with the object.

Object	**Default event**
Shape	Edit text
Group	Open group window
OLE	Do primary verb
Other	Do nothing

In earlier versions of Visio, this function appears as _DEFAULTEVENT. Visio versions 4.0 and greater accept either style.

DEG

Related Topics

Syntax DEG(angle)

Converts the value of **angle** from radians to degrees.

Example

DEG(PI( )/4)

Returns 45.

DEPENDSON

Related Topics

Syntax DEPENDSON(cellref[,cellref2,...])

Creates a cell reference dependency. This function always returns FALSE. DEPENDSON has no effect when used in an Event row or an Action cell.

Example

OpenTextWin() + DependsOn(PinX,PinY)

Opens the text block for a shape whenever the shape is moved.

DOOLEVERB

Related Topics

Syntax DOOLEVERB("verb")

Executes a **verb** for the OLE object.

Example

DOOLEVERB("edit")

Runs the OLE object program and displays the linked or embedded object so that it can be edited.

In earlier versions of Visio, this function appears as _DOOLEVERB. Visio versions 4.0 and greater accept either style.

EVALTEXT

Related Topics

Syntax EVALTEXT(shapename!theText)

Evaluates the text in **shapename** as if it were a formula and returns the result. If there is no text, the result is zero. If the text cannot be evaluated, returns an error.

The theText cell refers to all of the text contained in a shape's text block. Shapename! can be used to refer to the text of a shape other than the current shape.

Example

EVALTEXT(Line.2!theText)

Evaluates the text contained in the shape Line.2. For example, if Line.2 contains "4 ft + 0.5 ft", returns the value 4.5 ft.

FLOOR

Related Topics

Syntax FLOOR(number,multiple)

Rounds **number** toward 0 (zero) or to the next instance of **multiple**. If **multiple** is not specified, the number rounds toward 0 to the next integer.

Number and **multiple** must have the same signs or a #NUM! error is returned. If either **number** or **multiple** cannot be converted to a value, a #VALUE! error is returned. If either **number** or **multiple** is 0, the result is 0.

Examples

FLOOR(3.7)

Equals 3.

FLOOR(-3.7)

Equals -3.

FLOOR(3.7, 0.5)

Equals 3.5.

FORMAT

Related Topics

Syntax FORMAT(expression,"formatpicture")

Returns the result of **expression** as a string formatted according to **formatpicture**. The type of the expression and the type specified in the format picture govern the behavior of the returned string. The **formatpicture** must be appropriate for the type of expression.

Returns an error if the result of **expression** and the type expected in **formatpicture** are of a different kind or if there are syntax errors in **formatpicture**.

Examples

FORMAT(1cm/4, "0.000 u")

Displays 0.250 cm.

FORMAT(1cm/4, "0.00 U")

Displays 0.25 CM.

FORMAT(1cm/4, "0.0 u")

Displays 0.3 cm.

Valid format pictures

Related Topics

The following table shows symbols you can use to format different types of values for display.

You can construct a format picture in the Format cell of the Custom Properties section and as an argument to the FORMAT or FORMATEX functions.

String and numeric values

#	Digit placeholder. Displays either a digit or nothing. Leading and trailing zeroes are not displayed. If more digits than placeholders are to the left of the decimal, all digits are displayed. If more digits than placeholders are to the right of the decimal, the fraction is rounded to the number of placeholders. For a dimension, if the placeholder is the leftmost digit, subunits that are 0 are not displayed. For example, FORMAT(0ft. 11.25in.,"#.##u") displays 11.25in.
0	Digit placeholder (zero). Displays either a digit or nothing. Leading and trailing zeroes are displayed. If more digits than placeholders are to the left of the decimal, all digits are displayed. If more digits than placeholders are to the right of the decimal, the fraction is rounded to the number of placeholders. For a dimension, subunits that are 0 are displayed. For example, FORMAT(2ft. 11.33in.,"0.## u") displays 2 ft. 11.33 in.
.	Decimal placeholder. Determines how many digits are displayed to the left and right of the decimal position. In a multipart unit, the decimal is used in the smallest (rightmost) subunit. Displays the decimal character defined for the system's Regional Settings. For example, FORMAT(250 cm.,"0.000 u") displays 250.000 cm.
,	Thousands separator. If surrounded by digit placeholders (# or 0), the separator separates thousands from hundreds within a number that has four or more digits to the left of the decimal. Displays the thousands separator defined for the system's Regional Settings.
E- E+ e- e+	Scientific format. If the format contains at least one digit placeholder to the right of these symbols, the number is displayed in scientific format. Inserts the E or e between the number and its exponent. For E+ or e+, displays the + sign before positive exponents and the - sign before negative exponents. For E- or e-, displays the - sign only when the exponent is negative. For example, FORMAT(12345.67,"###.#e+#") results in .123.5e+2.
u or **U**	Short label placeholder. Inserts abbreviated unit labels after each subunit. For example: in., ft., deg. The u placeholder inserts lowercase labels, and U inserts uppercase labels. Inserts the same number of spaces before the label as before the placeholder. For example, FORMAT(12ciceros 13 didots,"#u") displays 12c13.
uu or **UU**	Long label placeholder. Inserts unit labels after each subunit. For example: inches, feet, degrees. The u placeholder inserts lowercase labels, and U inserts uppercase labels. Inserts the same number of spaces before the label as before the placeholder. For example, FORMAT(12.43in.,"# #/4 UU") displays 12 2/4 INCHES.
uuu or **UUU**	Universal label placeholder. Inserts the universal (internal to Visio) form of unit labels after each subunit. The u placeholder inserts lowercase labels, and U inserts uppercase labels. Inserts the same number of spaces before the label as before the placeholder.
/	Fraction placeholder. Displays expression as a whole number with fraction if a leading digit placeholder is present. Otherwise, displays only the whole number in the numerator. If a number follows the digit placeholder in the denominator, rounds the fraction to the nearest fraction whose numerator is 1 and simplifies it. If a number is specified in the denominator without the digit placeholder, rounds to the nearest fraction

but does not simplify it. For example, FORMAT(12.43,"# #/4") displays 12 2/4.

space Displays a space character in the formatted output. To display another character, use the \ (backslash) character.

Currency values

$ Currency symbol. Displays the currency symbol defined in the system's Regional Settings.

u or **U** Short label placeholder. Inserts the standard symbol for local currency or the three-character currency constant for nonlocal currencies. For example, $99.00, 42.70 FRF. The u placeholder inserts lowercase or mixed-case labels, and U inserts uppercase labels.

uu or **UU** Long label placeholder. Inserts long currency labels after each subunit. For example: United States dollar, French franc. The u placeholder inserts lowercase or mixed-case labels, and U inserts uppercase labels.

uuu or UUU Universal label placeholder. Inserts the universal, three-character currency constant for all currencies after each subunit. For example, 99.00 USD, 42.70 FRF. The u placeholder inserts lowercase labels, and U inserts uppercase labels. Inserts the same number of spaces before the label as before the placeholder.

Text

**** Display the next character as is. To display the backslash character, type \\. See also "text".

"text" or **'text'** Displays the text enclosed in quotes as is. See also \ (backslash).

@ Text placeholder. Replaces a string if the value of expression is a string. For example, FORMAT("Hello", "`You entered (`@')'") results in "You entered (Hello)".

@+ Uppercase text placeholder. For string values, substitutes the input with uppercase. For example, FORMAT("Hello", "@ @+ @-") results in "Hello HELLO hello)".

@- Text placeholder. For string values, substitutes the input with lowercase. For example, FORMAT("Hello", "@ @+ @-") results in "Hello HELLO hello)".

Date values

c or **C** Date or time placeholder. Displays date and time values using a short (c) or long (C) date format, and the general time format. Visio versions 4.0 and lower ignore this placeholder. For example: FORMAT(DATETIME("6/25/97 12:05"),"C") displays Wednesday, June 25, 1997 12:05:00 PM. FORMAT(DATETIME("Jun. 25, 1997"),"c") displays 6/25/97.

/ Date separator. If the expression is a date, separates the date components. Displays the date separator defined for the system's Regional Settings.

[ ] Elapsed date placeholder. Used with the d, dd, w, and ww placeholders to display duration units. For example, [d] is elapsed days, [ww] is elapsed weeks.

d Day placeholder. Displays the day as a number (1-31) without a leading zero.

dd Day placeholder. Displays the day as a number (01-31) with a leading zero.

ddd or **w** Short day of week placeholder. Displays the day as an abbreviation (Sun-Sat).

dddd or **ww** Long day of week placeholder. Displays the day as a full name (Sunday-Saturday).

ddddd Short date placeholder. Displays a date in the short form defined for the system's

Regional Settings.

dddddd Long date placeholder. Displays a date in the long form defined for the system's Regional Settings.

M Month placeholder. Displays the month as a number (1-12) without a leading zero. See also m (minute placeholder).

MM Month placeholder. Displays the month as a number (01-12) with a leading zero. See also mm (minute placeholder).

MMM Month placeholder. Displays the month in abbreviated form (Jan-Dec).

MMMM Month placeholder. Displays the full name of the month (January-December).

yy Year placeholder. Displays the year as a two-digit number (00-99).

yyyy Year placeholder. Displays the year as a four-digit number (1900-2078).

Time values

: Time separator. Displays the time defined for the system's Regional Settings.

[ ] Elapsed time placeholder. Used with the h, hh, m, mm, s, and ss placeholders to display duration units. For example, [h] is elapsed hours, [ss] is elapsed seconds.

h Hour placeholder. Displays the hour without a leading zero in 12-hour form (0-12).

hh Hour placeholder. Displays the hour with a leading zero in 12-hour form (00-12).

H Hour placeholder. Displays the hour without a leading zero in 24-hour form (0-24).

HH Hour placeholder. Displays the hour with a leading zero in 24-hour form (00-24).

m Minute placeholder. Displays the minutes without a leading zero (0-59).

mm Minute placeholder. Displays the minutes with a leading zero (00-59).

s Seconds placeholder. Displays the seconds without a leading zero (0-59).

ss Seconds placeholder. Displays the seconds with a leading zero (00-59).

t AM/PM abbreviation. Displays the abbreviation defined for the system's Regional Settings.

tt AM/PM designator. Displays the full designator defined for the system's Regional Settings.

T General time format.

FORMATEX

Related Topics

Syntax FORMATEX(expression,"formatpicture"[,inputunit][,outputunit])

Returns the result of **expression** evaluated in **inputunit** as a string formatted according to formatpicture expressed in **outputunit**. The type of the expression and the type specified in the format picture govern the behavior of the returned string. The **formatpicture** must be appropriate for the type of expression.

Inputunit and **outputunit** are optional. They can be specified as a numerical value or as a valid abbreviated unit of measure (in, in., inch, etc. See Units of Measure.) **Inputunit** is used to scale untyped expression results, such as 3 + 4. If the result of **expession** has an explicit type, such as 3 ft + 8 ft, then **inputunit** is ignored.

Outputunit is used to specify the units used for the formatted result. If **outputunit** is not specified, the units for the result of the expression are used.

Returns an error if the result of **expression** and the type expected in **formatpicture** are of a different kind, if there are syntax errors in **formatpicture**, or if it does not recognize the units passed as **inputunits** or **outputunits**.

Example

FORMATEX(5.5, "0.00 u", "cm", "ft")

Returns 0.18 ft.

GOTOPAGE

Related Topics

Syntax GOTOPAGE("pagename")

Displays **pagename** in the currently active window. If a window is already displaying **pagename**, that window becomes active. If **pagename** does not exist, or if Visio is acting as an in-place server, GOTOPAGE has no effect.

In earlier versions of Visio, this function appears as _GOTOPAGE. Visio versions 4.0 and greater accept either style.

GRAVITY

Related Topics

Syntax GRAVITY(angle,limit1,limit2)

Calculates the text block's correct angle of rotation for the indicated shape rotation—it prevents the text from turning upside-down. GRAVITY is usually used in the TxtAngle cell. The value returned is 180 degrees if **angle** is between the angles specified by **limit1** and **limit2**; otherwise the value returned is 0 degrees. The arguments **limit1** and **limit2** are optional, and the default limits are 90 and 270 degrees.

All of the arguments are automatically normalized between 0 and 360 degrees by the function. If an argument does not specify units, radians are assumed.

Examples

GRAVITY(Angle)

Returns 180 degrees if Angle is between 90 and 270 degrees; otherwise, returns 0 degrees.

GRAVITY(2)

Returns 180 degrees, because 2 radians is between 90 and 270 degrees.

GRAVITY(100 deg., 110 deg., 290 deg.)

Returns 0 deg.

GRAVITY(100 deg., 290 deg., 110 deg.)

Returns 0 deg.

GREEN

Related Topics

Syntax GREEN(expression)

Returns the value of a color's green component in **expression**. The return value is a number in the range 0 to 255, inclusive, or a cell reference that resolves to an index. If **expression** is invalid, returns 0 (black).

Examples

GREEN(Sheet.4!FillForegnd)

Returns the value of the fill foreground color's green component for the shape whose ID is Sheet.4.

GREEN(11)

Returns 128 if the document uses the default Visio color palette, where dark yellow is the color at index 11.

GUARD

Related Topics

Syntax GUARD(expression)

Protects expression from deletion and change by actions performed in the drawing window, for example, moving, sizing, grouping, or ungrouping shapes. The cells most often affected are Width, Height, PinX, and PinY.

Guarding a formula in any cell prevents that cell's value from being changed by actions in the drawing window. However, one action in the drawing window can affect several cells, and each of these cells must be guarded if you want to prevent unexpected changes to the shape's appearance. This function cannot protect an expression in a formatting cell such as Char.Size from change when a user applies a style.

Example

GUARD(TEXTWIDTH(theText) + 0.5 in)

This expression in the Width cell of a shape's Geometry section prevents the expression (TEXTWIDTH(theText) + 0.5 in) from being replaced with another value when the shape's width is changed in the drawing window. TheText is a Visio page formula that refers to all of the text contained in a shape's text block.

HELP

Syntax HELP("filename.hlp!keyword")

HELP("filename.hlp!#number")

Opens the help file **filename.hlp** and displays the Search dialog box with the selected **keyword**. Or displays the popup help topic associated with **number**, a numeric ID referenced in the MAP section of the help project file (.HPJ). For information on developing Microsoft Windows online help, see the Microsoft Windows Software Development Kit (SDK) documentation.

If no **keyword** or ID is specified, opens the contents page of the help file.

Example

HELP("visio.hlp!shapesheet")

Opens the file VISIO.HLP and displays the topic whose keyword is shapesheet.

HELP("shape.hlp!#9000")

Opens the file SHAPE.HLP and displays the topic associated with the ID 9000.

HOUR

Related Topics

Syntax HOUR("datetime"|expression[,lcid])

Returns an integer, 0 to 23, representing the hour of the day of **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. The date component in **datetime** and **expression** is discarded.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

No rounding is done. If the **datetime** is missing or cannot be converted to a valid result, returns an error.

The returned value is formatted according to the time style set by the system's current Regional Settings.

HOUR also accepts a single number value for **expression** where the decimal portion of the result represents the fraction of a day since midnight.

Example

HOUR("3:45 PM")

Returns 15.

HOUR("May 30, 1997 3:45:24 PM")

Returns 15.

HOUR(0.5)

Returns 12.

HOUR("5/30/97")

Returns 0.

HSL

Related Topics

Syntax HSL(hue,saturation,luminosity)

Returns a value representing an index in the document's color palette. Specifies a color by its **hue**, **saturation**, and **luminosity** components, where **hue** is a number in the range 0 to 239, inclusive, and **saturation** and **luminosity** are in the range 0 to 240, inclusive. Or specifies an expression that evaluates to such a number.

If the color returned by the function does not already exist in the current document's color palette, it is added to the document's list of available colors.

Examples

HSL(160,240,120)

Returns the index for the color blue.

HSL(HUE(FillForegnd),SAT(FillForegnd),LUM(FillForegnd)+100)

Returns the index for the fill foreground color with increased saturation.

The following table lists some standard colors and their hue, saturation, and luminosity values:

Color	Hue Value	Saturation Value	Luminosity Value
Black	160	0	0
Blue	160	240	120
Green	80	220	120
Cyan	120	240	120
Red	0	240	120
Magenta	200	240	120
Yellow	40	240	120
White	160	0	240

HUE

Related Topics

Syntax HUE(expression)

Returns the value of a color's hue component in **expression**. The return value is a number in the range 0 to 239, inclusive, or a cell reference that resolves to an index. If **expression** is invalid, returns 0 (black).

Examples

HUE(Sheet.4!FillForegnd)

Returns the value of the fill foreground color's hue component for the shape whose ID is Sheet.4.

HUE(7)

Returns 120 if the document uses the default Visio color palette, where cyan is the color at index 7.

HYPERLINK

Related Topics

Syntax HYPERLINK("address"[,"subaddress","extrainfo",window,"frame"])

Navigates to the specified **address**, which can be a DOS, UNC, or URL path.

Address can be a full path or a relative path based on the Hyperlink Base specified in the Properties dialog box. If the document has no base path, Visio navigates according to the document path. If the document hasn't been saved, the hyperlink is undefined.

Subaddress is an optional string that specifies a location within **address** to link to. For example, if **address** is a Visio file, **subaddress** can be a page name; if an Excel file, **subaddress** can be a worksheet or range within a worksheet; if a URL for an HTML page, **subaddress** can be an anchor.

Extrainfo is an optional string that passes information used in resolving the URL, such as the coordinates of an image map.

Window is an optional argument that specifies whether the hyperlink is opened in a new window. The default is FALSE.

Frame is an optional string that specifies the name of a frame to target when Visio is open as an ActiveX document in an ActiveX browser, such as Internet Explorer 3.0 or later. The default is an empty string.

Example

HYPERLINK("C:\My Documents\Drawing1.VSD")

HYPERLINK("\\Server\Share\Drawing1.VSD")

HYPERLINK("http://www.visio.com")

HYPERLINK("..\data.xls","sheet1!A1")

IF

Related Topics

Syntax IF(logicalexpression,valueiftrue,valueiffalse)

Returns valueiftrue if logicalexpression is true. Otherwise, returns valueiffalse.

Example

IF(Height > 1.25 in.,5,7)

Returns 5 if the shape's height is greater than 1.25 inches. Returns 7 if the shape's height is less than or equal to 1.25 inches.

INDEX

J Related Topics

Syntax INDEX(index,"list"[,[delimiter][,[errorvalue]]])

Returns the substring at the zero-based location **index** in the **list** delimited by **delimiter**. Or returns -1 if not found.

Delimiters can be more than one character in length and include multibyte characters. The default is a semicolon. If the list begins or ends with a delimiter, a null string is assumed to exist before or after the list. Consecutive delimiters imply a null string in between.

If the index is out of range, Visio returns an empty string or the optional token provided as the **errorvalue** argument.

Examples

INDEX(3,"cat;rat;;goat")

Returns "goat".

INDEX(4,";1;2;3;",,"ERROR")

Returns ERROR.

INTERSECTX

Related Topics

Syntax INTERSECTX(x1,y1,angle1,x2,y2,angle2)

Returns the x-coordinate (in the local coordinate system) of the point where two lines intersect. Each line is defined as a point (x,y) and an angle. Visio uses this function in the PinX cell of a shape glued to a rotated guide.

X1 is the x-coordinate of a point on the first line.

Y1 is the y-coordinate of a point on the first line.

Angle1 is the value of the Angle cell for the first line.

X2 is the x-coordinate of a point on the second line.

Y2 is the y-coordinate of a point on the second line.

Angle2 is the value of the Angle cell for the second line.

Notes

If the lines don't intersect, returns a DIV BY 0 error, which Visio ignores, restoring the last known value for the point.

Example

INTERSECTX(VertGuide!PinX,VertGuide!PinY,VertGuide!Angle,
HorzGuide!PinX,HorzGuide!PinY,HorzGuide!Angle)

Returns the x-coordinate of the intersection point of VertGuide and HorzGuide in page units.

INTERSECTY

Related Topics

Syntax INTERSECTY(pinx1,piny1,angle1,pinx2,piny2,angle2)

Returns the y-coordinate (in the local coordinate system) of the point where two lines intersect. Each line is defined as a point (x,y) and an angle. Visio uses this function in the PinY cell of a shape glued to a rotated guide.

X1 is the x-coordinate of a point on the first line.

Y1 is the y-coordinate of a point on the first line.

Angle1 is the value of the Angle cell for the first line.

X2 is the x-coordinate of a point on the second line.

Y2 is the y-coordinate of a point on the second line.

Angle2 is the value of the Angle cell for the second line.

Notes

If the lines don't intersect, returns a DIV BY 0 error, which Visio ignores, restoring the last known value for the point.

Example

INTERSECTY(VertGuide!PinX,VertGuide!PinY,VertGuide!Angle, HorzGuide!PinX,HorzGuide!PinY,HorzGuide!Angle)

Returns the y-coordinate of the intersection point of VertGuide and HorzGuide in page units.

INTUP

J Related Topics

Syntax INTUP(number)

Rounds **number** up to the next integer.

Examples

INTUP(3.2)

Equals 4.

INTUP(-3.2)

Equals -3.

INTUP(3)

Equals 3.

INT

Related Topics

Syntax INT(number)

Rounds **number** down to the next integer.

Examples

INT(7.2)

Returns 7.

INT(-7.2)

Returns -8.

ISERR

J Related Topics

Syntax ISERR(cellref)

Returns TRUE if the value of **cellref** is any error type except #N/A; otherwise returns FALSE.

The ISERR function is used in formulas that refer to another cell.

Example 1

Cell	Formula	Value Returned
Scratch.A1	=NA( )	#N/A!
Scratch.B1	=ISERR(Scratch.A1)	FALSE

Returns FALSE because the #N/A! error is not recognized by the ISERR function. Use ISERROR to find all error types.

Example 2

Cell	Formula	Value Returned
Scratch.X1	="House"	#VALUE!
Scratch.A1	=ISERR(Scratch.X1)	TRUE

Returns TRUE because the #VALUE! error is recognized by the ISERR function.

ISERRNA

Related Topics

Syntax ISERRNA(cellref)

Returns TRUE if the value of **cellref** is error type #N/A! (not available); otherwise returns FALSE.

The ISERRNA function is used in formulas that refer to another cell.

Example

Cell	Formula	Value Returned
Scratch.A1	="5 + 3"	"8"
Scratch.B1	=ISERRNA(Scratch.A1)	FALSE

Returns FALSE because the value returned is available.

Cell	Formula	Value Returned
Scratch.A1	=NA( )	#N/A!
Scratch.B1	=ISERRNA(Scratch.A1)	TRUE

Returns TRUE because the value returned is error type #N/A!

ISERROR

Related Topics

Syntax ISERROR(cellref)

Returns TRUE if the value of **cellref** is any error type; otherwise returns FALSE.

The ISERROR function is used in formulas that refer to another cell.

Example 1

Cell	Formula	Value Returned
Scratch.A1	=NA( )	#N/A!
Scratch.B1	=ISERROR(Scratch.A1)	TRUE

Returns TRUE because the #N/A! error is recognized by the ISERROR function. You can use ISERR to find all types but the #N/A! error.

Example 2

Cell	Formula	Value Returned
Scratch.X1	="House"	#VALUE!
Scratch.B1	=ISERR(Scratch.X1)	TRUE

Returns TRUE because the #VALUE! error is recognized by the ISERROR function. To build an expression based on the #VALUE! error, use the ISERRVALUE function.

ISERRVALUE

J Related Topics

Syntax ISERRVALUE(cellref)

Returns TRUE if the value of **cellref** is error type #VALUE, where an argument in the formula is the wrong type.

The ISERRVALUE function is used in logical expressions that refer to another cell.

Example 1

Cell	Formula	Value Returned
Scratch.X1	="House"	#VALUE!
Scratch.A1	=If (ISERRVALUE(Scratch.X1),2,Scratch.X1)	2

Note: Scratch cells A through D won't return a #VALUE! error because the formula can contain numbers and letters in the same string. Cells X and Y must contain numbers only.

Returns 2 because the value returned is a #VALUE! error, and the expression instructs Visio to return a 2 in place of the error.

Example 2

Cell	Formula	Value Returned
Scratch.A1	="5 + 7"	5 + 7
Scratch.B1	=If (ISERRVALUE(Scratch.A1),2,Scratch.A1)	5 + 7

Returns 12 because the value returned is not a #VALUE! error, and the expression instructs Visio to return the value of the original cell.

LN

Related Topics

Syntax LN(number)

Returns the natural logarithm of **number**. **Number** must be positive, or LN returns the error value #NUM.

Example

LN(42)

Returns 3.7377.

LOC

Related Topics

Syntax LOC(point)

Returns the **x,y** coordinates of point measured from the lower-left corner of the shape's selection rectangle. In Visio, a point is a single value that embodies a pair of x- and y-coordinates.

Example

LOC(PNT(Sheet.5!LocPinX, Sheet.5!LocPinY))

In this expression, PNT converts a set of coordinates in Sheet.5 to a point. (Sheet.5 is another shape on the same drawing page.) LOC then converts the point to a set of coordinates in relation to the lower-left corner of the selection rectangle of the current shape.

The 5 in Sheet.5 is the ID number for the shape, which is displayed in the Special dialog box.

LOG10

Related Topics

Syntax LOG10(number)

Returns the base 10 logarithm of **number**. **Number** must be positive or LOG10 returns the error value #NUM.

Example

LOG10(42)

Returns 1.6232.

LOOKUP

Related Topics

Syntax LOOKUP("key","list"[,"delimiter"])

Returns a zero-based index that indicates the location of the substring **key** in a **list**, or returns -1 if the target string contains the **delimiter**. Uses a case-insensitive search. If the list begins or ends with a delimiter, a null string is assumed to exist before or after the list. Consecutive delimiters imply a null string in between.

Delimiters may be more than one character in length and may include multibyte characters. The default delimiter is the semicolon. All the arguments must be strings or able to be converted to strings. If they are not, an empty string is substituted for the offending argument.

Examples

LOOKUP("rat","cat;rat;;goat")

Returns 1.

LOOKUP("",";cat;rat;;goat")

Returns 0.

LOOKUP("t","cat;rat;;goat","a")

Returns 3.

LOTUSNOTES

Related Topics

Syntax LOTUSNOTES("field")

Returns the most recent information read from Lotus Notes.

If you create a field in Lotus Notes, you need to create a corresponding field in the Lotus Notes Fields dialog box. In order for Visio to read this value, you need to define the field in a Lotus Notes document and check Read From Lotus Notes in the Lotus Notes Fields dialog box.

Example

LOTUSNOTES ("BoxWidth") returns the value of the Lotus Notes field "BoxWidth"

Returns 12 inches.

LOWER

Related Topics

Syntax LOWER("string")

Returns **string** in lowercase. **String** can be a string, a cell reference, or an expression the result of which is converted to a string.

The case conversion is locale specific, based on the current user (for Microsoft Windows NT) or system (for Microsoft Windows 95) settings.

Example

LOWER("mIxEd CAse")

Returns "mixed case".

LUM

Related Topics

Syntax LUM(expression)

Returns the value of a color's luminosity component in **expression**. The return value is a number in the range 0 to 240, inclusive, or a cell reference that resolves to an index. If **expression** is invalid, returns 0 (black).

Examples

LUM(Sheet.4!FillForegnd)

Returns the value of the fill foreground color's luminosity component for the shape whose ID is Sheet.4.

LUM(6)

Returns 120 if the document uses the default Visio color palette, where magenta is the color at index 6.

MAGNITUDE

Related Topics

Syntax MAGNITUDE(constantA,A,constantB,B)

Returns the magnitude of the vector whose rise is **A** and whose run is **B**, multiplied by the respective constants **constantA** and **constantB**. MAGNITUDE is calculated according to the following formula:

SQRT((constantA * A)^2 + (constantB * B)^2)

Example

MAGNITUDE(1, 3, 1, 4)

Returns 5.

MAX

Related Topics

Syntax MAX(number1,number2,...,number14)

Returns the largest **number** from a list of up to 14 numbers.

Example

MAX(13 in.,1 ft, 20 cm)

Returns 13 in.

MIN

Related Topics

Syntax MIN(number1,number2,...,number14)

Returns the smallest **number** from a list of up to 14 numbers.

Example

MIN(13 in.,1 ft, 20 cm)

Returns 20 cm.

MINUTE

Related Topics

Syntax MINUTE("datetime"|expression[,lcid])

Returns an integer, 0 to 59, representing the minutes component of **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. The date component in **datetime** and **expression** is discarded.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

No rounding is done. If **datetime** is missing or cannot be converted to a valid result, returns an error.

The returned value is formatted according to the time style set by the system's current Regional Settings.

MINUTE also accepts a single number value for **expression** where the decimal portion represents the fraction of a day since midnight.

Example

MINUTE("7/7/97 3:45:24 PM")

Returns 45.

MINUTE(TIMEVALUE("Jan. 25, 1997 12:07:45")+6 em.)

Returns 13.

MINUTE(0.575)

Returns 48.

MODULUS

Related Topics

Syntax MODULUS(number,divisor)

Returns the remainder (modulus) resulting when **number** is divided by **divisor**. The result has the same sign as **divisor**. A #DIV/0! error is returned if **divisor** is 0.

In almost all situations, the MODULUS function should be used rather than the MOD function.

Examples

MODULUS(5, 1.4)

Returns 0.8.

MODULUS(5, -1.4)

Returns -0.6.

MODULUS(-5, 1.4)

Returns 0.6.

MODULUS(-5, -1.4)

Returns -0.8.

MOD

Related Topics

Syntax MOD(number,divisor)

Returns the remainder that results when **number** is divided by **divisor**. The MOD function supplied with earlier versions of Visio is supported for backward compatibility. The MODULUS function should be used in new formulas.

MONTH

Related Topics

Syntax MONTH("datetime"|expression[,lcid])

Returns an integer, 1 to 12, representing the month in **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. The time component of **datetime** or **expression** is discarded.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

No rounding is done. If the input string is missing or cannot be converted to a valid result, returns an error.

The returned value is formatted according to the short date style set by the system's current Regional Settings.

MONTH also accepts a single number value for **expression** where the integer portion of the result represents the number of days since December 31, 1899.

Example

MONTH("May 30, 1997 3:45:24 PM")

Returns 5.

MONTH(DATEVALUE("May 30, 1996")+7 ed.)

Returns 6.

MONTH(35580.6337)

Returns 5.

NA

Related Topics

Syntax NA( )

Returns the error value #NA!. NA is the error value that means "no value available." Use the NA function in a ShapeSheet cell where information is missing, so that the cell's default value is not used in calculations.

NOT

Related Topics

Syntax NOT(logicalexpression)

Returns 1 if **logicalexpression** is false. Otherwise, NOT returns 0.

Example

NOT(Height > 0.75 in.)

Returns 1 if Height is less than or equal to 0.75 inches. Returns 0 if Height is greater than 0.75 inches.

NOW

J Related Topics

Syntax NOW( )

Returns today's date and time formatted according to the short date style and time style in the system's Regional Settings.

Example

NOW( )

Returns the current date and time, such as 9/27/96 12:03:30 PM.

FORMAT(NOW(),"dd MMM., yyyy hh:mm")

Returns the current date and time formatted as 27 Sep., 1996 12:03.

NOW()+2EW.

Returns the current date and time plus two elapsed weeks, such as 10/11/96 12:03:30 PM.

OPENFILE

Related Topics

Syntax OPENFILE("filename")

Opens the specified Visio document if not already open. Transfers focus to and activates the target document window at the completion of the command.

Multiple OpenFile requests are queued and executed in order of evaluation at the completion of a command, with the last named document receiving final focus. If the current Visio document is activated for visual (in-place) editing, a new instance of Visio is launched with the requested file name. This function always returns FALSE.

In earlier versions of Visio, this function appears as _OPENFILE. Visio versions 4.0 and greater accept either style.

Example

OPENFILE("basic.vss")

Opens the specified stencil file BASIC.VSS in a new window and gives that window the focus.

OPENGROUPWIN

J Related Topics

Syntax OPENGROUPWIN( )

Opens the group in the group window. If the object is not a group, OPENGROUPWIN does nothing.

In earlier versions of Visio, this function appears as _OPENGROUPWIN. Visio versions 4.0 and greater accept either style.

OPENPAGE

Related Topics

Syntax OPENPAGE("pagename")

Displays **pagename** in new window. If a window is already displaying **pagename**, that window becomes active. If the **pagename** does not exist, or if Visio is acting as an in-place server, OPENPAGE has no effect.

In earlier versions of Visio, this function appears as _OPENPAGE. Visio versions 4.0 and greater accept either style.

OPENSHEETWIN

J Related Topics

Syntax OPENSHEETWIN( )

Opens the ShapeSheet window in a new window.

In earlier versions of Visio, this function appears as _OPENSHEETWIN. Visio versions 4.0 and greater accept either style.

OPENTEXTWIN

Related Topics

Syntax OPENTEXTWIN( )

Opens the shape's text block so that the text can be edited.

In earlier versions of Visio, this function appears as _OPENTEXTWIN. Visio versions 4.0 and greater accept either style.

OR

Related Topics

Syntax OR(logicalexpression1,logicalexpression2,...,logicalexpression14)

Returns 1 if any of the **logical** expressions is true. Any expression that evaluates to a non-zero value is considered true. If all of the logical expressions are false or equal 0, OR returns 0.

Example

OR(Height > 1,PinX > 1)

Returns 1 if either expression is true. Returns 0 only if both expressions are false.

PAR

Related Topics

Syntax PAR(point)

Returns the **x,y** coordinates of **point** in the coordinate system of the shape's parent. In Visio, a point is a single value that embodies a pair of **x**- and **y**-coordinates. If the shape is in a group, its parent is the group. If the shape is not in a group, its parent is the page the shape is on.

Example

PAR(PNT(PinX,PinY)

In this expression, PNT converts a pair of coordinates in the current shape into a point. PAR then converts the point into a pair of coordinates in relation to the lower-left corner of the page that contains the current shape.

PI

Related Topics

Syntax PI( )

Returns the mathematical constant **pi**, which is 3.1415926535898.

Example

(14 in. / 2)^2 * PI( )

Returns 153.9380 in., the area of a circle whose diameter is 14 inches.

PLAYSOUND

Related Topics

Syntax PLAYSOUND("filename"|"alias",isAlias,beep,synch)

On systems with a sound card, plays the sound recorded in **filename**, or plays the system sound identified by **alias** if **isAlias** is nonzero. If the sound cannot be played, Visio beeps if **beep** is a nonzero number.

Sounds should usually be played asynchronously (**synch**=0) so Visio can continue processing while the sound is being played. To string several sounds together, play them synchronously (**synch** = nonzero number), or some might fail to play.

Examples

PLAYSOUND("chord.wav", 0, 0, 0)

Plays the wave audio file chord.wav asynchronously with no warning beep.

PLAYSOUND("SystemExclamation", 1, 0, 0)

Plays the system exclamation sound asynchronously with no warning beep.

PNT

Related Topics

Syntax PNT(x,y)

Returns the point represented by the coordinates **x** and **y** as a single value. Converting coordinates to points allows you to change a shape's geometry without having to manipulate x- and y-coordinates separately.

Example

PNT(PinX,PinY)

Returns the point represented by PinX and PinY.

PNTX

Related Topics

Syntax PNTX(point)

Returns the **x-**coordinate of point measured from the lower-left corner of the shape's selection rectangle.

Example

PNTX(PNT(7,12))

Returns 7.

PNTY

Related Topics

Syntax PNTY(point)

Returns the **y-**coordinate of point measured from the lower-left corner of the shape's selection rectangle.

Example

PNTY(PNT(7,12))

Returns 12.

POW

Related Topics

Syntax POW(number,exponent)

Returns **number** raised to the power of **exponent**. Both **number** and **exponent** may be non-integers, and they may be negative. If **number** is not 0 and **exponent** is 0, POW returns 1. If **number** is 0 and **exponent** is negative, POW returns 0.0. If both **number** and **exponent** are 0, or if **number** is negative and **exponent** is not an integer, POW returns 0.0.

Example

POW(5,2)

Returns 25.

RAD

Related Topics

Syntax RAD(angle)

Converts the value of **angle** from degrees to radians.

Example

RAD(45)

Returns 0.7854.

RAND

Related Topics

Syntax RAND( )

Returns a random number from 0 to 1. Inclusive, with 15 digits of precision. Returns a different number each time the function is evaluated, which is once per minute according to the system clock. Use RAND to create animation effects by setting shapes' properties to random values.

Example

RAND( )

Returns a decimal fraction such as 0.3503.

RECTSEKT

Related Topics

Syntax	RECTSECT(width,height,x,y,option)

Calculates the sector of a rectangle associated with **x** and **y** and returns an integer 0 to 4, indicating the sector.

Consider a rectangle that has a **width** and a **height**, and a point (**x,y**) from the center point of the rectangle. Draw diagonal lines through the corners of the rectangle to divide it into four sectors and a center point. The sectors are:

0 = the center point of the rectangle

1 = the sector containing the right side of the rectangle

2 = the sector containing the top side of the rectangle

3 = the sector containing the left side of the rectangle

4 = the sector containing the bottom side of the rectangle

Option is used to specify how points that fall on the diagonals are treated.

Option	**Treatment**
0	Use the left and right sectors for points on a diagonal
1	Use the top and bottom sectors for points on a diagonal

Example

RECTSECT(1 in., 2 in., 1 in., -7 in., 0)

Returns 4.

RED

Related Topics

Syntax RED(expression)

Returns the value of a color's red component in **expression**. The return value is a number in the range 0 to 255, inclusive, or a cell reference that resolves to an index. If **expression** is invalid, returns 0 (black).

Example

RED(22)

Returns 51 if the document uses the default Visio color palette, where dark gray is the color at index 22.

RED(Char.Color)

Returns the value of the red component of the current font color.

REF

Related Topics

Syntax REF( )

Returns the error value #REF!.

Example

REF( )

Returns #REF!

RGB

Related Topics

Syntax RGB(red,green,blue)

Returns a value representing an index in the document's color palette. Specifies a color by its **red**, **green**, and **blue** components, where each is a number in the range 0 to 255, inclusive, or an expression that evaluates to such a number.

If the color returned by the function does not already exist in the current document's color palette, it is added to the palette.

Examples

RGB(0,0,255)

Returns the index for the color blue.

RGB(RED(Sheet.1!FillForegnd),120,0))

Returns the index for a color based on the fill foreground color of the shape whose ID is Sheet.1.

The following table lists some standard colors and their red, green, and blue values:

Color	Red Value	Green Value	Blue Value
Black	0	0	0
Blue	0	0	255
Green	0	255	0
Cyan	0	255	255
Red	255	0	0
Magenta	255	0	255
Yellow	255	255	0
White	255	255	255

ROUND

Related Topics

Syntax ROUND(number,numberofdigits)

Rounds **number** to the precision represented by **numberofdigits**. If **numberofdigits** is greater than 0, **number** is rounded by **numberofdigits** to the right of the decimal. If **numberofdigits** is 0, **number** is rounded to the integer. If **numberofdigits** is less than 0, **number** is rounded by **numberofdigits** to the left of the decimal.

Examples

ROUND(123.654,2)

Returns 123.65.

ROUND(123.654,0)

Returns 124.

ROUND(123.654,-1)

Returns 120.

RUNADDON

Related Topics

Syntax RUNADDON("string")

Passes **string** as code to be executed to the VBA project of the document containing the operand. If the project cannot execute **string**, Visio runs the add-on named **string**. If no add-on named **string** can be found, Visio does nothing and reports no error.

String can be any code that VBA recognizes, such as the name of a macro, a procedure with arguments, or a command that the VBA Immediate window can execute, or **string** can be the name of an add-on.

In earlier versions of Visio, this function appears as _RUNADDON. Visio 4.0 and later accepts either style.

Examples

RUNADDON("summation 1 2 3")

Calls the VBA procedure named summation and passes three arguments.

RUNADDON("debug.print ""xyz""")

Prints "xyz" in the VBA Immediate window.

RUNADDON("ThisDocument.Save")

Saves the document containing the function.

RUNADDON("calendar.exe")

Launches the CALENDAR.EXE add-on.

RUNADDONWARGS

Related Topics

Syntax RUNADDONWARGS("filename","arguments")

Runs **filename** and passes the command line **arguments** to the program as a string. In practice, **argument** should be 50 characters or fewer. Use the RUNADDONWARGS function to bind a program, such as an add-on, to an Actions or Events cell.

In earlier versions of Visio, this function appears as _RUNADDONWARGS. Visio versions 4.0 and greater accept either style.

Example

RUNADDONWARGS("GRAPHMKR.EXE","/GraphMaker=Stack")

Launches the add-on GRAPHMKR.EXE and passes it the argument /GraphMaker=Stack.

SAT

Related Topics

Syntax SAT(expression)

Returns the value of a color's saturation component in **expression**. The return value is a number in the range 0 to 240, inclusive, or a cell reference that resolves to an index. If **expression** is invalid, returns 0 (black).

Examples

SAT(Sheet.4!FillForegnd)

Returns the value of the fill foreground color's saturation component for the shape whose ID is Sheet.4.

SAT(8)

Returns 240 if the document uses the default Visio color palette, where dark red is the color at index 8.

SECOND

Related Topics

Syntax SECOND("datetime"|expression[,lcid])

Returns an integer, 0 to 59, representing the seconds component of **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. The date component in **datetime** or **expression** is discarded.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

No rounding is done. If **datetime** is missing or cannot be converted to a valid result, returns an error.

SECOND also accepts a single number value for **expression** where the decimal portion of the result represents the fraction of a day since midnight.

Example

SECOND("05/30/97 3:45:32 PM")

Returns 32.

SECOND(TIMEVALUE("May 30, 1996 12:07:45")+7 es.)

Returns 52.

SECOND(0.6337)

Returns 32.

SETF

Related Topics

Syntax SETF("cell",formula)

SETF("cell","formula")

SETF("cell","""formula""")

When evaluated, the result of the expression in **formula** becomes the new formula in **cell**. If formula is enclosed in quotation marks, the quoted expression is written to **cell**. To set **cell** to a string, enclose **formula** in three sets of quotation marks.

If **cell** is not a string, the function returns an error, and no cell's formula is changed. If **formula** contains a syntax error, the function returns an error, and the formula in **cell** is not changed.

Examples

SETF("Scratch.A1", 1.5in.*6+1ft.)

Sets the formula of Scratch.A1 to 21 in.

SETF("Scratch.A1", "1.5in.*6+1ft.")

Sets the formula of Scratch. A1 to the expression 1.5 in.*6+1ft.

SETF("Scratch.A1", """Say """"ahh""""""")

Sets the formula of Scratch.A1 to the string "Say ""ahh""" which evaluates to Say "ahh".

SIGN

Related Topics

Syntax SIGN(number,fuzz)

Returns 1, 0, or -1 depending on if **number** is positive, zero, or negative. **Fuzz** is an optional argument that specifies how close to zero **number** must be in order to be considered equal to zero. **Fuzz** helps avoid floating-point roundoff errors when a calculation is almost zero. If you do not specify a fuzz value, Visio uses 1E-9 (0.000000001). You may want to supply a different value when you scale drawings or when you want an exact comparison.

Examples

SIGN(-5)

Equals -1.

SIGN(0)

Equals 0.

SIGN(0.00000000001)

Equals 0.

SIGN(0.00000000001,0)

Equals 1.

SIN

Related Topics

Syntax SIN(angle)

Returns the sine of **angle.** Uses radians unless a unit of measure is specified with **angle**.

Example

SIN(45 degrees)

Returns 0.7071.

SINH

Related Topics

Syntax SINH(angle)

Returns the hyperbolic sine of **angle.** Uses radians unless a unit of measure is specified with **angle**.

Example

SINH(45 degrees)

Returns 0.8687.

SQRT

Related Topics

Syntax SQRT(number)

Returns the square root of **number**. If **number** is negative, SQRT returns the error value #NUM!.

Example

SQRT(2)

Returns 1.4142.

SUM

J Related Topics

Syntax SUM(number1,number2,...,number14)

Returns the sum of a list of up to 14 numbers.

Example

SUM(5,7,12)

Returns 24.

TAN

Related Topics

Syntax TAN(angle)

Returns the tangent of **angle**. Uses radians unless a unit of measure is specified with **angle**.

Example

TAN(45 degrees)

Returns 1.

TANH

Related Topics

Syntax TANH(angle)

Returns the hyperbolic tangent of **angle**. Uses radians unless a unit of measure is specified with **angle**.

Example

TANH(45)

Returns -.6558.

TEXTHEIGHT

Related Topics

Syntax TEXTHEIGHT(shapename!theText,maximumwidth)

Returns the height of the composed text in a shape where no text line exceeds **maximumwidth**. The returned value includes the height of the text including the space before and after text, the line spacing in text, and the top and bottom text block margins. TEXTHEIGHT is commonly used to adjust the height of a shape to fit the text it contains.

TheText is a Visio page formula that refers to all of the text contained in a shape's text block. Shapename! can be used to refer to the text of a shape other than the current shape.

Example

TEXTHEIGHT(theText,(Width - 0.5 in.))

Returns the height of the text when wrapped to the width of the shape minus 0.5 inches.

TEXTWIDTH

Related Topics

Syntax TEXTWIDTH(shapename!theText,maximumwidth)

Returns the width of the composed text in a shape. TEXTWIDTH is commonly used to adjust the width of a shape to fit the text it contains.

TheText is a Visio page formula that refers to all of the text contained in a shape's text block. If **maximumwidth** is specified, the result is the longest line of text that fits within **maximumwidth**. If **maximumwidth** is omitted, the result is the total width of the text. Shapename! can be used to refer to the text of a shape other than the current shape.

Example

TEXTWIDTH(theText)

Returns the total length of the text in the current shape.

TIME

Related Topics

Syntax TIME(hour,minute,second)

Returns the time represented by **hour**, **minute**, and **second** formatted according to the time style in the system's Regional Settings.

Examples

TIME(15,30,30)

Returns the value representing 3:30:30 PM.

FORMAT(TIME(15,30,30),"HH:mm")

Returns the value representing 15:30.

TIME(15,30,30) + 8 eh.

Returns the value representing 11:30:30 PM.

TIMEVALUE

Related Topics

Syntax TIMEVALUE("datetime"|expression[,lcid])

Returns the time value represented by **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. Any date component in **datetime** or **expression** is discarded.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

If **datetime** is missing or cannot be converted to a valid result, returns a #VALUE! error.

The returned value is formatted according to the time style set by the system's Regional Settings.

TIMEVALUE also accepts a single number value for **expression** where the decimal portion of the result represents the fraction of a day since midnight.

Examples

TIMEVALUE("6:00 AM")

Returns the value representing 6:00 AM.

TIMEVALUE("14:30")+4 eh.+30 em.

Returns the value representing 7:00:00 PM.

TIMEVALUE("11 AM, July 1, 1997")

Returns the value representing 11:00 AM.

TIMEVALUE(0.6337)

Returns the value representing 3:12:32 PM.

TIMEVALUE("7:89")

Returns a #VALUE! error.

TRUNC

Related Topics

Syntax TRUNC(number,numberofdigits)

Returns **number** truncated by **numberofdigits**.

Examples

TRUNC(123.654,2)

Returns 123.65.

TRUNC(123.654,0)

Returns 123.

TRUNC(123.654,-1)

Returns 120.

UPPER

Related Topics

Syntax UPPER("string")

Returns **string** in uppercase. **String** can be a string, a cell reference, or an expression the result for which is converted to a string.

The case conversion is locale specific, based on the current user (for Microsoft Windows NT) or system (for Microsoft Windows 95) settings.

Example

UPPER("mIxEd CAse")

Returns "MIXED CASE".

USE

Related Topics

Syntax USE("name")

When placed in the LinePattern, FillPattern, BeginArrow, or EndArrow cell, applies the line pattern, fill pattern, or line end called **name** to the shape. This function always returns 254.

Name can be any string that is a valid master name. If a master with **name** is present on a local stencil of the document, the pattern is applied as a line pattern, fill pattern, begin arrow, or end arrow.

Example

USE("Railroad Tracks")

Formats the shape by applying the master pattern named Railroad Tracks to the shape containing the formula.

USERUI

Related Topics

Syntax USERUI(state,defaultexpression,userexpression)

Evaluates one of the two expressions depending on the value of **state**. If **state** is 0, evaluates the **defaultexpression**. Otherwise, if **state** is 1, evaluates the **userexpression**.

Example

USERUI(1,if(Width>6in.,6in.,Width),Width*0.75)

Evaluates the expression Width*.075 and returns the result.

WEEKDAY

Related Topics

Syntax WEEKDAY("datetime"|expression[,lcid])

Returns an integer, 1 to 7, representing the weekday in **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. The time component in **datetime** or **expression** is discarded.

The result corresponds to Monday (1) through Sunday (7). No rounding is done. If **datetime** is missing or cannot be interpreted as a valid date or time, returns a #VALUE! error.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

WEEKDAY also accepts a single number value for **expression** where the integer portion of the result represents the number of days since December 31, 1899.

Examples

WEEKDAY("May 30, 1997")

Returns 5.

WEEKDAY(DATEVALUE("May 30, 1997")+2 ed.)

Returns 7.

=FORMAT(WEEKDAY("May 30, 1997"),"DDDD")

Returns Thursday.

WEEKDAY(35880.6337)

Returns 5.

YEAR

Related Topics

Syntax YEAR("datetime"|expression[,lcid])

Returns an integer representing the year in **datetime** or **expression**. **Datetime** can be any string commonly recognized as a date and time or a reference to a cell containing a date and time. **Expression** can be any expression that yields a date and time. The time component in **datetime** or **expression** is discarded.

The optional argument **lcid** specifies the locale identifier to be used in evaluating a nonlocal **datetime**. The locale identifier is a number described in the system header files.

No rounding is done. If **datetime** is missing or cannot be interpreted as a valid date or time, returns an error.

The returned value is formatted according to the short date style set by the system's current Regional Settings.

YEAR also accepts a single number value for **expression** where the integer portion of the result represents the number of days since December 31, 1899.

Examples

YEAR("10/27/97 3:45:24 PM")

Returns 1997.

YEAR(DATEVALUE("Dec. 25, 1996") + 7 ed.)

Returns 1997.

YEAR(35580.6337)

Returns 1997.

Displaying the ShapeSheet window

You can display a shape, group, guide, guide point, object from another application, master, or page in the ShapeSheet window.

To display a shape, group, guide, guide point, or foreign object in the ShapeSheet window:

1. Select the shape or object.

   To select a shape in a group, open the group window and select the shape. Or subselect a shape in the group.
2. Choose Window > Show ShapeSheet.
3. If you want, choose Window > Tile to see both the drawing page and the ShapeSheet window.

To display a master in the ShapeSheet window:

1. In a stencil opened as original or copy, double-click the master to open the master drawing window.
2. Select the shape or object.
3. Choose Window > Show ShapeSheet.

To display a page in the ShapeSheet window:

1. Make sure nothing is selected on the page.
2. Choose Window > Show ShapeSheet.

TIP Press the Ctrl key and choose Window > Tile to horizontally tile all open windows.

Overview

Displaying values and formulas in cells

You can display either formulas or values in ShapeSheet cells. Display formulas when you are entering new formulas or editing existing ones or to see how formulas in cells relate to each other. A value is the result you get when Visio evaluates a cell's formula. You can display values in cells to see the result of an evaluation.

To display formulas in cells:

- Choose View > Formulas. Or right-click in the ShapeSheet window, then choose Formulas.

  Visio displays formulas in the cells; some long formulas are truncated.

To display values in cells:

- Choose View > Values. Or right-click in the ShapeSheet window, then choose Values.

Overview

Showing and hiding sections

To display the sections you need in the ShapeSheet window, use the Sections command. You can also hide sections you don't need.

To show or hide sections:

1. Choose View > Sections. Or right-click in the ShapeSheet window, then choose View Sections.
2. Check the sections you want to show, or uncheck the sections you want to hide.

   If a section is dimmed in the dialog box, it is not available for the selected shape, page, or object. For example, the Image Info section is available only for objects from other programs. If you don't see the section you want, you must add it.

   You can also choose All to show all the sections, or choose None to hide all the sections.
3. Click OK.

Overview

Collapsing and expanding sections

You can choose which ShapeSheet sections are displayed. You can hide the cells in a section but leave the section heading visible, and you can expand the section to see its cells again.

To collapse or expand a section:

1. Point to the section title.

   If the mouse pointer changes to a minus sign, you can collapse the section.

   If the mouse pointer changes to a plus sign, you can expand the section.

2. Click the section title.

   Visio changes the display to either hide or show the section's cells.

Overview

Moving around the ShapeSheet

To move from cell to cell in the ShapeSheet window, use the arrow and Tab keys. You can also click a cell to activate it or scroll through the ShapeSheet cells without changing the active cell.

To:	Do this:
Activate a cell	Click the cell with the mouse pointer.
Select a numbered row of cells	Click the row number.
Move left one cell	Press the Left Arrow key or Shift+Tab.
Move right one cell	Press the Right Arrow key or Tab.
Move up one cell	Press the Up Arrow key.
Move down one cell	Press the Down Arrow key.
Move to the last cell in a row	Press the End key.
Move to the first cell in a row	Press the Home key.
Scroll down one screen	Press the Page Down key.
Scroll up one screen	Press the Page Up key.
Scroll right one screen	Press Ctrl+Page Down.
Scroll left one screen	Press Ctrl+Page Up.
Scroll to the window's lower-right corner	Press Ctrl+End.
Scroll to the window's upper-right corner	Press Ctrl+Home

Overview

Adding and deleting sections

By adding new sections to a shape in the ShapeSheet window, you can add new characteristics to the shape. For example, you can add a Connection Points section to create connection points. When you no longer need a section, you can delete it.

To add a section:

1. Choose Insert > Section. Or right-click in the ShapeSheet window, then choose Insert Sections.
2. Choose the sections you want to add.

   If a section is dimmed in the dialog box, it is already included or is not available for the selected shape. If the shape is in a group, you may need to open the group in the group window and subselect the shape that you want to change.
3. Click OK.

If you use the Section command to add a section to a shape, but you don't see the new section in the ShapeSheet window, choose View > Sections to make sure the section is set to display.

You can delete a section by using the Delete Section command. You must select a cell in the section before you can delete it.

To delete a section:

- Select a cell in the section you want to delete, then choose Edit > Delete Section.

  Or right-click a cell, then choose Edit > Delete Section.

NOTE Certain sections cannot be deleted. For example, you cannot delete the Shape Transform section because it specifies a shape's position. If you cannot delete a section, Delete Section is dimmed on the menu.

Overview

Adding and deleting rows

You can add rows to ShapeSheet sections to hold additional formulas or to change a shape's appearance. For example, you can add a row to a Geometry section to add a segment to a shape. Similarly, you can delete rows you no longer need.

To add a new row before an existing row:

- Select a cell in the row in the position you want the new row, then choose Insert > Row. Or right-click the cell, then choose Insert Row.

To add a new row after an existing row:

- Select a cell in the row that's one row above the position you want the new row, then choose Insert > Row After. Or right-click the cell, then choose Insert Row After.

To delete a row:

- Select a cell in the row you want to delete, Then choose Edit > Delete Row. Or right-click the cell, then choose Delete Row.

  Visio removes the selected row and moves any following rows up one row in the section.

Overview

Creating formulas

You can type formulas in the formula bar, much as you do in a spreadsheet. If you don't want to replace an entire formula, you can select and edit the part you want to change. For details, see Editing formulas.

To type a new formula for a cell:

1. In the ShapeSheet window, click the cell to select it.

   The cell's formula appears in the formula bar.

2. Type the formula.

   The formula you type replaces the cell's previous formula.

   If you type a number but don't supply units of measure, Visio uses the units from the cell's previous value if there was one. Otherwise, if the cell requires units, Visio uses the default units for the cell. (For details, see Units of measure.)

3. To accept the formula, click the Enter box (marked with a checkmark) in the formula bar or press the Enter key.

   If the number or formula contains an error, Visio displays a message and then highlights the error in the formula bar. Correct the error, and click the Enter box or press the Enter key to accept the correction.

   To cancel the changes, click the Cancel box (marked with an X) in the formula bar, or press the Esc key.

Overview

Editing formulas

You can edit formulas in the formula bar, much as you do in a spreadsheet.

To edit a selected part of a formula:

1. In the ShapeSheet window, select the cell you want.
2. In the formula bar, edit the formula using the following techniques.

To:	Do this:
Place the insertion point	Click where you want the insertion point to appear.
Move the insertion point	Press the Left or Right Arrow key.
Delete the character to the left	Press the Backspace key.
Delete the character to the right	Press the Delete key.
Select text	Drag the mouse over the text.
Extend the selection to the left	Press Shift+Left Arrow.
Extend the selection to the right	Press Shift+Right Arrow.
Select a word	Press Ctrl+Shift+Right Arrow or double-click.
Select to the end	Press Shift+End.
Select to the beginning	Press Shift+Home.
Replace selected text	Type the new text.
Delete selected text	Press the Delete key or choose Cut or Clear from the Edit menu.

Overview

Building a formula by pasting

You can enter ShapeSheet formulas by typing, but Visio also has commands for pasting cell names and functions into a formula.

To paste a cell name into a formula:

1. In the formula bar, place the insertion point where you want to paste the name.

   Until you place the insertion point, the Name command is unavailable.

2. Choose Insert > Name.
3. Choose the name you want from the list.
4. When the formula is finished, click the Enter box in the formula bar to accept the formula, or click the Cancel box to cancel the changes.

TIP You can paste the name of a cell into a formula by placing the insertion point in the formula bar, then clicking the cell you want. Visio inserts the name of the cell at the insertion point.

To paste a function into a formula:

1. In the formula bar, place the insertion point where you want to paste the function.

   Until you place the insertion point, the Function command is unavailable.

2. Choose Insert > Function.
3. Choose the function you want from the list.
4. If you want Visio to include the syntax for the function's arguments, check Insert Arguments.
5. Click OK.
6. Supply the arguments you want for the function by typing them, by clicking ShapeSheet cells, or by using the Name command.
7. When the formula is finished, click the Enter box in the formula bar to accept the formula, or click the Cancel box to cancel the changes.

Overview

Copying and pasting formulas between cells

Often the formula you want is similar to a formula in another cell, either in the same shape or in another one. Rather than retype the formula, you can save time by copying and pasting it into a cell and then editing the formula if needed.

To copy and paste a formula:

1. In the ShapeSheet window, select the cell you want.
2. In the formula bar, select the formula.
3. Choose Edit > Copy.

   Visio copies the formula to the Clipboard.
4. Cancel the activation of the source cell by clicking the Cancel box or pressing the Esc key.

   If you don't cancel the source cell, the cell reference of the next cell you click will be added to the source cell's formula.
5. Select the cell where you want to paste the formula.
6. In the formula bar, select the formula, or click in the formula bar if it is blank.
7. Choose Edit > Paste.

   Visio pastes the formula into the formula bar.
8. Click the Enter box or press the Enter key to accept the formula.

   To discard the formula, click the Cancel box or press the Esc key.

Overview

Using inherited formulas

To use an inherited formula, you can delete the local formula by entering an "empty" one in the cell.

To enter an empty formula:

1. In the ShapeSheet window, select the cell.
2. In the formula bar, delete the formula's text.
3. Press the Enter key.

TIP Try this in a Scratch cell after inserting a local formula. It reverts to No Formula, because there's no formula to inherit from the master.

Deleting all local formulas

An easy way to delete all local formulas in a format section (line, fill, text, character, or paragraph) is to apply an appropriate style to the shape. Applying a style deletes all local formulas in the related cells unless the Preserve Local Formatting option is checked in the Style dialog box when the style is applied.

Overview

Visio and Automation

Related Topics

You can write programs to control Visio in Visual Basic for Applications (VBA), Visual Basic, C++, or any programming language that supports Automation. A program can use Automation to incorporate Visio drawing and diagramming capabilities or to automate simple repetitive tasks in Visio. For example, a program might generate an organization chart from a list of names and positions or print all of the masters on a stencil.

A program controls Visio by accessing its objects and then using their properties, methods, and events. Here is a brief overview:

- Objects represent items you work with in Visio, such as documents, drawing pages, shapes, and formulas.
- Properties are attributes that determine the appearance or behavior of objects. For example, a Shape object has a Name property, which represents the name of that shape.
- Methods are actions that can be performed with an object. For instance, a program can use the Copy method of a Window object to copy a selected shape to the Clipboard. This method is equivalent to selecting a shape on the drawing page and using the Copy command on the Edit menu in Visio.
- Events are occurrences or notifications that trigger code or entire programs. For example, you can programmatically trigger code when a document is opened, or you can trigger a program by double-clicking a shape.

For details about writing programs to control Visio, see *Developing Visio Solutions* (published by Visio Corporation). For details about Visio objects, properties, methods, and events, see the online Visio Automation Reference located on the Visio Help menu.

Using the Visual Basic Editor

You can use the Visual Basic Editor to create, view, debug, and run Visual Basic for Applications (VBA) programs stored in a Visio drawing.

To open the Visual Basic Editor:

- Choose Tools > Macro > Visual Basic Editor.

With the Visual Basic Editor, you can:

- Create VBA programs, by inserting modules, class modules, and user forms into your VBA project.
- View VBA project items by choosing the project of an open Visio document in the Project Explorer. To view the code window for individual items, open the appropriate folder in the Project Explorer and double-click the project item.
- Debug VBA programs by adding breakpoints, including watch expressions, and stepping through program execution.
- Run VBA programs by choosing Run > Run Sub/User Form in the Visual Basic Editor or by choosing Tools > Macro > Macros in Visio.

For details about using the Visual Basic Editor and definitions of VBA programming terms, see *Developing Visio Solutions* (published by Visio Corporation) or the Microsoft Visual Basic online help located on the Visual Basic Editor Help menu.

Overview

Running macros and add-ons

Macros and add-ons are programs that extend the functionality of Visio. Exactly how you run a macro or add-on depends on the context for which it was designed.

There are several ways you can run a program from Visio. Here are a few of the most common ways:

- Choose a macro or add-on from the Visio Macro submenu or the Macros dialog box. If your program is an add-on, the program's .exe file must be located in a folder along the Visio add-ons path, which is specified in the File Paths dialog box, to appear on the Macro menu or in the Macros dialog box.
- Double-click a shape associated with an add-on or macro. The program must be selected in the Run Macro option of the Double-Click dialog box for that shape.
- Right-click a shape and choose a custom menu item for an add-on or macro from the shortcut menu. The program associated with the custom menu item must be entered in the Actions section of the ShapeSheet window for the shape.

If an add-on is designed to be run outside of Visio, you run it like any Windows program (for example, by double-clicking an icon on the desktop). For details, see your Windows documentation.

To run a macro or add-on from the Visio Tools menu:

1. Choose Tools > Macro.
2. From the Macro submenu, choose the program you want to run.
3. Click OK.

To run a macro or add-on from the Visio Macros dialog box:

1. Choose Tools > Macro > Macros.
2. Choose the program you want to run.
3. Click OK.

NOTE If you have more than one Visio drawing open and the macro you want to run does not appear in the Macros dialog box, make sure you have chosen the Visio document in which the macro is stored in the Macros In list.

To double-click a shape to run an associated add-on or macro:

1. Select the shape.
2. Choose Format > Double-Click.
3. In the Double-Click dialog box, choose the Run Macro option.
4. From the list of macros and add-ons, select the one you want to run.
5. Click OK.
6. Double-click the shape to run the program.

To right-click a shape and use the shape's shortcut menu to run an associated add-on or macro:

1. Select the shape.
2. Choose Window > Show ShapeSheet.
3. Click in an Action cell. Insert an Actions section if one doesn't appear by choosing Insert > Section and checking Actions.
4. Choose Edit > Action.

5. Enter the menu and prompt properties and select the program you want to run from the macros and add-ons list.
6. Right-click the shape and choose the custom menu command to run the program.

You can also associate a macro or add-on with a shape by entering a formula that uses the CALLTHIS or RUNADDON function in any ShapeSheet cell.

Overview

Viewing a sample VBA program

Related Topics

This sample VBA program examines the open drawings in an instance of Visio and then prints the path and name of each drawing file, as well as the names of the pages in the drawing, in the Visual Basic Editor's Immediate window. If an open drawing has not been saved, this program prints only the default Visio drawing name.

The sample program does the following:

- Accesses a drawing file that is open in Visio.
- Prints the name and path of the drawing file in the Immediate window.
- Accesses the drawing file pages.
- Prints the name of each page in the Immediate window.
- Repeats each step above until all open drawing files have been printed in the Immediate window.

Here is a look at the code in the program and what it does.

VBA code	What it does
`Public Sub ShowNames ()`	
`Dim iDoc As Integer, iPag As Integer`	'Variable declarations
`Dim pagObj As Visio.Page`	'Visio object type
`Dim docObj As Visio.Document`	'Visio object type
`For iDoc = 1 To Documents.Count`	'Iterate through all open documents.
`Set docObj = Documents(iDoc)`	'Get one drawing.
`Debug.Print docObj.FullName`	'Print the drawing name in the Visual Basic Editor's Immediate window.
`For iPag = 1 To docObj.Pages.Count`	'Iterate through all pages in a drawing.
`Set pagObj = docObj.Pages(iPag)`	'Get a page.
`Debug.Print Tab(5); pagObj.Name`	'Print the page name in the Visual Basic Editor's Immediate window.
`Next iPag`	
`Next iDoc`	
`End Sub`	

Here is an example of the program's output assuming Office.vsd and Recycle.vsd are open and have been saved in the following locations:

Sample output	Description
`c:\visio\solutions\office.vsd`	The name of the first drawing
`Background-1`	The name of page 1
`Background-2`	The name of page 2

`c:\visio\solutions\recycle.vsd`	The name of the second drawing
`Page-1`	The name of page 1
`Page-2`	The name of page 2
`Page-3`	The name of page 3

Adding ActiveX controls

How To
Related Topics

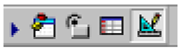
You can add ActiveX controls directly to Visio 5.0 drawings to make your Visio solution interactive. For example, you might add standard Windows dialog box controls such as single-click buttons, checkboxes, or list boxes. Or you might add custom controls that you develop or purchase to incorporate more complex functionality, such as animation.

Using ActiveX controls in your Visio solutions allows you to create a user interface that is consistent with solutions based on other Windows applications. Because the controls are on the drawing page, they're nonmodal—that is, the user can work freely with both controls and Visio shapes without having to display and dismiss a Visual Basic for Applications (VBA) form.

Working in design mode

Related Topics

To work with ActiveX controls in a Visio drawing, you switch between design mode and run mode. In design mode, you can insert controls, move and size them, and set their properties. In run mode, you can use the controls-click a command button to run its **Click** handler, for example. For other tasks, it doesn't matter whether Visio is in design mode or run mode-all other Visio commands and tools work the same way in either mode.



Developer toolbar with the Design Mode button selected

The document's mode is synchronized with that of its VBA project, so both the document and its project are always in the same mode. While a document is in design mode, none of its objects (including controls) issues events.

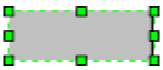
A Visio document opens in run mode by default, unless macro virus protection is set in Visio. To switch to design mode, make sure the Developer toolbar is displayed (if not, choose View > Toolbars > Developer). Then click the Design Mode button, which inverts to indicate that Visio is in design mode.

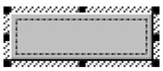
NOTE If macro virus protection is set in Visio, it prompts the user to enable or disable macros when a document is opened. If the user disables macros, the document opens in design mode and cannot be switched to run mode until the document is closed and reopened with macros enabled. To set macro virus protection, choose Tools > Options, then check or clear Macro Virus Protection on the General tab.

Inserting a control in a drawing

How To
Related Topics

Before you can insert an ActiveX control in a Visio drawing, the control must be installed on your system. Certain controls might also require that you have a design license to use them in applications that you develop.


A selected control


A control activated for in-place editing

You insert a control by selecting it in the Insert Control dialog box, which lists all of the ActiveX controls installed on your system, including those installed by other applications. Such applications typically provide a runtime license for the ActiveX controls they contain. The runtime license entitles you to use those controls in the application that contains them, but not to insert the controls in applications that you develop. To insert such controls in your applications, you need a design license for them.

Protecting controls

When you distribute a solution that contains controls, you might want users to be able to edit the shapes in the drawing, but you typically won't want them to be able to edit the controls. You can protect controls from user changes, even in design mode, by locking the shapes and protecting the document.

To insert an ActiveX control in a drawing:

1. On the Developer toolbar, click Insert Control.
2. In the Control dialog box, select a control.

   For example, Microsoft Forms 2.0 Command Button.
3. Click OK to insert the control on the drawing page.
4. Move and size the control as needed.

   A selected control has green selection handles, just like a selected shape, and you move and size it in exactly the same way.
5. Edit the control and set its properties as needed.

   To edit a control, double-click it. A control activated for in-place editing in this way looks the same in Visio as in any ActiveX container. To set a control's properties, use the Visual Basic Editor.

After you insert a control in a drawing, you can work with it in much the same way as with a Visio shape-for example, you can cut or copy and paste the control, duplicate it with Ctrl+drag, or make it into a master by dragging it to a stencil.

Overview

To protect controls from changes

1. Select the controls on the drawing.
2. Choose Format > Protection, then check From Selection.
3. Choose Tools > Protect Document, then check Shapes.
4. For added security, define a password in the Protect Document dialog box.

The user will be able to modify the drawing but not the controls.

Overview

Setting tabbing order of controls

Related Topics

When Visio is in run mode, pressing the Tab key moves the focus from one control to another on the drawing page. If you add more than one control to a drawing, you'll want the focus to move in a logical order.

The tabbing order of controls corresponds to the stacking order of the controls on the drawing page, starting with the backmost control. Initially, this is the order in which you inserted the controls in the drawing, with the most recently inserted control at the front.

To adjust the tab order, use the Bring Forward, Bring To Front, Send Backward, and Send To Back commands on the Visio Shape menu to change the stacking order of controls relative to each other.