About the ShapeSheet spreadsheet

Everything in Microsoft Visio—every document, page, style, shape, group, shape or object within a group, master, object from another program, guide, and guide point—has a ShapeSheet spreadsheet where information about that object is stored. This spreadsheet contains information such as height, width, angle, color, and other attributes that determine the shape's appearance and behavior.

As a shape developer, you need precise control over the appearance and behavior of the shapes you create. You can change a shape's default behavior and enhance what it can do by editing it in its ShapeSheet spreadsheet, which you can access through a ShapeSheet window or programmatically.

- Viewing an object in a ShapeSheet window
- ▶ Structure of the ShapeSheet window
- ▶ Editing an object in the ShapeSheet window
- Additional ShapeSheet spreadsheet references

About formulas

The key to controlling shape actions is to write formulas that define the behavior you want. You can edit a cell's formula to change the value of the cell and, as a result, change a particular shape's behavior. For example, the Height cell in the Shape Transform section contains a formula that you can change to alter the shape's height.

Microsoft Visio formulas are similar to typical spreadsheet formulas in many ways. Visio regards anything in a cell—even if it is a numeric value or simple cell reference—as a formula.

A formula in a cell can be inherited from the equivalent cell of a master or a style or defined locally. Visio evaluates the formula, and then converts the results to an appropriate value and appropriate units for the cell. In a ShapeSheet window, you can display cell contents as either formulas or values.

- ▶ Elements of a formula
- Default formulas
- Inherited formulas
- Local formulas
- Automatic updates to formulas

About operators

You can use operators in formulas to perform arithmetic operations (addition, subtraction, multiplication, and so on) or logical comparisons (greater than, less than, equal to, and so on). You also can control the order of evaluation in a formula by enclosing expressions in parentheses. Use the ampersand operator to combine (concatenate) character strings.

Microsoft Visio automatically attempts to convert data types when an operation or function requires a specific type of data. For example, the multiplication operator requires numeric arguments, and the ampersand (string concatenation) operator requires string arguments. If the argument cannot be converted to the required data type, a default value is provided. The default value is the typed equivalent of nothing: zero for numbers, FALSE for Boolean values, "" for strings, and so on.

The following table shows examples of expressions and their results.

Expression	Result	Description
2 * 5 & " cents"	"10 cents"	The & operator (string concatenation) requires string arguments, so the numeric result of 2 * 5 is automatically converted to the string "10".
5 * "2"	10	The * operator (multiplication)

requires numeric arguments, so the string "2" is automatically converted to the equivalent number 2.

5 * "sheep" 0

The * operator (multiplication) requires numeric arguments, so because the string "sheep" cannot be converted to a number, zero is used as its numeric equivalent.

- Arithmetic operators
- Comparison operators
- Order of evaluation
- Ampersand operator

About functions

A function performs a single well-defined task. Most functions take a number of arguments as input. Although the type and number of arguments depend on the function, all functions have the same general syntax:

FUNCTION(argument1, argument2, ...argumentN [, argumentA | argumentB])

Syntax element	Description
0	If a function takes no arguments, it must be followed by an empty set of parentheses ().
,	Arguments are separated by a comma.
•••	Used for notation only; do not include in a function.
	Optional argument. Used for notation only; do not include in a function.
1	A choice; you can include <i>argumentA</i> or <i>argumentB</i> . Used for notation only; do not include in a function.

Many functions that you can use in formulas resemble those you've seen in spreadsheet programs: mathematical, such as SUM or SQRT; trigonometric, such as SIN or COS; or logical, such as IF or NOT. Many other functions are unique to Microsoft Visio, such as GUARD, GRAVITY, and RUNADDON.

For more information on specific functions, see the **ShapeSheet Reference** in this Developer Reference.

Note Certain functions appear in formulas generated by Microsoft Visio, but are not listed in the **Insert Function** dialog box or described in this Developer Reference because they are reserved for internal use and should not be used in

other formulas. Here are examples: ELLIPSE_THETA, _ELLIPSE_ECC, _UCON_C1, and _SHAPEMIN.

About strings

Formulas can contain strings. 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 function. When you insert a text field, format pictures appear in the list of formats in the **Field** dialog box (on the **Insert** menu, click **Field**).

- Using functions to format strings
- Displaying formatted custom properties

About cell references

You can create interdependencies among formulas by means of ShapeSheet cell references. Cell references give you the power to calculate a value for one cell based on another cell's value. For example, a shape's Width cell might contain a formula that calculates the shape's width by referring to the value of its Height cell, so that when a user resizes the shape vertically, its width stays in proportion.

In a cell's formula, you can refer to a cell of the same shape or another object, such as a document or page, so that Microsoft Visio calculates a value for one cell based on another cell's value.

- What cell references can include
- ▶ Cell reference syntax and examples

About units of measure

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

Microsoft Visio evaluates the result of a formula differently depending on the cell in which you enter the formula. 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 don't require units and evaluate to a string, to TRUE or FALSE, or to an index. For example, the same formula that in the FillForegnd cell means color 5 from the drawing's color palette means TRUE (and locks the shape's width) in the LockWidth cell.

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, type units, duration units, or angular units.

- Units of measure
- ▶ Compound units of measure
- Fractional units of measure
- Multidimensional units of measure
- Universal strings
- ▶ Implicit units of measure

▶ Default units of measure

About date, time, and duration values

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

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 30, 1899. The value to the right of the decimal represents the fraction of a day since midnight. Noon is represented by .5.

To use dates and times within an expression (rather than as a single constant), you must use the appropriate function to identify them as date and time values.

- Valid dates
- Valid times
- Date and time functions
- Duration

About format pictures

A format picture is used to determine how a value will be displayed. For example, you can control the number of digits displayed to the right or left of a decimal point, or whether a text string appears as uppercase or lowercase.

The following tables show symbols you can use to format different types of values for display.

- String and numeric values
- Currency values
- Text values
- Date values
- ▶ Time values

About currency constants

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

If you show currency symbols for nonlocal currencies, and the system does not know the symbol for a given currency, the dollar symbol (\$) is displayed.

▶ IDs and abbreviations

About error values

Error values are displayed 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 function ISERR, ISERRNA, ISERROR, or ISERRVALUE to look for error values.

Error values

Add a new row

See also

Select a cell in the existing row.

On the **Insert** menu, click **Row** to add the new row before the existing row, or click **Row After** to insert the row after the existing row.

Or right-click the cell, and then click **Insert Row** or **Insert Row** After on the shortcut menu.

Add a section

See also

Click in the ShapeSheet window to make it active.

Do one of the following:

On the **Insert** menu, click **Section**.

Or right-click in the ShapeSheet window, and then click **Insert Section** on the shortcut menu.

Select the check box for the section you want to add, and then click **OK**.

Note If you don't see the new section in the ShapeSheet window, make sure the section is set to appear. On the **View** menu, click **Sections**, and then select the check boxes for the sections you want to appear.

Add the Show ShapeSheet command to the shortcut menu

On the **Tools** menu, click **Options**, and then click the **Advanced** tab.

Under **Developer Settings**, select the **Run in developer mode** check box.

Click **OK**.

Change a row type

See also

Select a row by clicking its label.

On the **Edit** menu, click **Change Row Type**.

Or right-click a row label or cell, and then click **Change Row Type** on the shortcut menu.

Change the name and view the ID of a shape

See also

Select a shape in a drawing window.

On the **Format** menu, click **Special**.

In the **Special** dialog box, enter a name.

The ID of the shape is listed at the top of the **Special** dialog box.

Click **OK**.

Collapse or expand a section

See also

Click in the ShapeSheet window to make it active.

Point to the section title.

If the pointer changes to a minus sign \Box , you can collapse the section.

If the pointer changes to a plus sign ⊕, you can expand the section.

Click the section title.

The display changes to either hide or show the section's cells.

Copy and paste a formula from one cell into another cell

See also

Select the cell that contains the formula you want.

On the **Edit** menu, click **Copy**.

To replace the entire formula, select the cell whose formula you want to replace.

To paste the formula into an existing formula, double-click the cell, and then click an insertion point where you want to paste the formula.

On the **Edit** menu, click **Paste**.

Press ENTER to accept the formula.

Customize ShapeSheet menus and toolbars

See also

Click in the ShapeSheet window to make it active.

On the **View** menu, point to **Toolbars**, and then click **Customize**.

Or right-click the toolbar, and then click **Customize** on the shortcut menu.

Delete a row

See also

Select a cell in the row you want to delete.

On the **Edit** menu, click **Delete Row**.

Or right-click the cell, and then click **Delete Row** on the shortcut menu.

Following rows move up one row in the section.

Delete a section

See also

Select any cell in the section you want to delete, and then on the **Edit** menu, click **Delete Section**.

Or right-click a cell, and then click **Delete Section** on the shortcut menu.

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.

Display formulas or values in cells

Click in the ShapeSheet window to make it active.

On the **View** menu, click **Formulas** to view formulas, or click **Values** to view values.

You can also right-click in the ShapeSheet window, and then click **Formulas** or **Values** on the shortcut menu.

Edit a part of a formula

Double-click the cell. Or select the cell and press F2.

In the cell or the formula bar, edit the formula using the techniques in the following table.

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 BACKSPACE.
Delete the character to the right	Press DELETE.
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 DELETE. Or on the Edit menu, click
	Cut.

Enter an empty formula

See also

Select the cell that will contain the formula.

Press DELETE, and then press ENTER.

Tip Try this in a Scratch cell after inserting a local formula. The cell reverts to No Formula, because there is no formula to inherit from the master.

Move around a ShapeSheet window

See also

To move from cell to cell in a ShapeSheet window, use the arrow and TAB keys. Click a cell to activate it or scroll through the ShapeSheet cells without changing the active cell.

You also can use the techniques in the following table.

То	Do this	
Activate a cell	Click the cell with the 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 END.	
Move to the first cell in a row	Press HOME.	
Scroll down one screen	Press PAGEDOWN.	
Scroll up one screen	Press PAGEUP.	
Scroll right one screen	Press CTRL+PAGEDOWN.	
Scroll left one screen	Press CTRL+PAGEUP.	
Scroll to the window's lower-right	Press CTRL+END.	
corner		
Scroll to the window's upper-right Press CTRL+HOME.		
corner		

Insert a function into a formula

See also

Double-click the cell, and then place the insertion point where you want to paste the function.

On the **Insert** menu, click **Function**.

In the **Insert Function** dialog box, click the function you want in the list.

If you want Microsoft Visio to include the syntax for the function's arguments, select the **Insert arguments into formula bar** check box.

Click **OK**.

Supply the arguments you want for the function by typing them, by clicking ShapeSheet cells, or by using the NAME function to select ShapeSheet cells.

When the formula is finished, press ENTER to accept changes. Or press ESC to cancel the changes.

Note Certain functions appear in formulas generated by Visio, but are not listed in the **Insert Function** dialog box or described in this Developer Reference. These functions begin with a single underscore (for example, _ELLIPSE_THETA). They are reserved for internal use and should not be used in other formulas.

Reference a cell in a formula

See also

Double-click the cell in which you want to place the reference, and then place the insertion point at the location you want in the formula.

On the **Insert** menu, click **Name**.

In the **Insert Name** dialog box, click the cell in the list that you want to reference, and then click **OK**.

To insert more than one reference in your formula, click **Insert** after each selection. Click **OK** when you are finished.

Press ENTER to accept the formula. Or press ESC to cancel the changes.

Tip You can quickly reference a cell in a formula by placing the insertion point in the formula bar, clicking the cell you want to reference, and then pressing ENTER. The name of the cell is inserted at the insertion point.

Resize a column in a ShapeSheet window

See also

Click in the ShapeSheet window to make it active.

Place the pointer over the column divider, and then drag the divider to increase or decrease the size of the column.

Show an object in the ShapeSheet window

Visio objects are stored internally as a set of formulas. You can view a Visio object in the ShapeSheet window to see and edit the underlying formulas that determine its appearance and behavior.

For details on how to show a specific object in the ShapeSheet window, click an element in the following list.

- ▶ Show a shape, group, guide, guide point, or foreign object
- Show a master
- Show a page
- Show a document
- Show a style

Show or hide sections

See also

Click in the ShapeSheet window to make it active.

On the **View** menu, click **Sections**. Or right-click in the ShapeSheet window, and then click **View Sections**.

Select the check boxes for the sections you want to show, or clear the check boxes for the sections you want to hide.

Click **OK**.

Show or hide ShapeSheet toolbars

Click in the ShapeSheet window to make it active.

On the **View** menu, click **Toolbars**. Or right-click any toolbar.

Select the toolbar you want to show or hide.

A check mark specifies that a toolbar is shown.

Show the Developer toolbar

On the **View** menu, point to **Toolbars**, and then click **Developer**. Or right-click the toolbar, and then click **Developer** on the shortcut menu.

You then can select an object and click the **Show ShapeSheet** button **■** for quicker access to the ShapeSheet window.

Type a new formula for a cell

Select the cell.

The cell's formula appears in the cell and in the formula bar.

Type the new formula.

The formula you type replaces the cell's previous formula.

If you type a number but don't supply units of measure, Microsoft 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.

To accept the formula, press ENTER.

If the number or formula contains an error, Visio displays a message and then highlights the error in the formula bar or cell. Correct the error, and press ENTER to accept the correction.

A cell (Geometry section)

See also

Represents different information in different rows. This table describes the A cell based on the row in which it's located.

Row	Description
<u>ArcTo</u>	The distance from the arc's midpoint to the midpoint of its chord.
EllipticalArcTo	The <i>x</i> -coordinate of the arc's control point—a point on the arc. The control point is best located about halfway 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.
<u>PolylineTo</u>	The polyline formula.
NURBST ₀	The second to the last knot of the nonuniform rational B-spline (NURBS).
<u>SplineStart</u>	The second knot of the spline.
<u>SplineKnot</u>	One of the spline's knots (other than the last one or the first two).
<u>InfiniteLine</u>	An <i>x</i> -coordinate of a point on the infinite line; paired

with *y*-coordinate represented by the **B** cell.

Ellipse An *x*-coordinate of a point on the ellipse; paired with *y*-

coordinate represented by the **B** cell.

Remarks

To reference the A cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometryi.Aj

where *i* and j = <1>, 2, 3...

Geometryi.A1 (InfiniteLine and Ellipse rows)

where i = <1>, 2, 3...

To reference the A cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: visRowVertex + j

where j = 0, 1, 2...

visRowVertex (InfiniteLine and Ellipse rows)

Cell index: **visBow** (ArcTo row)

visControl (EllipticalArcTo row)
visPolylineData (Polyline row)
visNURBSKnot (NURBSTo row)

visSplineKnot (SplineStart and SplineKnot rows)

visInfiniteLineX2 (InfiniteLine row)

visEllipseMajorX (Ellipse row)

Action cell (Actions section)

See also Rows

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

Remarks

An Action cell is evaluated only when the action occurs, not when the formula is entered.

To reference the Action cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Actions.Action[i]

where i = <1>, 2, 3 ...

To reference the Action cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: $\mathbf{visSectionAction}$ Row index: $\mathbf{visRowAction} + i$

where i = 0, 1, 2...

Cell index: visActionAction

Active cell (Layers section)

See also Rows

Specifies whether the layer is active. Shapes without pre-assigned layers are assigned to the active layer(s) when dropped on the drawing page.

Value	Description
TRUE	Layer is active.
FALSE	Layer is not active.

Remarks

The value in this cell corresponds to the **Active** option in the **Layer Properties** dialog box (click **Layer Properties** on the **View** menu).

To reference the Active cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Layers.Active[i] where i = <1>, 2, 3...

To reference the Active cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionLayer**Row index: **visRowLayer** + i where i = 0, 1, 2...

Cell index: visLayerActive

Address cell (Hyperlinks section)

See also

Specifies a URL address, DOS file name, or UNC path to which to jump.

You can specify Address as a relative path based on the base path defined for the document in the **Hyperlink Base** field on the **Summary** tab of the **Properties** dialog box (click **Properties** on the **File** menu). If the document has no base path, the application navigates based on the document path. If the document hasn't been saved, the hyperlink is undefined.

Remarks

You can also set the value of the Address cell in the **Hyperlinks** dialog box (click **Hyperlinks** on the **Insert** menu).

To reference the Address cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Hyperlink.*Name*.Address

where Hyperlink. *Name* is the name of the hyperlink

row

To reference the Address cell by index from a program, use the **CellsSRC**

property with the following arguments:

Section index: vis Section Hyperlink

visRow1stHyperlink + i where i = 0, 1, 2...Row index:

visHLinkAddress Cell index:

AlignBottom cell (Alignment section)

See also Rows

Determines 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.

Remarks

To reference the AlignBottom cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: AlignBottom

To reference the AlignBottom cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowAlign
Cell index: visAlignBottom

AlignCenter cell (Alignment section)

See also Rows

Determines 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.

Remarks

To reference the AlignCenter cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: AlignCenter

To reference the AlignCenter cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowAlign
Cell index: visAlignCenter

AlignLeft cell (Alignment section)

See also Rows

Determines 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.

Remarks

To reference the AlignLeft cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: AlignLeft

To reference the AlignLeft cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowAlign
Cell index: visAlignLeft

Alignment cell (Tabs section)

See also Rows

Specifies the tab alignment.

Value	Alignment	Automation constant
0	Left	visTabStopLeft
1	Center	visTabStopCenter
2	Right	visTabStopRight
3	Decimal	visTabStopDecimal
4	Comma	visTabStopComma

Remarks

To reference the Alignment cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Tabs.ii

where *i* and j = <1>, 2, 3

To reference the Alignment cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: $\mathbf{visSectionTab}$ Row index: $\mathbf{visRowTab} + i$ where i = 0, 1, 2...Cell index: $(j*3) + \mathbf{visTabAlign}$

AlignMiddle cell (Alignment section)

See also Rows

Determines 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.

Remarks

To reference the AlignMiddle cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: AlignMiddle

To reference the AlignMiddle cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowAlign
Cell index: visAlignMiddle

AlignRight cell (Alignment section)

See also Rows

Determines 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.

Remarks

To reference the AlignRight cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: AlignRight

To reference the AlignRight cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowAlign**Cell index: **visAlignRight**

AlignTop cell (Alignment section)

See also Rows

Determines 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.

Remarks

To reference the AlignTop cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: AlignTop

To reference the AlignTop cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowAlign
Cell index: visAlignTop

Angle cell (Shape Transform section)

See also Rows

Represents 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).

Remarks

To reference the Angle cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Angle

To reference the Angle cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormAngle

Ask cell (Custom Properties section)

See also

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.

Value	Description
TRUE	Ask user to enter custom property data in the Custom Properties dialog box.
FALSE	Do not ask user to enter data.

Remarks

To reference the Ask cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Prop.*Name*.Verify where Prop.*Name* is the name of the custom property row

To reference the Ask cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionProp**Row index: **visRowProp** + *i*

where i = 0, 1, 2,...

Cell index: visCustPropsAsk

AvenueSizeX cell (Page Layout section)

See also Rows

Determines the amount of horizontal space between shapes on the drawing page when you lay out shapes using the **Lay Out Shapes** dialog box (click **Lay Out Shapes** on the **Shape** menu).

Remarks

You can also set this value in the **Layout and Routing Spacing** dialog box (on the **File** menu, click **Page Setup**, click the **Layout and Routing** tab, and then click **Spacing**).

The dynamic grid uses the setting in the AvenueSizeX cell when only one shape is available for calculating horizontal spacing. To use the dynamic grid, click **Snap & Glue** on the **Tools** menu, and then select the **Dynamic Grid** check box under **Currently Active**.

To reference the AvenueSizeX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: AvenueSizeX

To reference the AvenueSizeX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout

Cell index: visPLOAvenueSizeX

AvenueSizeY cell (Page Layout section)

See also Rows

Determines the amount of vertical space between shapes on the drawing page when you lay out shapes using the **Lay Out Shapes** dialog box (click **Lay Out Shapes** on the **Shape** menu).

Remarks

You can also set this value in the **Layout and Routing Spacing** dialog box (on the **File** menu, click **Page Setup**, click the **Layout and Routing** tab, and then click **Spacing**).

The dynamic grid uses the setting in the AvenueSizeY cell when only one shape is available for calculating vertical spacing. To use the dynamic grid, click **Snap & Glue** on the **Tools** menu, and then select the **Dynamic Grid** check box under **Currently Active**.

To reference the AvenueSizeY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: AvenueSizeY

To reference the AvenueSizeY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout

Cell index: visPLOAvenueSizeY

B cell (Geometry section)

See also

Represents different information in different rows. This table describes the B cell based on the row in which it's located.

Row	Description
EllipticalArcTo	The <i>y</i> -coordinate of an arc's control point.
<u>NURBSTo</u>	The last weight of the nonuniform rational B-spline (NURBS).
<u>SplineStart</u>	The first knot of a spline.
<u>InfiniteLine</u>	A <i>y</i> -coordinate of a point on an infinite line; paired with <i>x</i> -coordinate represented by the \underline{A} cell.
Ellipse	A <i>y</i> -coordinate of a point on an ellipse; paired with <i>x</i> -coordinate represented by the \underline{A} cell.

Remarks

To reference the B cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometry i.Bj where i and j = <1>, 2, 3...

Geometryi.B1 (InfiniteLine and Ellipse rows)

To reference the B cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: visRowVertex + j

where j = 0, 1, 2...

visRowVertex (InfiniteLine and Ellipse rows)

Cell index: visControl (EllipticalArcTo row)

visNURBSWeight (NURBSTo row)
visSplineKnot2 (SplineStart row)
visInfiniteLineY2 (InfiniteLine row)

visEllipseMajorY (Ellipse row)

BeginArrow cell (Line Format section)

See also Rows

Indicates whether a line has an arrowhead or other line end format at its first vertex. Enter a number from 0 to 45 or the USE function with the name of a custom line end, or use the **Line** dialog box.

Value	Description
0	No arrowhead.
1 - 45	Assorted arrowhead styles that correspond to indexed entries in the Line dialog box.

Remarks

The size of the arrowhead is set in the BeginArrowSize cell.

To reference the BeginArrow cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: BeginArrow

To reference the BeginArrow cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowLine**

Cell index: visLineBeginArrow

BeginArrowSize cell (Line Format section)

Rows

Determines the size of the arrowhead at the beginning of the line.

Value	Size	Automation constant
0	Very small	visArrowSizeVerySmall
1	Small	visArrowSizeSmall
2	Medium	visArrowSizeMedium
3	Large	visArrowSizeLarge
4	Very large	visArrowSizeVeryLarge
5	Jumbo	visArrowSizeJumbo
6	Colossal	visArrowSizeColossal

Remarks

You can also set the size of the arrowhead in the **Line** dialog box.

To reference the BeginArrowSize cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: BeginArrowSize

To reference the BeginArrowSize cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLine

Cell index: visLineBeginArrowSize

BeginX cell (1-D Endpoints section)

See also Rows

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

Remarks

To reference the BeginX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: BeginX

To reference the BeginX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXForm1D

Cell index: vis1DbeginX

BeginY cell (1-D Endpoints section)

See also Rows

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

Remarks

To reference the BeginY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: BeginY

To reference the BeginY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXForm1D

Cell index: vis1DbeginY

BegTrigger cell (Glue Info section)

See also Rows

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

Remarks

When you glue a 1-D shape to another shape using dynamic glue, the application 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 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.

To reference the BegTrigger cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: BegTrigger

To reference the BegTrigger cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowGroup
Cell index: visBegTrigger

BlockSizeX cell (Page Layout section)

See also Rows

Determines the horizontal block size, the area in which each of your shapes must fit on the drawing page when you lay out shapes using the **Lay Out Shapes** dialog box (click **Lay Out Shapes** on the **Shape** menu).

Remarks

You can also set this value in the **Layout and Routing Spacing** dialog box (on the **File** menu, click **Page Setup**, click the **Layout and Routing** tab, and then click **Spacing**).

To reference the BlockSizeX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: BlockSizeX

To reference the BlockSizeX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOBlockSizeX

BlockSizeY cell (Page Layout section)

See also Rows

Determines the vertical block size, the area in which each of your shapes must fit on the drawing page when you lay out shapes using the **Lay Out Shapes** dialog box (click **Lay Out Shapes** on the **Shape** menu).

Remarks

You can also set this value in the **Layout and Routing Spacing** dialog box (on the **File** menu, click **Page Setup**, click the **Layout and Routing** tab, and then click **Spacing**).

To reference the BlockSizeY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: BlockSizeY

To reference the BlockSizeY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOBlockSizeY

Blur cell (Image Properties section)

See also Rows

Blurs or softens a bitmap image. The default value is 0%.

Remarks

To reference the Blur cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Blur

To reference the Blur cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowImage
Cell index: visImageBlur

BottomMargin cell (Text Block Format section)

See also Rows

Determines 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.

Remarks

To reference the BottomMargin cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: BottomMargin

To reference the BottomMargin cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowText

Cell index: visTxtBlkBottomMargin

Brightness cell (Image Properties section)

See also Rows

Adjusts the brightness of a bitmap image. Decrease brightness of an image by entering a value between 0% and 49%, or increase brightness by entering a value between 51% and 100%. The default value is 50%.

Remarks

To reference the Brightness cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Brightness

To reference the Brightness cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowImage**

Cell index: visImageBrightness

Bullet cell (Paragraph section)

See also Rows

Determines the bullet style.

Value	Bullet style
0	None
1	•
2	•
3	•
4	•
5	*
6	>
7	✓

Remarks

You can also set the value of this cell by selecting a shape, clicking **Text** on the **Format** menu, and then clicking the **Bullets** tab.

To reference the Bullet cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Para.Bullet[i] where i = <1>, 2, 3, ...

To reference the Bullet cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionParagraph**Row index: **visRowParagraph** + *i*

where i = 0, 1, 2, ...

Cell index: visBulletIndex

BulletString cell (Paragraph section)

See also Rows

Allows you to create a custom bullet style. Enter the style as a string (within quotation marks). For example, you could enter the string, "ooo."

Remarks

You can also set the value of this cell by selecting a shape, clicking **Text** on the **Format** menu, and then clicking the **Bullets** tab.

To reference the BulletString cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Para.BulletStr[i]

where i = <1>, 2, 3, ...

To reference the BulletString cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionParagraph**Row index: **visRowParagraph** + *i*

where i = 0, 1, 2, ...

Cell index: visBulletString

C cell (Geometry section)

See also

Represents different information in different rows. This table describes the C cell based on the row in which it's located.

Row	Description
EllipticalArcTo	The angle of an arc's major axis relative to the <i>x</i> -axis of its parent.
NURBSTo	The first knot of the nonuniform rational B-spline (NURBS).
<u>SplineStart</u>	The last knot of a spline.
Ellipse	An <i>x</i> -coordinate of a point on an ellipse; paired with the <i>y</i> -coordinate represented by the \underline{D} cell.

Remarks

To reference the C cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometry i. Cj where i and j = <1>, 2, 3... Geometry i. C1 (Ellipse row)

To reference the C cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: $\mathbf{visRowVertex} + j$

where j = 0, 1, 2...

visRowVertex (Ellipse row)

Cell index: visEccentricityAngle (EllipticalArcTo row)

visNURBSKnotPrev (NURBSTo row)

visSplineKnot3 (SplineStart row)
visEllipseMinorX (Ellipse row)

Can Glue cell (Controls section)

See also Rows

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

Value	Description
TRUE	Control handle can be glued.
FALSE	Control handle cannot be glued.

Remarks

To reference the Can Glue cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Controls.CanGlue[i] where i = <1>, 2, 3, ...

To reference the Can Glue cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: $\mathbf{visSectionControls}$ Row index: $\mathbf{visRowControl} + i$

where i = 0, 1, 2, ...

Cell index: visCtlGlue

Case cell (Character section)

See also Rows

Determines the case of a shape's text. All capital (uppercase) letters (1) and initial capital letters (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.

Value	Description	Automation constant
0	Normal case	visCaseNormal
1	All capital (uppercase) letters	visCaseAllCaps
2	Initial capital letters only	visCaseInitialCaps

Remarks

To reference the Case cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Case[i]

where i = <1>, 2, 3, ...

To reference the Case cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2, ...

Cell index: visCharacterCase

Checked cell (Actions section)

See also Rows

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

Value	Description
TRUE	Check mark is displayed.
FALSE	Check mark is not displayed.

Remarks

To reference the Checked cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Actions.Ci

where i = <1>, 2, 3, ...

To reference the Checked cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionAction**Row index: **visRowAction** + *i*

where i = 0, 1, 2, ...

Cell index: visActionChecked

Color cell (Character section)

Rows

Determines the color used for the shape's text.

Remarks

To set the color, enter a number from 0 to 23. This number corresponds to a color in the current color palette, which you can also set in the **Color Palette** dialog box (click **Color Palette** on the **Tools** menu).

To enter a custom color, use the RGB or HSL function. The value of a custom color is its RGB color, and RGB(r, g, b), rather than a number, will be shown in the ShapeSheet window. When used in numeric operations, custom colors have values of 24 and above.

You can set the transparency of the text color in the Transparency cell.

To reference the Color cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Color[*i*]

where i = <1>, 2, 3, ...

To reference the Color cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2, ...

Cell index: visCharacterColor

Color cell (Layers section)

Rows

Specifies the color used to display the layer.

Remarks

To set the color, enter a number from 0 to 23. This number corresponds to a color in the current color palette, which you can also set in the **Color Palette** dialog box (click **Color Palette** on the **Tools** menu).

This cell corresponds to the **Color** option in the **Layer Properties** dialog box (click **Layer Properties** on the **View** menu).

To enter a custom color, use the RGB or HSL function. The value of a custom color is its RGB color, and RGB(r, g, b), rather than a number, will be shown in the ShapeSheet window. When used in numeric operations, custom colors have values of 24 and above.

You can set the transparency of the layer color in the Transparency cell.

To reference the Color cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Layers.Color[*i*]

where i = <1>, 2, 3, ...

To reference the Color cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionLayer**Row index: **visRowLayer** + *i*

where i = 0, 1, 2, ...

Cell index: visLayerColor

Comment cell (Miscellaneous section)

See also Rows

Contains the comment text in string format for a shape.

Remarks

You can also insert a comment using the **Comment** dialog box (click **Comment** on the **Insert** menu).

To reference the Comment cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Comment

To reference the Comment cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowMisc
Cell index: visComment

ConFixedCode cell (Shape Layout section)

See also Rows

Determines when a connector reroutes.

Value	Description	Automation constant
0	Reroute freely	visSLOConFixedRerouteFreely
1	Reroute as needed (manual reroute)	visSLOConFixedRerouteAsNeeded
2	Never reroute	visSLOConFixedRerouteNever
3	Reroute on crossover	vis SLOC on Fixed Reroute On Crossover
4	For internal use only	visSLOConFixedByAlgFrom
5	For internal use only	visSLOConFixedByAlgTo
6	For internal use only	vis SLOC on Fixed By Alg From To

Remarks

You can also set the value of this cell by selecting a connector, clicking **Behavior** on the **Format** menu, and then clicking the **Connector** tab.

To reference the ConFixedCode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ConFixedCode

To reference the ConFixedCode cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowShapeLayout
Cell index: visSLOConFixedCode

ConLineJumpCode cell (Shape Layout section)

See also Rows

Determines when a connector jumps.

Value	Description	Automation constant
0	As page specifies; on the File menu, click Page Setup , and then click the Layout and Routing tab to see the page specifications	visSLOJumpDefault
1	Never	visSLOJumpNever
2	Always	visSLOJumpAlways
3	Other connector jumps	visSLOJumpOther
4	Neither connector jumps	visSLOJumpNeither

Remarks

You can also set the value of this cell by selecting a connector, clicking **Behavior** on the **Format** menu, and then clicking the **Connector** tab.

To reference the ConLineJumpCode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ConLineJumpCode

To reference the ConLineJumpCode cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowShapeLayout
Cell index: visSLOJumpCode

ConLineJumpDirX cell (Shape Layout section)

Rows

Determines the line jump direction for line jumps occurring on a horizontal dynamic connector for a shape.

Value	Line jump direction	Automation constant
0	Page default	visLOJumpDirXDefault
1	Up	visLOJumpDirXUp
2	Down	visLOJumpDirXDown

Remarks

To set the default horizontal direction for *all* connector jumps on a page, use the PageLineJumpDirX cell in the Page Layout section.

To reference the ConLineJumpDirX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ConLineJumpDirX

To reference the ConLineJumpDirX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowShapeLayout**

Cell index: visSLOJumpDirX

ConLineJumpDirY cell (Shape Layout section)

Rows

Determines the line jump direction for line jumps occurring on a vertical dynamic connector for a shape.

Value	Line Jump Direction	Automation constant
0	Page default	visLOJumpDirYDefault
1	Left	visLOJumpDirYLeft
2	Right	visLOJumpDirYRight

Remarks

To set the default vertical direction for *all* connector jumps on a page, use the PageLineJumpDirY cell in the Page Layout section.

To reference the ConLineJumpDirY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ConLineJumpDirY

To reference the ConLineJumpDirY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowShapeLayout**

Cell index: visSLOJumpDirY

ConLineJumpStyle cell (Shape Layout section)

See also Rows

Determines the line jump style for line jumps on a dynamic connector.

Value	Line Jump Style	Automation constant
0	Page default	visLOJumpStyleDefault
1	Arc	visLOJumpStyleArc
2	Gap	visLOJumpStyleGap
3	Square	visLOJumpStyleSquare
4	Triangle	visLOJumpStyleTriangle
5	3 sides	visLOJumpStyle2Point
6	4 sides	visLOJumpStyle3Point
7	5 sides	visLOJumpStyle4Point
8	6 sides	visLOJumpStyle5Point
9	7 sides	visLOJumpStyle6Point

Remarks

You can also set the value of this cell by selecting a dynamic connector, clicking **Behavior** on the **Format** menu, and then clicking the **Connector** tab.

To reference the ConLineJumpStyle cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ConLineJumpStyle

To reference the ConLineJumpStyle cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowShapeLayout
Cell index: visSLOJumpStyle

ConLineRouteExt cell (Shape Layout section)

Rows

Determines the appearance of a connector.

Value	Description	Automation constant
0	Default; use page setting	visLORouteExtDefault
1	Straight	visLORouteExtStraight
2	Curved	visLORouteExtNURBS

Remarks

To reference the ConLineRouteExt cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ConLineRouteExt

To reference the ConLineRouteExt cell by index from a program, use the CellsSRC property with the following arguments:

Section index: **visSectionObject**Row index: **visRowShapeLayout**

Cell index: visSLOLineRouteExt

Contrast cell (Image Properties section)

See also Rows

Adjusts the contrast of a bitmap image. Decrease the contrast of an image by entering a value between 0% and 49%, or increase the contrast by entering a value between 51% and 100%. The default value is 50%.

Remarks

To reference the Contrast cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Contrast

To reference the Contrast cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowImage

Cell index: visImageContrast

CtrlAsInput cell (Page Layout section)

See also Rows

Determines which shape is the parent when using shapes with control handles. This cell sets the behavior for all the shapes on the drawing page.

Value	Description
TRUE	Set the shape that the control handle is connected to as
	the parent.
FALSE	The default. Set shape that contains the control handle as the parent.

Remarks

To reference the CtrlAsInput cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: CtrlAsInput

To reference the CtrlAsInput cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOCtrlAsInput

D cell (Connection Points section)

See also

A scratch cell that you can use for entering or testing formulas.

Remarks

To access the D cell, right-click a row, and then click **Change Row Type** on the shortcut menu.

To reference the D cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Connections. AutoGen[i]

where i = <1>, 2, 3...

To reference the D cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionConnectionPts**Row index: **visRowConnectionPts** + *i*

where i = 0, 1, 2...

Cell index: visCnnctAutoGen

D cell (Geometry section)

Represents different information in different rows. This table describes the D cell based on the row in which it's located.

Row	Description
EllipticalArcTo	The ratio of an arc's major axis to its minor axis. Despite the usual meaning of these words, the "major" axis does not have to be greater than the "minor" axis, so this ratio does not have to be greater than 1. Setting this cell to a value less than or equal to 0 or greater than 1000 can lead to unpredictable results.
NURBST ₀	The first weight of the nonuniform rational B-spline (NURBS).
<u>SplineStart</u>	The degree of a spline (an integer from 1 to 25).
Ellipse	A <i>y</i> -coordinate of a point on an ellipse; paired with the x -coordinate represented by the C cell.

Remarks

To reference the D cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometryi.Dj

where i and j = <1>, 2, 3... Geometryi.D1 (Ellipse row)

where i = <1>, 2, 3...

To reference the D cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: $\mathbf{visRowVertex} + j$

where j = 0, 1, 2...

visRowVertex (Ellipse row)

Cell index visAspectRatio (EllipticalArcTo row)

visNURBSWeightPrev (NURBSTo row)

visSplineDegree (SplineStart row)
visEllipseMinorY (Ellipse row)

Default cell (Hyperlinks section)

See also

Determines the default hyperlink for a shape or page. Set the value of this cell to TRUE to set a hyperlink as the default.

Remarks

You can also set the default hyperlink by selecting a shape, clicking **Hyperlinks** on the **Insert** menu, selecting a hyperlink, and then clicking **Default**. The default hyperlink appears in bold text.

To reference the Default cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Hyperlink.*Name*.Default

where Hyperlink. Name is the row name

To reference the Default cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionHyperlink**Row index: **visRow1stHyperlink** + *i*

where i = 0, 1, 2...

Cell index: visHLinkDefault

DefaultTabstop cell (Text Block Format section)

See also Rows

Determines the interval of the default tab stops in a text block. The default value is 0.5 inches for documents created in imperial units and 1.5 centimeters for documents created in metric units.

Remarks

Whether a document is created in imperial or metric units depends on the setting in the **Page** box on the **Regional** tab in the **Options** dialog box (on the **Tools** menu, click **Options**.)

To reference the DefaultTabstop cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: DefaultTabstop

To reference the DefaultTabstop cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowText**

Cell index: visTxtBlkDefaultTabStop

Denoise cell (Image Properties section)

See also Rows

Removes noise (pixels with randomly distributed color levels) from a bitmap image. The default value is 0%.

Remarks

To reference the Denoise cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Denoise

To reference the Denoise cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowImage
Cell index: visImageDenoise

Description cell (Hyperlinks section)

See also

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

Remarks

You can also set the value of this cell using the **Hyperlinks** dialog box (click **Hyperlinks** on the **Insert** menu).

To reference the Description cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Description

To reference the Description cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowImage**

Cell index: visImageDescription

DirX / A cell (Connection Points section)

See also

Determines the *x*-component for the required alignment vector of a matching connection point. The DirX / A cell is also used to orient the attached leg of a dynamic connector. This cell takes a floating point value.

Remarks

To reference the DirX / A cell by name from another formula, or from a program using the **Cells** property, use:

Cell name: Connections.DirX[*i*]

where i = <1>, 2, 3...

To reference the DirX / A cell by index from a program, use the CellsSRC property with the following arguments:

Section index: visSectionConnectionPts visRowConnectionPts + iRow index:

where i = 0, 1, 2

Cell index: visCnnctDirX (non-extended rows) visCnnctA (extended rows)

For information about non-extended and extended rows, see Connections.Row row.

DirY / B cell (Connection Points section)

See also

Determines the *y*-component for the required alignment vector of a matching connection point. It is also used to orient the attached leg of a dynamic connector. This cell takes a floating point value.

Remarks

To reference the DirY / B cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: DirY[i]

where i = <1>, 2, 3...

To reference the DirY / B cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionConnectionPts**Row index: **visRowConnectionPts**+ *i*

where i = 0, 1, 2...

Cell index: visCnnctDirY (non-extended rows) visCnnctB (extended rows)

For information about non-extended and extended rows, see Connections.Row row.

Disabled cell (Actions section)

See also Rows

Toggles the display of the command name on the shortcut menu.

Value	Description
TRUE	Disable (dim) command name.
FALSE	Enable (highlight) command name.

Remarks

To reference the Disabled cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Actions.Di

where i = <1>, 2, 3 ...

To reference the Disabled cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionAction**Row index: **visRowAction** + *i*

visRowAction + i where i = 0, 1, 2...

Cell index: visActionDisabled

DisplayMode cell (Group Properties section)

See also Rows

Determines how the group shape and its members are displayed.

Value	Display Mode	Automation constant
0	Hides the group shape and text.	visGrpDispModeNone
1	Displays the group shape behind member shapes.	visGrpDispModeBack
2	Displays the group shape in front of member shapes.	visGrpDispModeFront

Remarks

You can also set this value by selecting the group, clicking **Behavior** on the **Format** menu, and then selecting a display mode from the **Group data** list.

To reference the DisplayMode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: DisplayMode

To reference the DisplayMode cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowGroup**

Cell index: visGroupDisplayMode

DontMoveChildren cell (Group Properties section)

See also Rows

Determines whether you can drag shapes in a group using the mouse.

Value	Description
TRUE	Don't allow shapes in a group to be dragged using the mouse.
FALSE	Allow shapes in a group to be dragged using the mouse.

Remarks

When the value of this cell is TRUE, you can still flip, rotate, resize, or reposition shapes in groups using other methods.

The value of this cell is TRUE for groups in masters and groups in instances of masters that were created in versions of Microsoft Visio earlier than Visio 2000.

To reference the DontMoveChildren cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: DontMoveChildren

To reference the DontMoveChildren cell by index from a program, use the CellsSRC property with the following arguments:

Section index: **visSectionObject**Row index: **visRowGroup**

Cell index: visGroupDontMoveChildren

DoubleULine cell (Character section)

See also Rows

Determines whether the range of text has a double underline below it.

Value	Description
TRUE	Text has a double underline below it.
FALSE	Text does not have a double underline below it.

Remarks

The DoubleULine cell contains formatting information applied to a sub-range of a shape's text if the Characters section contains multiple rows. Otherwise, it contains formatting information for all of the shape's text.

You can also set the value of this cell by using the **Text** dialog box (click **Text** on the **Format** menu).

To reference the DoubleULine cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.DblUnderline[i] where i = <1>, 2, 3...

To reference the DoubleULine cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + iwhere i = 0, 1, 2...

Cell index: visCharacterDblUnderline

DrawingScale cell (Page Properties section)

Rows

Represents 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.

Remarks

This cell corresponds to the settings in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

The units of the formula in the DrawingScale cell determine the measurement units used by the rulers in the drawing window. If you do not want to also change the drawing's scale, you can do one of the following:

Keep the distance expressed in the DrawingScale cell the same but express it in different units.

Change the distance expressed in the PageScale cell by the same factor that you change DrawingScale.

To reference the DrawingScale cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: DrawingScale

To reference the DrawingScale cell by index from a program, use the CellsSRC

property with the following arguments:

Section index: visSectionObject

Row index: **visRowPage**

Cell index: visPageDrawingScale

Example

You can set the DrawingScale cell to change the units of a page's rulers from a program. Here's an example of changing the measurement units from inches to centimeters from a program. In this case we use the <u>ConvertResult</u> method to keep the distance the same but express it in different units.

```
Public Sub SetActivePageMeasurementToCM()
Dim dsCell As Visio.Cell
Set dsCell = ActivePage.PageSheet.Cells("DrawingScal dsCell.Result(visCentimeters) = _
Application.ConvertResult _
(dsCell.ResultIU,visInches,visCentimeters)
End Sub
```

You can determine the measurement system in a drawing by examining the <u>Units</u> property of the DrawingScale cell. After running the above macro the following statement executed in the Visual Basic Editor Immediate window will return **True**.

debug.print ActivePage.PageSheet.Cells("DrawingScale" visCentimeters

DrawingScaleType cell (Page Properties section)

See also Rows

Determines the drawing scale selected in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

Value	Description	Automation constant
0	No Scale	visNoScale
1	Architectural Scale	visArchitectural
2	Civil Engineering Scale	visEngineering
3	Custom Scale	visScaleCustom
4	Metric	visScaleMetric
5	Mechanical Engineering Scale	visScaleMechanical

Remarks

To reference the DrawingScaleType cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: DrawingScaleType

To reference the DrawingScaleType cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowPage

Cell index: visPageDrawScaleType

DrawingSizeType cell (Page Properties section)

See also Rows

Determines the drawing size.

Value	Description	Automation constant
0	Same as printer	visPrintSetup
1	Fit page to drawing contents	visTight
2	Standard	visStandard
3	Custom page size	visCustom
4	Custom scaled drawing size	visLogical
5	Metric (ISO)	visDSMetric
6	ANSI Engineering	visDSEngr
7	ANSI Architectural	visDSArch

Remarks

To set the drawing size, use the **Page Setup** dialog box (click **Page Setup** on the **File** menu) or manually resize the page with the mouse.

To reference the DrawingSizeType cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: DrawingSizeType

To reference the DrawingSizeType cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowPage

Cell index: visPageDrawSizeType

DynamicsOff cell (Page Layout section)

See also Rows

Determines whether placeable shapes move and connectors reroute around other shapes and connectors on the drawing page.

Value	Description
TRUE	Disable dynamics.
FALSE	Enable dynamics.

Remarks

You can disable dynamics to increase your solution's performance. For example, if your solution adds placeable shapes to a drawing and you don't want the application to reroute connectors and reposition shapes each time you add a shape, you can disable dynamics. After your solution adds the shapes, re-enable dynamics.

To reference the DynamicsOff cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: DynamicsOff

To reference the DynamicsOff cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout

Cell index: visPLODynamicsOff

DynFeedback cell (Miscellaneous section)

See also Rows

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.

Value	Description	Automation constant
0	Remains straight (no legs).	visDynFBDefault
1	Shows three legs when dragged.	visDynFBUCon3Leg
2	Shows five legs when dragged.	visDynFBUCon5Leg

Remarks

To reference the DynFeedback cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: DynFeedback

To reference the DynFeedback cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowMisc

Cell index: visDynFeedback

E cell (Geometry section)

See also

Contains a nonuniform rational B-spline (NURBS) formula.

Remarks

To reference the E cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometryi.E*j*

where *i* and j = <1>, 2, 3...

To reference the E cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: visRowVertex + j

where j = 0, 1, 2...

Cell index: visNURBSData

EditMode cell (Text Fields section)

See also Rows

Reserved for future use.

EnableFillProps cell (Style Properties section)

See also Rows

Determines whether a style includes fill properties.

Value	Description
TRUE	Include fill properties.
FALSE	Exclude fill properties.

Remarks

You can also set this value by using the **Define Styles** dialog box (click **Define Styles** on the **Format** menu).

To reference the EnableFillProps cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EnableFillProps

To reference the EnableFillProps cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowStyle**

Cell index: visStyleIncludesFill

EnableGrid cell (Page Layout section)

See also Rows

Determines whether the application lays out shapes based on an internal, invisible page grid when using the **Lay Out Shapes** command. This grid is created using the **Space between shapes** and the **Average shape size** values in the **Layout and Routing Spacing** dialog box (click **Lay Out Shapes** on the **Shape** menu, and then click **Spacing**).

Value	Description
TRUE	Use the internal page grid.
FALSE	Do not use the internal page grid.

Remarks

When you enable this feature, the application aligns each placeable shape's center point with the center of a block on the internal page grid.

To reference the EnableGrid cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EnableGrid

To reference the EnableGrid cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOEnableGrid

EnableLineProps cell (Style Properties section)

See also Rows

Determines whether a style includes line properties.

Value	Description
TRUE	Include line properties.
FALSE	Exclude line properties.

Remarks

You can also set this value by using the **Define Styles** dialog box (click **Define Styles** on the **Format** menu).

To reference the EnableLineProps cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EnableLineProps

To reference the EnableLineProps cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowStyle**

Cell index: visStyleIncludesLine

EnableTextProps cell (Style Properties section)

See also Rows

Determines whether a style includes text properties.

Value	Description
TRUE	Include text properties.
FALSE	Exclude text properties.

Remarks

You can also set this value by using the **Define Styles** dialog box (click **Define Styles** on the **Format** menu).

To reference the EnableTextProps cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EnableTextProps

To reference the EnableTextProps cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowStyle**

Cell index: visStyleIncludesText

EndArrow cell (Line Format section)

Rows

Indicates whether a line has an arrowhead or other line end format at its last vertex.

Value	Description
0	No arrowhead.
1 - 45	Assorted arrowhead styles that correspond to indexed entries in the Line dialog box.

Remarks

You can also set this value in the **Line** dialog box (click **Line** on the **Format** menu). The size of the arrowhead is set in the EndArrowSize cell.

You can specify a custom line end using the USE function in this cell.

To reference the EndArrow cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EndArrow

To reference the EndArrow cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowLine**

Cell index: visLineEndArrow

EndArrowSize cell (Line Format section)

See also Rows

Determines the size of the arrowhead at the end of the line.

Value	Size	Automation constant
0	Very small	visArrowSizeVerySmall
1	Small	visArrowSizeSmall
2	Medium	visArrowSizeMedium
3	Large	visArrowSizeLarge
4	Extra large	visArrowSizeVeryLarge
5	Jumbo	visArrowSizeJumbo
6	Colossal	visArrowSizeColossal

Remarks

You can also set this value in the **Line** dialog box (click **Line** on the **Format** menu).

To reference the EndArrowSize cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EndArrowSize

To reference the EndArrowSize cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLine

Cell index: visLineEndArrowSize

EndTrigger cell (Glue Info section)

Rows

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

Remarks

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 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.

To reference the EndTrigger cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EndTrigger

To reference the EndTrigger cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowMisc**Cell index: **visEndTrigger**

EndX cell (1-D Endpoints section)

See also Rows

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

Remarks

To reference the EndX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EndX

To reference the EndX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowXForm1D**

Cell index: vis1DendX

EndY cell (1-D Endpoints section)

See also Rows

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

Remarks

To reference the EndY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EndY

To reference the EndY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowXForm1D**

Cell index: vis1DendY

EventDblClick cell (Events section)

See also Rows

An event cell that is evaluated when a shape is double-clicked.

Remarks

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

To reference the EventDblClick cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EventDblClick

To reference the EventDblClick cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowEvent**

Cell index: visEvtCellDblClick

EventDrop cell (Events section)

See also Rows

An event cell that is evaluated when a shape is dropped on the drawing page, either as an instance or when the shape is duplicated or pasted.

Remarks

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

To reference the EventDrop cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EventDrop

To reference the EventDrop cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowEvent**Cell index: **visEvtCellDrop**

EventXFMod cell (Events section)

See also Rows

An event cell that is evaluated when a shape's position or orientation on the page is transformed ("XF").

Remarks

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

To reference the EventXFMod cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: EventXFMod

To reference the EventXFMod cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: **visRowEvent**

Cell index: visEvtCellXFMod

ExtraInfo cell (Hyperlinks section)

See also

Represents 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.

Remarks

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

To reference the ExtraInfo cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Hyperlink.*Name*.Frame

where Hyperlink. Name is the row name

To reference the ExtraInfo cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionHyperlink**Row index: **visRow1stHyperlink** + *i*

where i = 0, 1, 2...

Cell index: visHLinkExtraInfo

FillBkgnd cell (Fill Format section)

Rows

Determines the color used for the background (fill) of the shape's fill pattern.

Remarks

To set the color, enter a number from 0 to 23. The number corresponds to a color in the current color palette, which you can also set in the **Color Palette** dialog box (click **Color Palette** on the **Tools** menu).

To enter a custom color, use the RGB or HSL function. The value of a custom color is its RGB color, and RGB(r, g, b), rather than a number, will be shown in the ShapeSheet window. When used in numeric operations, custom colors have values of 24 and above.

You can set the transparency of the background fill in the FillBkgndTrans cell.

To reference the FillBkgnd cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: FillBkgnd

To reference the FillBkgnd cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: **visRowFill**Cell index: **visFillBkgnd**

FillBkgndTrans cell (Fill Format section)

Rows

Determines the transparency level for the background (fill) color of the shape's fill pattern.

Value	Description
0 – 100	Represents the percentage of transparency. The default is 0% (completely opaque).

Remarks

Values are rounded to the nearest half percent. A value of 100% is completely transparent. Although a shape with a completely transparent fill appears the same as a shape with no fill on the drawing page, it will interact with other objects on the page in the same ways as if its transparency is 0%.

You can also set this value using the slider control in the **Fill** dialog box (click **Fill** on the **Format** menu). This value controls the value of both the background and foreground fill transparencies. To set these values independently, you must enter them in the ShapeSheet window.

To reference the FillBkgndTrans cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: FillBkgndTrans

To reference the FillBkgndTrans cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowFill

Cell index: visFillBkgndTrans

FillForegnd cell (Fill Format section)

Rows

Determines the color used for the foreground (stroke) of the shape's fill pattern.

Remarks

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 in the **Color Palette** dialog box (click **Color Palette** on the **Tools** menu).

To enter a custom color, use the RGB or HSL function. The value of a custom color is its RGB color, and RGB(r, g, b), rather than a number, will be shown in the ShapeSheet window. When used in numeric operations, custom colors have values of 24 and above.

You can set the transparency of the foreground fill in the FillForegndTrans cell.

To reference the FillForegnd cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: FillForegnd

To reference the FillForegnd cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowFill

Cell index: visFillForegnd

FillForegndTrans cell (Fill Format section)

Rows

Determines the transparency level for the foreground (fill) color of the shape's fill pattern.

Value	Description
0 – 100	Represents the percentage of transparency. The default is 0% (completely opaque).

Remarks

Values are rounded to the nearest half percent. A value of 100% is completely transparent. Although a shape with a completely transparent fill appears the same as a shape with no fill on the drawing page, it will interact with other objects on the page in the same ways as if its transparency is 0%.

You can also set this value using the slider control in the **Fill** dialog box (click **Fill** on the **Format** menu). This value controls the value of both the background and foreground fill transparencies. To set these values independently, you must enter them in the ShapeSheet window.

To reference the FillForegndTrans cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: FillForegndTrans

To reference the FillForegndTrans cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowFill

Cell index: visFillForegndTrans

FillPattern cell (Fill Format section)

Rows

Determines the fill pattern for the shape. To specify a custom fill pattern, use the USE function in this cell.

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

Remarks

You can also set this value using the **Fill** dialog box (click **Fill** on the **Format** menu).

To reference the FillPattern cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: FillPattern

To reference the FillPattern cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowFill
Cell index: visFillPattern

FlipX cell (Shape Transform section)

See also Rows

Indicates whether the shape has been flipped horizontally.

Value	Description
TRUE	The shape has been flipped horizontally.
FALSE	The shape has not been flipped horizontally.

Remarks

To reference the FlipX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: FlipX

To reference the FlipX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormFlipX

FlipY cell (Shape Transform section)

See also Rows

Indicates whether the shape has been flipped vertically.

Value	Description
TRUE	The shape has been flipped vertically.
FALSE	The shape has not been flipped vertically.

Remarks

To reference the FlipY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: FlipY

To reference the FlipY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormFlipY

Font cell (Character section)

See also Rows

Represents 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 typically Arial.

Remarks

To reference the Font cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Font[*i*]

where i = <1>, 2, 3...

To reference the Font cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2...

Cell index: visCharacterFont

Format cell (Custom Properties section)

Specifies the formatting of a custom property that is a string, a fixed list, a number, a variable list, a date or time, a duration, or a currency.

Remarks

Custom property type	Value	Format cell contents
String	0	A format picture appropriate for the data type
Fixed list	1	The items to appear in the list, separated by commas
Number	2	A format picture appropriate for the data

type

Variable list	4	The items to appear in the list, separated by commas
Date or time	5	A format picture appropriate for the data type
Duration	6	A format picture appropriate for the data type
Currency	7	A format picture appropriate for the data type

As an example of specifying a format picture appropriate for the data type, the format picture "# #/4 UU" formats the number 12.43 in. as 12 2/4 INCHES. For more information about specifying a format picture, see About format pictures.

An example of specifying items for a list is "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.

To reference the Format cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Prop.*Name*.Format

where Prop. Name is the row name

To reference the Format cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionProp**Row index: **visRowProp** + *i*

where i = 0, 1, 2...

Cell index: visCustPropsFormat

Format cell (Text Fields section)

Rows

Specifies the formatting of a text field that is a string, a number, a date or time, a duration, or a currency.

Remarks

If the value of the Type cell is 0, 2, 5, 6, or 7 (string, number, date or time, duration, or currency, respectively), 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. For more information about specifying a format picture, see About format pictures.

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.

To reference the Format cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Fields.Format[i]

where i = <1>, 2, 3...

To reference the Format cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionTextField**

Row index: visRowField + i

where i = 0, 1, 2...

Cell index: visFieldFormat

Frame cell (Hyperlinks section)

See also

Represents the name of a frame to target when the application is open as an Active document in a container application. The default is an empty string.

Remarks

To reference the Frame cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Hyperlink.*Name*.Frame

where Hyperlink. Name is the row name

To reference the Frame cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionHyperlink**Row index: **visRow1stHyperlink** + *i*

where i = 0, 1, 2...

Cell index: visHLinkFrame

Gamma cell (Image Properties section)

See also Rows

Adjusts or corrects the intensity of an image for a particular output device, such as a monitor or scanner. The default value is 1 (no correction).

Remarks

To reference the Gamma cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Gamma

To reference the Gamma cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowImage
Cell index: visImageGamma

Glue cell (Layers section)

See also Rows

Specifies whether shapes belonging to the layer can be glued.

Value	Description
TRUE	Glue is enabled.
FALSE	Glue is disabled.

Remarks

This cell corresponds to the **Glue** option in the **Layer Properties** dialog box (click **Layer Properties** on the **View** menu).

To reference the Glue cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Layers.Glue[i] where i = <1>, 2, 3...

To reference the Glue cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionLayer**Row index: **visRowLayer** + *i*

where i = 0, 1, 2...

Cell index: visLayerGlue

GlueType cell (Glue Info section)

Rows

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

Value	Description	Automation constant
&H0	Default. Allow dynamic glue for	visGlueTypeDefault
	the dynamic connector only;	
	otherwise, use static glue.	
&H1	Allow dynamic glue.	Obsolete in Microsoft Visio 2002
&H2	Allow dynamic glue.	visGlueTypeWalking
&H4	Do not allow dynamic glue.	visGlueTypeNoWalking
&H8	Do not allow this 2-D shape to be	visGlueTypeNoWalkingTo
	connected to with dynamic glue.	

Remarks

If this cell contains a value of 1, 2 or 3, dynamic glue will be established when either of the following occurs:

Shapes are dynamically glued in the user interface.

Shapes are glued to the PinX or PinY cell of another shape from a program.

If shapes are glued to shape cells other than PinX or PinY from a program, then static glue is used.

Changing this value from allowing to not allowing dynamic glue does not sever

or change existing dynamic glue. Programs can establish dynamic glue even if dynamic glue is disabled in the GlueType cell.

To reference the GlueType cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: GlueType

To reference the GlueType cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowMisc**Cell index: **visGlueType**

HAlign cell (Paragraph section)

See also Rows

Determines the horizontal alignment of text in the shape's text block.

Value	Description	Automation constant
0	Left align	visHorzLeft
1	Center	visHorzCenter
2	Right align	visHorzRight
3	Justify	visHorzJustify
4	Force justify	visHorzForce

Remarks

Justify adds space between words in every line except the last line of the paragraph to align both the left and right sides of text with the margins.

Force justify justifies every line in the paragraph, including the last.

To reference the HAlign cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Para.HorzAlign[i] where i = <1>, 2, 3...

To reference the HAlign cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionParagraph**

Row index: **visRowParagraph** + i where i = 0, 1, 2...

Cell index: visHorzAlign

Height cell (Shape Transform section)

See also Rows

Determines the height of the shape in drawing units.

Remarks

To reference the Height cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Height

To reference the Height cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormHeight

HideForApply cell (Style Properties section)

See also Rows

Determines where a style is shown in the Microsoft Visio user interface.

Value	Description
TRUE	Show the style only in the Drawing Explorer and Define Styles dialog box.
FALSE	Show the style in the Drawing Explorer , Define Styles dialog box, Style dialog box, and All Styles list on the Formatting toolbar.

Remarks

You can also set this value in the **Define Styles** dialog box (click **Define Styles** on the **Format** menu).

When you base a new style on a style that is hidden, the new style does not inherit this attribute.

To reference the HideForApply cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: HideForApply

To reference the HideForApply cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowStyle
Cell index: visStyleHidden

HideText cell (Miscellaneous section)

See also Rows

Hides the text for a shape. 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 FALSE (0).

Value	Description
TRUE	Text is hidden and does not print.
FALSE	Text is not hidden.

Remarks

To reference the HideForApply cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: HideForApply

To reference the HideForApply cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowStyle
Cell index: visStyleHidden

ImgHeight cell (Foreign Image Info section)

See also Rows

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

= Height * 1

Remarks

To reference the ImgHeight cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ImgHeight

To reference the ImgHeight cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowForeign
Cell index: visFrgnImgHeight

ImgOffsetX cell (Foreign Image Info section)

See also Rows

Determines 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.

Remarks

To reference the ImgOffsetX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ImgOffsetX

To reference the ImgOffsetX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowForeign

Cell index: visFrgnImgOffsetX

ImgOffsetY cell (Foreign Image Info section)

See also Rows

Determines 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.

Remarks

To reference the ImgOffsetY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ImgOffsetY

To reference the ImgOffsetY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowForeign

Cell index: visFrgnImgOffsetY

ImgWidth cell (Foreign Image Info section)

See also Rows

Determines 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.

Remarks

To reference the ImgWidth cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ImgWidth

To reference the ImgWidth cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowForeign
Cell index: visFrgnImgWidth

IndFirst cell (Paragraph section)

See also Rows

Represents 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.

Remarks

To reference the IndFirst cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Para.IndFirst[*i*]

where i = <1>, 2, 3...

To reference the IndFirst cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionParagraph**Row index: **visRowParagraph** + *i*

where i = 0, 1, 2...

Cell index: visIndentFirst

IndLeft cell (Paragraph section)

See also Rows

Represents the distance all lines of text in a 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.

Remarks

To reference the IndLeft cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Para.IndLeft[i]

where i = <1>, 2, 3...

To reference the IndLeft cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionParagraph**Row index: **visRowParagraph** + *i*

where i = 0, 1, 2...

Cell index: visIndentLeft

IndRight cell (Paragraph section)

See also Rows

Represents the distance all lines of text in a 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.

Remarks

To reference the IndRight cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Para.IndRight[i]

where i = <1>, 2, 3...

To reference the IndRight cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionParagraph**Row index: **visRowParagraph** + *i*

where i = 0, 1, 2...

Cell index: visIndentRight

InhibitSnap cell (Page Properties section)

See also Rows

Determines whether the shapes on a foreground page snap to other objects on the page and shapes on the background page.

Value	Description
TRUE	Inhibit all snapping on the page, except for snapping to the ruler and grid.
FALSE	Enable snapping.

Remarks

To reference the InhibitSnap cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: InhibitSnap

To reference the InhibitSnap cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowPage**

Cell index: visPageInhibitSnapIndentRight

Invisible cell (Custom Properties section)

See also

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

Value	Description
TRUE	Custom property is not visible.
FALSE	Custom property is visible.

Remarks

To reference the Invisible cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Prop.*Name*.Invisible

where Prop. Name is the row name

To reference the Invisible cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionProp**Row index: **visRowProp** + *i*

where i = 0, 1, 2...

Cell index: visPageInhibitSnapIndentRight

IsDropSource cell (Miscellaneous section)

Rows

Determines whether the shape can be added to a group by dropping it onto the group.

Value	Description
TRUE	Can add the shape to a group by dropping it onto the group.
FALSE	Cannot add the shape to a group.

Remarks

You can also set this value by selecting the shape, clicking **Behavior** on the **Format** menu, and then selecting the **Add shape to groups on drop** check box.

In addition to enabling this behavior for a shape, you must also enable a group to accept shapes that are dragged into it. To do so, select the group, click **Behavior** on the **Format** menu, and then select the **Accept dropped shapes** check box. This value is stored in the IsDropTarget cell in the Group Properties section.

To reference the IsDropSource cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: IsDropSource

To reference the IsDropSource cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowMisc
Cell index: visDropSource

IsDropTarget cell (Group Properties section)

Rows

Determines whether the group allows you to add a shape to it by dropping it on the group.

Value	Description
TRUE	Can add a shape to the group by dropping it onto the group.
FALSE	Cannot drop shape onto the group.

Remarks

You can also set this value by selecting the group, clicking **Behavior** on the **Format** menu, and then selecting the **Accept dropped shapes** check box.

To add a shape to a group by dropping it on the group, you must also enable similar shape behavior. You must select the shape, click **Behavior** on the **Format** menu, and then select the **Add shape to groups on drop** check box. This value is stored in the IsDropSource cell in the Miscellaneous section.

To reference the IsDropTarget cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: IsDropTarget

To reference the IsDropTarget cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowGroup

Cell index: visGroupIsDropTarget

IsSnapTarget cell (Group Properties section)

See also Rows

Determines whether you snap to a group or shapes within the group.

Value	Description
TRUE	Enable snapping to shapes within a group.
FALSE	Snap only to the group.

Remarks

You can also set this value by selecting the group, clicking **Behavior** on the **Format** menu, and then selecting the **Snap to member shapes** check box.

To reference the IsSnapTarget cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: IsSnapTarget

To reference the IsSnapTarget cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowGroup**

Cell index: visGroupIsSnapTarget

IsTextEditTarget cell (Group Properties section)

See also Rows

Determines text assignment for a group.

Value	Description
TRUE	Text is added to the group shape.
FALSE	Text is added to the shape in the group at the top of the stacking order.

Remarks

You can also set this value by selecting the group, clicking **Behavior** on the **Format** menu, and then selecting the **Edit text of group** check box.

Groups created in versions earlier than Microsoft Visio 2000 have a default value of FALSE. Beginning with Visio version 2000, the default value is TRUE.

To reference the IsTextEditTarget cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: IsTextEditTarget

To reference the IsTextEditTarget cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowGroup**

Cell index: visGroupIsTextEditTarget

Label cell (Custom Properties section)

See also

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

Remarks

The application automatically encloses the Label string in quotation marks in the cell, but the quotation marks are not displayed in the **Custom Properties** dialog box and **Custom Properties** window.

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

To reference the Label cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Prop.*Name*.Label

where Prop. *Name* is the row name

To reference the Label cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionProp**Row index: **visRowProp** + *i*

where i = 0, 1, 2...

Cell index: visCustPropsLabel

LeftMargin cell (Text Block Format section)

See also Rows

Determines 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.

Remarks

To reference the LeftMargin cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LeftMargin

To reference the LeftMargin cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowText**

Cell index: visTxtBlkLeftMargin

LineAdjustFrom cell (Page Layout section)

See also Rows

Determines which dynamic connectors the application spaces apart if they route on top of each other.

Value	Adjustment	Automation constant
0	Unrelated lines	visPLOLineAdjustFromNotRelated
1	All lines	visPLOLineAdjustFromAll
2	No lines	visPLOLineAdjustFromNone
3	Routing style default	vis PLOLine Adjust From Routing Default

Remarks

You can also set the value of this cell on the **Layout and Routing** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

To reference the LineAdjustFrom cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineAdjustFrom

To reference the LineAdjustFrom cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout

Cell index: visPLOLineAdjustFrom

LineAdjustTo cell (Page Layout section)

See also Rows

Determines which dynamic connectors line up on top of one another.

Value	Adjustment	Automation constant
0	Routing style default	visPLOLineAdjustToDefault
1	Lines that are close to each other	visPLOLineAdjustToAll
2	No lines	visPLOLineAdjustToNone
3	Related lines	visPLOLineAdjustToRelated

Remarks

You can also set the value of this cell on the **Layout and Routing** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

To reference the LineAdjustTo cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineAdjustTo

To reference the LineAdjustTo cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOLineAdjustTo

LineCap cell (Line Format section)

See also Rows

Indicates whether a line has rounded or square line ends.

Value	Line end style
0	Rounded
1	Square

Remarks

You can also set the value of this cell in the **Line** dialog box (click **Line** on the **Format** menu).

To reference the LineCap cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineCap

To reference the LineCap cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowLine**Cell index: **visLineEndCap**

LineColor cell (Line Format section)

Rows

Determines the line color of the shape.

Remarks

To set the line color, enter a number from 0 to 23, which is an index into a collection of line colors. You can view the line color collection in the **Line** dialog box (click **Line** on the **Format** menu). You can also set the value of LineColor in the **Line** dialog box. To switch to a different color palette, use the **Color Palette** dialog box (click **Color Palette** on the **Tools** menu).

To enter a custom color, use the RGB or HSL function. The value of a custom color is its RGB color, and RGB(r, g, b), rather than a number, will be shown in the ShapeSheet window. When used in numeric operations, custom colors have values of 24 and above.

You can set the transparency of the line color in the LineColorTrans cell.

To reference the LineColor cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineColor

To reference the LineColor cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowLine
Cell index: visLineColor

LineColorTrans cell (Line Format section)

Rows

Determines the transparency level of a shape's line color.

Value	Description
0 – 100	Represents the percentage of transparency. The default is 0% (completely opaque).

Remarks

Values are rounded to the nearest half percent. A value of 100% is completely transparent. Although a shape with a completely transparent line color appears the same as a shape with no lines on the drawing page, it will interact with other objects on the page in the same ways as if its transparency is 0%.

You can also set this value using the slider control in the **Line** dialog box (click **Line** on the **Format** menu).

To reference the LineColorTrans cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineColorTrans

To reference the LineColorTrans cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLine

Cell index: visLineColorTrans

LineJumpCode cell (Page Layout section)

See also Rows

Determines the connectors to which you want to add jumps.

Value	Connectors to which you want to add jumps	Automation constant
0	None	visPLOJumpNone
1	Horizontal lines	visPLOJumpHorizontal
2	Vertical lines	visPLOJumpVertical
3	Last routed line	visPLOJumpLastRouted
4	Last displayed line (top shape in the z-order)	visPLOJumpDisplayOrder
5	First displayed line (shape at the bottom of the <i>z</i> -order)	t visPLOJumpReverseDisplayOrder

Remarks

You can also set the value of this cell on the **Layout and Routing** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

To reference the LineJumpCode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineJumpCode

To reference the LineJumpCode cell by index from a program, use the CellsSRC

property with the following arguments:

Section index: visSectionObject

Row index: visRowPageLayout

Cell index: visPLOJumpCode

LineJumpFactorX cell (Page Layout section)

See also Rows

Determines the size of line jumps on horizontal dynamic connectors on the page, relative to the value of the LineToLineX cell. The value of this cell can range from 0 to 10 but fractional values from 0 to 1 are suggested.

Remarks

You can also set the value of this cell on the **Layout and Routing** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

To reference the LineJumpFactorX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineJumpFactorX

To reference the LineJumpFactorX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOJumpFactorX

LineJumpFactorY cell (Page Layout section)

See also Rows

Determines the size of line jumps on vertical dynamic connectors on the page, relative to the value of the LineToLineY cell. The value of this cell can range from 0 to 10 but fractional values from 0 to 1 are suggested.

Remarks

You can also set the value of this cell on the **Layout and Routing** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

To reference the LineJumpFactorY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineJumpFactorY

To reference the LineJumpFactorY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOJumpFactorY

LineJumpStyle cell (Page Layout section)

See also Rows

Determines the line jump style for all connectors on the drawing page that don't have a local line jump style.

Value	Line jump style	Automation constant
0	Arc	visLOJumpStyleDefault
1	Arc	visLOJumpStyleArc
2	Gap	visLOJumpStyleGap
3	Square	visLOJumpStyleSquare
4	2 sides	visLOJumpStyleTriangle
5	3 sides	visLOJumpStyle2Point
6	4 sides	visLOJumpStyle3Point
7	5 sides	visLOJumpStyle4Point
8	6 sides	visLOJumpStyle5Point
9	7 sides	visLOJumpStyle6Point

Remarks

You can also set the value of this cell on the **Layout and Routing** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

To reference the LineJumpStyle cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineJumpStyle

To reference the LineJumpStyle cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOJumpStyle

LinePattern cell (Line Format section)

Rows

Determines the line pattern of the shape. The value entered in the LinePattern cell is a number that is an index into a collection of line patterns.

Value	Description
0	No line pattern
1	Solid
2-23	Assorted line patterns

Remarks

You can view the line pattern collection in the **Line** dialog box (click **Line** on the **Format** menu).

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

To reference the LinePattern cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LinePattern

To reference the LinePattern cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowPageLayout
Cell index: visPLOJumpStyle

LineRouteExt cell (Page Layout section)

Rows

Determines the default appearance for all connectors on a drawing page.

Value	Description	Automation constant
0	Default (straight)	visLORouteExtDefault
1	Straight	visLORouteExtStraight
2	Curved	visLORouteExtNURBS

Remarks

To reference the LineRouteExt cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineRouteExt

To reference the LineRouteExt cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowPage**

Cell index: visPLOLineRouteExt

LineToLineX cell (Page Layout section)

See also Rows

Determines the horizontal clearance between all connectors on the drawing page.

Remarks

You can also set the value of this cell in the **Layout and Routing Spacing** dialog box (on the **File** menu, click **Page Setup**, click the **Layout and Routing** tab, and then click **Spacing**).

To reference the LineToLineX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineToLineX

To reference the LineToLineX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOLineToLineX

LineToLineY cell (Page Layout section)

See also Rows

Determines the vertical clearance between all connectors on the drawing page.

Remarks

You can also set the value of this cell in the **Layout and Routing Spacing** dialog box (on the **File** menu, click **Page Setup**, click the **Layout and Routing** tab, and then clicking **Spacing**).

To reference the LineToLineY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineToLineY

To reference the LineToLineY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOLineToLineY

LineToNodeX cell (Page Layout section)

See also Rows

Determines the horizontal clearance between all connectors and shapes on the drawing page.

Remarks

You can also set the value of this cell in the **Layout and Routing Spacing** dialog box (on the **File** menu, click **Page Setup**, click the **Layout and Routing** tab, and then click **Spacing**).

To reference the LineToNodeX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineToNodeX

To reference the LineToNodeX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOLineToNodeX

LineToNodeY cell (Page Layout section)

See also Rows

Determines the vertical clearance between all connectors and shapes on the drawing page.

Remarks

You can also set the value of this cell in the **Layout and Routing Spacing** dialog box (on the **File** menu, click **Page Setup**, click the **Layout and Routing** tab, and then click **Spacing**).

To reference the LineToNodeY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineToNodeY

To reference the LineToNodeY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOLineToNodeY

LineWeight cell (Line Format section)

Rows

Determines the line weight of a shape. Set the line weight by entering a number with a valid unit of measure.

Remarks

You can also set the value of LineWeight in the **Line** dialog box (click **Line** on the **Format** menu).

If the unit of measure is not entered, then the unit of measure for text specified in the **Options** dialog box is used (click **Options** on the **Tools** menu). Line weight is independent of the scale of the drawing. If the drawing is scaled, the line weight remains the same.

To reference the LineWeight cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LineWeight

To reference the LineWeight cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLine
Cell index: visLineWeight

Lock cell (Layers section)

See also Rows

Specifies whether shapes belonging to the layer are locked against being selected or edited.

Value	Description
TRUE	Shapes are locked.
FALSE	Shapes are not locked.

Remarks

You can also set this value using the **Lock** option in the **Layer Properties** dialog box (click **Layer Properties** on the **View** menu).

To reference the Lock cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Layers.Locked[i] where i = <1>, 2, 3...

To reference the Lock cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionLayer**Row index: **visRowLayer** + iwhere i = 0, 1, 2...

Cell index: visLayerLock

LockAspect cell (Protection section)

See also Rows

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

Value	Description
TRUE	Aspect ratio is locked.
FALSE	Aspect ratio is not locked.

Remarks

To reference the LockAspect cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockAspect

To reference the LockAspect cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowLock
Cell index: visLockAspect

LockBegin cell (Protection section)

See also Rows

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

Value	Description
TRUE	Begin point is locked.
FALSE	Begin is not locked.

Remarks

To reference the LockBegin cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockBegin

To reference the LockBegin cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock
Cell index: visLockBegin

LockCalcWH cell (Protection section)

See also Rows

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.

Value	Description
TRUE	Width and height cannot be recalculated.
FALSE	Width and height can be recalculated.

Remarks

To reference the LockCalcWH cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockCalcWH

To reference the LockCalcWH cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock

Cell index: visLockCalcWH

LockCrop cell (Protection section)

See also Rows

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

Value	Description
TRUE	Shape cannot be cropped
FALSE	Shape can be cropped.

Remarks

To reference the LockCrop cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockCrop

To reference the LockCrop cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock
Cell index: visLockCrop

LockDelete cell (Protection section)

See also Rows

Locks the shape so that it cannot be deleted.

Value	Description
TRUE	Shape cannot be deleted
FALSE	Shape can be deleted.

Remarks

To reference the LockDelete cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockDelete

To reference the LockDelete cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock
Cell index: visLockDelete

LockEnd cell (Protection section)

See also Rows

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

Value	Description
TRUE	End point is locked.
FALSE	End point is not locked.

Remarks

To reference the LockEnd cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockEnd

To reference the LockEnd cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock
Cell index: visLockEnd

LockFormat cell (Protection section)

See also Rows

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

Value	Description
TRUE	Formatting cannot be changed.
FALSE	Formatting can be changed.

Remarks

To reference the LockFormat cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockFormat

To reference the LockFormat cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock

Cell index: visLockFormat

LockGroup cell (Protection section)

See also Rows

Locks a group so that it cannot be ungrouped.

Value	Description
TRUE	Group cannot be ungrouped.
FALSE	Group can be ungrouped.

Remarks

To reference the LockGroup cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockGroup

To reference the LockGroup cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowLock**

Cell index: visLockGroup

LockHeight cell (Protection section)

See also Rows

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

Value	Description
TRUE	Height is locked.
FALSE	Height is not locked.

Remarks

To reference the LockHeight cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockHeight

To reference the LockHeight cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowLock
Cell index: visLockHeight

LockMoveX cell (Protection section)

See also Rows

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

Value	Description
TRUE	Horizontal position is locked.
FALSE	Horizontal position is not locked.

Remarks

To reference the LockMoveX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockMoveX

To reference the LockMoveX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock
Cell index: visLockMoveX

LockMoveY cell (Protection section)

See also Rows

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

Value	Description
TRUE	Vertical position is locked.
FALSE	Vertical position is not locked.

Remarks

To reference the LockMoveY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockMoveY

To reference the LockMoveY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock
Cell index: visLockMoveY

LockPreview cell (Document Properties section)

See also Rows

Determines whether a preview is saved each time you save a drawing.

Value	Description
TRUE	Do not save a preview each time a drawing is saved.
FALSE	Save a preview each time a drawing is saved.

Remarks

To reference the LockPreview cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockPreview

To reference the LockPreview cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowDoc

Cell index: visDocLockPreview

LockRotate cell (Protection section)

Rows

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

Value	Description
TRUE	Shape cannot be rotated.
FALSE	Shape can be rotated.

Remarks

The LockRotate cell does not prevent a 1-D shape from being rotated when an endpoint is dragged. To lock a 1-D shape against rotation, set the LockWidth cell to a non-zero value (TRUE).

To reference the LockRotate cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockRotate

To reference the LockRotate cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowLock
Cell index: visLockRotate

LockSelect cell (Protection section)

See also Rows

Prevents a shape from being selected.

Value	Description
TRUE	Shape cannot be selected.
FALSE	Shape can be selected.

Remarks

In order for LockSelect to take effect, the **Shapes** check box must be selected in the **Protect Document** dialog box.

To reference the LockSelect cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockSelect

To reference the LockSelect cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock
Cell index: visLockSelect

LockTextEdit cell (Protection section)

See also Rows

Locks the text of a shape so it cannot be edited.

Value	Description
TRUE	Text cannot be edited.
FALSE	Text can be edited.

Remarks

You can still format text by applying a style using the **Font** style list or options in the **Text** dialog box (click **Text** on the **Format** menu).

To reference the LockTextEdit cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockTextEdit

To reference the LockTextEdit cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: **visRowLock**

Cell index: visLockTextEdir

LockVtxEdit cell (Protection section)

See also Rows

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

Value	Description
TRUE	Vertices cannot be edited.
FALSE	Vertices can be edited.

Remarks

To reference the LockVtxEdit cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockVtxEdit

To reference the LockVtxEdit cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock

Cell index: visLockVtxEdit

LockWidth cell (Protection section)

See also Rows

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

Value	Description
TRUE	Width is locked.
FALSE	Width is not locked.

Remarks

To reference the LockWidth cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LockWidth

To reference the LockWidth cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLock
Cell index: visLockWidth

LocPinX cell (Shape Transform section)

See also Rows

Represents 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

Remarks

To reference the LocPinX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LocPinX

To reference the LocPinX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormLocPinX

LocPinY cell (Shape Transform section)

See also Rows

Represents 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

Remarks

To reference the LocPinY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: LocPinY

To reference the LocPinY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormLocPinY

Menu cell (Actions section)

See also Rows

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.

Remarks

To reference the Menu cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Actions.Menu[i]

where i = <1>, 2, 3, ...

To reference the Menu cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionAction

Row index: $\mathbf{visRowAction} + i$

where i = 0, 1, 2, ...

Cell index: visActionMenu

NewWindow cell (Hyperlinks section)

See also

Specifies whether to open the hyperlink in a new window.

Value	Description
TRUE	Open the linked page, document, or Web site in a new window.
FALSE	Default. Do not open a new window for the hyperlink.

Remarks

To reference the NewWindow cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Hyperlink.*Name*.NewWindow

where Hyperlink. Name is the row name

To reference the NewWindow cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionHyperlink**Row index: **visRow1stHyperlink** + *i*

where i = 0, 1, 2, ...

Cell index: visHLinkNewWin

NoAlignBox cell (Miscellaneous section)

See also Rows

Toggles the display of the selection rectangle on and off for the selected shape.

Value	Description
TRUE	Selection rectangle is not displayed when the shape is selected.
FALSE	Selection rectangle is displayed when the shape is selected.

Remarks

To reference the NoAlignBox cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: NoAlignBox

To reference the NoAlignBox cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowMisc**Cell index: **visNoAlignBox**

NoCtlHandles cell (Miscellaneous section)

See also Rows

Toggles the display of control handles on and off for the selected shape.

Value	Description
TRUE	Control handles are not displayed when a shape is selected.
FALSE	Control handles are displayed when a shape is selected.

Remarks

To reference the NoCtlHandles cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: NoCtlHandles

To reference the NoCtlHandles cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowMisc

Cell index: visNoCtlHandles

NoFill cell (Geometry section)

See also Rows

Indicates whether a path can be filled.

Value	Description
TRUE	The path is not filled even if other paths in the shape are filled.
FALSE	The shape's fill applies to the path, even if it isn't closed.

Remarks

If you set a shape's fill pattern to none (0), none of its paths are filled. This cell is used to turn the fill off selectively for a path within a shape.

To reference the NoFill cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometry *i*. No Fill where i = <1>, 2, 3...

To reference the NoFill cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionFirstComponent + i

where i = 0, 1, 2...

Row index: **visRowComponent**

Cell index: visCompNoFill

NoLine cell (Geometry section)

See also Rows

Determines whether a line is drawn around the boundary of the path.

Value	Description
TRUE	A line is not drawn around the boundary of the path that is the
	boundary of a filled region.
FALSE	A line is drawn around the boundary of a path.

Remarks

When you change the color of a line to white, the line still exists even though you can't see it on a white background. When you set the value of this cell to TRUE, no line is drawn.

To reference the NoLine cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometry *i*. No Line where i = <1>, 2, 3...

To reference the NoLine cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: visRowComponent
Cell index: visCompNoLine

NoLiveDynamics cell (Miscellaneous section)

Rows

Determines whether a shape dynamically resizes or rotates as you are manipulating it.

Value	Description
TRUE	Do not dynamically update the shape while you are manipulating
	it.
FALSE	Dynamically update the shape while you are manipulating it.

Remarks

As you resize or rotate a two-dimensional (2-D) shape without live dynamics, you see a selection box. If the shape is one-dimensional (1-D), the visual feedback is based on the value of the DynFeedback cell.

To reference the NoLiveDynamics cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: NoLiveDynamics

To reference the NoLiveDynamics cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowMisc**

Cell index: visNoLiveDynamics

NonPrinting cell (Miscellaneous section)

See also Rows

Toggles printing on and off for the selected shape.

Value	Description
TRUE	Printing disabled, but the shape will be displayed in the drawing
	window.
FALSE	Printing enabled.

Remarks

You can print a guide by selecting it, and then setting the value of its NonPrinting cell to FALSE.

To reference the NonPrinting cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: NonPrinting

To reference the NonPrinting cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowMisc**Cell index: **visNonPrinting**

NoObjHandles cell (Miscellaneous section)

See also Rows

Toggles the display of selection handles on and off for the selected shape.

Value	Description
TRUE	Selection handles are not displayed when a shape is selected.
FALSE	Selection handles are displayed when a shape is selected.

Remarks

To reference the NoObjHandles cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: NoObjHandles

To reference the NoObjHandles cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowMisc

Cell index: visNoObjHandles

NoShow cell (Geometry section)

See also Rows

Indicates whether a path is displayed on the drawing page.

Value	Description
TRUE	The stroke and fill of the path represented by the section is hidden.
FALSE	The stroke and fill of the path is shown.

Remarks

To reference the NoShow cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometry *i*. No Show where i = <1>, 2, 3...

To reference the NoShow cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: **visRowComponent**Cell index: **visCompNoShow**

NoSnap cell (Geometry section)

See also Rows

Determines whether other shapes snap to a path.

Value	Description
TRUE	Do not allow other shapes to snap to this path.
FALSE	Allow other shapes to snap to this path.

Remarks

To reference the NoSnap cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometry *i*. No Snap where i = <1>, 2, 3...

To reference the NoSnap cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: **visRowComponent**Cell index: **visCompNoSnap**

ObjType cell (Miscellaneous section)

See also Rows

Determines whether objects are placeable or routable in diagrams when laying out shapes using the **Lay Out Shapes** command.

Value	Description	Automation constant
&H0	Default. The application decides based on the drawing context.	visLOFlagsVisDecides
&H1	Shape is placeable.	visLOFlagsPlacable
&H2	Shape is routable. Must be a one-dimensional (1-D) shape.	visLOFlagsRoutable
&H4	Shape is not placeable, not routable.	visLOFlagsDont
&H8	Group contains placeable/routable shapes.	e visLOFlagsPNRGroup

Remarks

By default, the ObjType cell is set to No Formula for a shape, which evaluates to 0, meaning that the application 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 **Connector** tool to connect the rectangle to another shape, Microsoft Visio resets the value of the rectangle's ObjType cell to 1 (placeable).

The value of the ObjType cell can be a combination of values. If the non-placeable bit is set (&H4), however, it takes precedence over other values except the group value (&H8).

To reference the ObjType cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ObjType

To reference the ObjType cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowMisc**Cell index: **visLOFlags**

OutputFormat cell (Document Properties section)

See also Rows

Determines the output format for a drawing. Drawing pages are usually formatted for printing (default); however, you can choose other output formats.

Value	Output format
0	Printing (default)
1	PowerPoint slide show
2	HTML or GIF output

Remarks

To reference the OutputFormat cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: OutputFormat

To reference the OutputFormat cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: **visRowDoc**

Cell index: visDocOutputFormat

Overline cell (Character section)

See also Rows

Determines whether the text has a line above it.

Value	Description
TRUE	Text has a line above it.
FALSE	Text does not have a line above it.

Remarks

You can also set the value of this cell by using the **Text** dialog box (click **Text** on the **Format** menu).

To reference the Overline cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Overline[i] where i = <1>, 2. 3...

To reference the Overline cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2...

Cell index: visCharacterOverline

PageHeight cell (Page Properties section)

See also Rows

Contains the height of the printed page in drawing units.

Remarks

You can also set the page height in the **Page Size** tab of the **Page Setup** dialog box (click **Page Setup** on the **File** menu), or by manually resizing the page with the mouse.

To reference the PageHeight cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PageHeight

To reference the PageHeight cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowPage**Cell index: **visPageHeight**

PageLineJumpDirX cell (Page Layout section)

See also Rows

Determines the direction of line jumps on horizontal dynamic connectors on the drawing page for which you haven't applied a local jump direction.

Value	Line jump direction	Automation constant
0	Default; left or the page's setting for shapes	visLOJumpDirXDefault
1	Up	visLOJumpDirXUp
2	Down	visLOJumpDirXDown

Remarks

To reference the PageLineJumpDirX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PageLineJumpDirX

To reference the PageLineJumpDirX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOJumpDirX

PageLineJumpDirY cell (Page Layout section)

See also Rows

Determines the direction of line jumps on vertical dynamic connectors on the drawing page for which you haven't applied a local jump direction.

Value	Line jump direction	Automation constant
0	Default; up or the page's setting for shapes	visLOJumpDirYDefault
1	Left	visLOJumpDirYLeft
2	Right	visLOJumpDirYRight

Remarks

To reference the PageLineJumpDirY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PageLineJumpDirY

To reference the PageLineJumpDirY cell by index from a program, use the CellsSRC property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOJumpDirY

PageScale cell (Page Properties section)

Rows

Determines 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.

Remarks

You can also set the value of the PageScale cell on the **Drawing Scale** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu). The value of the cell is the first of the two numbers in the **Pre-defined scale** box or **Custom scale** box, depending on the drawing scale setting selected under **Drawing scale**. For example, if you select an architectural scale for your drawing, the drawing scale for the page is 3/32" = 1'0". The value in the PageScale cell is 0.0938 in. (or 3/32") and the value in the DrawingScale cell is 1 ft.

To reference the PageScale cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PageScale

To reference the PageScale cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowPage

Cell index: visPageScale

PageWidth cell (Page Properties section)

See also Rows

Determines the width of the printed page in drawing units.

Remarks

You can also set the page width on the **Page Size** tab of the **Page Setup** dialog box (click **Page Setup** on the **File** menu) or by manually resizing the page with the mouse. To do this, drag the edge of the page while holding down the CTRL key.

To reference the PageWidth cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PageWidth

To reference the PageWidth cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowPage**Cell index: **visPageWidth**

Perpendicular cell (Character section)

See also Rows

Determines whether a text field appears perpendicular to the other text in a text block.

Value	Description
TRUE	Text field appears perpendicular to the other text in a text block.
FALSE	Text field does not appear perpendicular to the other text in a text block.

Remarks

This cell is used to implement Kumimoji in Japanese text.

To reference the Perpendicular cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Perpendicular[i] where i = <1>, 2, 3...

To reference the Perpendicular cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + iwhere i = 0, 1, 2...

Cell index: visCharacterPerpendicular

PlaceDepth cell (Page Layout section)

See also Rows

Determines the method by which the drawing is analyzed before creating the layout, and determines the type of layout.

Value	Placement depth for vertical and horizontal layouts	Automation constant
0	Page default	visPLOPlaceDepthDefault
1	Medium	visPLOPlaceDepthMedium
2	Deep	visPLOPlaceDepthDeep
3	Shallow	visPLOPlaceDepthShallow

Remarks

You can also set the value of this cell by using the **Lay Out Shapes** dialog box (click **Lay Out Shapes** on the **Shape** menu).

To reference the PlaceDepth cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PlaceDepth

To reference the PlaceDepth cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOPlaceDepth

PlaceFlip cell (Page Layout section)

Rows

Determines how placeable shapes flip and/or rotate on a page when you use the **Lay Out Shapes** command.

Value	Description	Automation constant
&H0	Default. Do not flip.	visLOFlipDefault
&H1	Flip horizontal.	visLOFlipX
&H2	Flip vertical.	visLOFlipY
&H4	Flip in 90 degree	visLOFlipRotate
	increments.	
&H8	Do not flip.	visLOFlipNone

Remarks

The value in the PlaceFlip cell helps orient a placeable shape toward the next placeable shape it is connected to. It is typically used when laying out drawings that use static glue.

To set this behavior for a particular shape, use the ShapePlaceFlip cell in the Shape Layout section.

To reference the PlaceFlip cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PlaceFlip

To reference the PlaceFlip cell by index from a program, use the CellsSRC

property with the following arguments:

Section index: visSectionObject

Row index: visRowPageLayout

Cell index: visPLOPlaceFlip

PlaceStyle cell (Page Layout section)

See also Rows

Determines how shapes are placed on the page when you are laying out shapes using the **Lay Out Shapes** command.

Value	Placement style	Automation constant
0	Default; radial for page and shape	visPLOPlaceDefault
1	Top to bottom	visPLOPlaceTopToBottom
2	Left to right	visPLOPlaceLeftToRight
3	Radial	visPLOPlaceRadial
4	Bottom to top	visPLOPlaceBottomToTop
5	Right to left	${f visPLOPlaceRightToLeft}$
6	Circular	visPLOPlaceCircular

Remarks

You can also set the value of this cell in the **Lay Out Shapes** dialog box (click **Lay Out Shapes** on the **Shape** menu).

To reference the PlaceStyle cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PlaceStyle

To reference the PlaceStyle cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOPlaceStyle

PinX cell (Shape Transform section)

See also Rows

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

Remarks

To reference the PinX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PinX

To reference the PinX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormPinX

PinY cell (Shape Transform section)

See also Rows

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

Remarks

To reference the PinY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PinY

To reference the PinY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormPinY

PlowCode cell (Page Layout section)

See also Rows

Determines whether placeable shapes move away when you drop a placeable shape near another placeable shape on the drawing page.

Value	Description	Automation constant
0	Don't move shapes	visPLOPlowNone
1	Move shapes	visPLOPlowAll

Remarks

You can also set the value of this cell on the **Layout and Routing** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu) using the **Move other shapes away on drop** check box.

To reference the PlowCode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PlowCode

To reference the PlowCode cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOPlowCode

Pos cell (Character section)

See also Rows

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

Value	Description	Automation constant
0	Normal position	visPosNormal
1	Superscript	visPosSuper
2	Subscript	visPosSub

Remarks

To reference the Pos cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Pos[i] where i = <1>, 2, 3...

To reference the Pos cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + iwhere i = 0, 1, 2...Cell index: **visCharacterPos**

Position cell (Tabs section)

See also Rows

Specifies the position of a tab stop. The tab position is independent of the scale of the drawing. If the drawing is scaled, the tab position remains the same.

Remarks

To reference the Position cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Tabs.*ij*

where *i* and j = <1>, 2, 3...

To reference the Position cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionTab**Row index: **visRowTab** + *i*

where i = 0, 1, 2...

Cell index: (j*3) + visTabPos

PreviewQuality cell (Document Properties section)

See also Rows

Determines whether the drawing preview is draft quality or detailed.

Value	Preview quality	Automation constant
0	Draft	visDocPreviewQualityDraft
1	Detailed	visDocPreviewQualityDetailed

Remarks

You can also set this value on the **Summary** tab in the **Properties** dialog box (click **Properties** on the **File** menu).

To reference the PreviewQuality cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PreviewQuality

To reference the PreviewQuality cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObj**Row index: **visRowDoc**

Cell index: visDocPreviewQuality

PreviewScope cell (Document Properties section)

See also Rows

Determines whether your drawing includes a preview. If your drawing does include a preview, it determines whether the preview shows the first page only or all of the pages in the drawing.

Value	Preview scope	Automation constant
0	First page	visDocPreviewScope1stPage
1	None	visDocPreviewScopeNone
2	All pages	visDocPreviewScopeAllPages

Remarks

You can also set this value on the **Summary** tab in the **Properties** dialog box (click **Properties** on the **File** menu).

To reference the PreviewScope cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: PreviewScope

To reference the PreviewScope cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObj**Row index: **visRowDoc**

Cell index: visDocPreviewScope

Print cell (Layers section)

See also Rows

Specifies whether shapes belonging to the layer can be printed.

Value	Description
TRUE	Shapes can be printed.
FALSE	Shapes cannot be printed.

Remarks

You can also set this value using the **Print** option in the **Layer Properties** dialog box (click **Layer Properties** on the **View** menu).

To reference the Print cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Layers.Print[i]

where i = <1>, 2, 3...

To reference the Print cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionLayer**Row index: **visRowLayer** + *i*

where i = 0, 1, 2...

Cell index: visDocPreviewScope

Prompt cell (Actions section)

See also Rows

Beginning with Microsoft Visio 2002, this cell is no longer used.

Remarks

In earlier versions of Visio, this cell specified a descriptive prompt that appeared in the status bar when you selected the corresponding shortcut command.

Prompt cell (Custom Properties section)

See also

Specifies descriptive or instructional text that appears to users in the **Custom Properties** box when the property is selected. This text also appears as a tip when the mouse is paused over the property in the **Custom Properties** window.

Remarks

To reference the Prompt cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Prop.Name.Prompt

where *Name* is the row name

To reference the Prompt cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionProp**Row index: **visRowProp** + iwhere i = 0, 1, 2...

Cell index: visCustPropsPrompt

Prompt cell (User-Defined Cells section)

See also

Specifies a descriptive prompt or comment for the user-defined cell. The application 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 the application evaluates.

Remarks

To reference the Prompt cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: User.*Name*.Prompt

where User. Name is the row name

To reference the Prompt cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionUser**

Row index: visRowUser + i

where i = 0, 1, 2...

Cell index: visUserPrompt

ResizeMode cell (Shape Transform section)

See also Rows

Shows the current resize behavior setting for the shape.

Value	Description	Automation constant
0	Use group's setting.	visXFormResizeDontCare
1	Reposition only.	visXFormResizeSpread
2	Scale with group.	visXFormResizeScale

Remarks

You can also set this value on the **Behavior** tab in the **Behavior** dialog box (click **Behavior** on the **Format** menu). To reference the ResizeMode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ResizeMode

To reference the ResizeMode cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormResizeMode

ResizePage cell (Page Layout section)

See also Rows

Determines whether to enlarge the page to enclose the drawing after laying out shapes using the **Lay Out Shapes** command (click **Lay Out Shapes** on the **Shape** menu).

Value	Description
TRUE	Enlarge the page.
FALSE	Do not enlarge the page.

Remarks

To reference the ResizePage cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ResizePage

To reference the ResizePage cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLOResizePage

RightMargin cell (Text Block Format section)

See also Rows

Determines the distance between the right border of the text block and the text it contains. The default is 0.1 inch.

Remarks

This value is independent of the scale of the drawing. If the drawing is scaled, the right margin remains the same.

To reference the RightMargin cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: RightMargin

To reference the RightMargin cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowText**

Cell index: visTxtBlkRightMargin

Rounding cell (Line Format section)

See also Rows

Indicates the radius of the rounding arc applied where two contiguous segments of a path meet. For example, rounding can be used to give a rectangle rounded corners. To set rounding, enter a value with units of measure (a number-unit pair).

Remarks

You can also set this value in the **Line** dialog box (click **Line** on the **Format** menu).

To reference the Rounding cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Rounding

To reference the Rounding cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowLine

Cell index: visLineRounding

RouteStyle cell (Page Layout section)

Rows

Determines the routing style and direction for all connectors on the drawing page that don't have a local routing style.

Value	Routing style	Direction	Automation constant
0	Default; right angle	None	visLORouteDefault
1	Right angle	None	vis LOR out e Right Angle
2	Straight	None	visLORouteStraight
3	Organization chart	Top to bottom	visLORouteOrgChartNS
4	Organization chart	Left to right	visLORouteOrgChartWE
5	Flowchart	Top to bottom	vis LOR outeFlow chart NS
6	Flowchart	Left to right	${\bf vis LOR oute Flow chart WE}$
7	Tree	Top to bottom	${\bf vis LORoute Tree NS}$
8	Tree	Left to right	${\bf vis LORoute Tree WE}$
9	Network	None	visLORouteNetwork
10	Organization chart	Bottom to top	visLORouteOrgChartSN
11	Organization chart	Right to left	${\bf vis LORoute Org Chart EW}$
12	Flowchart	Bottom to top	${\bf vis LORoute Flow chart SN}$
13	Flowchart	Right to left	vis LOR outeFlow chart EW
14	Tree	Bottom to top	visLORouteTreeSN
15	Tree	Right to left	${\bf vis LORoute Tree EW}$
16	Center to center	None	vis LOR out e Center To Center
17	Simple	Top to bottom	visLORouteSimpleNS
18	Simple	Left to right	visLORouteSimpleWE

visLORouteSimpleSN	Bottom to top	Simple	19
vis LOR oute Simple EW	Right to left	Simple	20
visLORouteSimpleHV	None	Simple horizontal- vertical	21
visLORouteSimpleVH	None	Simple vertical- horizontal	22

Remarks

You can also set the value of this cell on the **Layout and Routing** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

You can set a local routing style for a connector in the ShapeRouteStyle cell of the Shape Layout section.

To reference the RouteStyle cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: RouteStyle

To reference the RouteStyle cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowPageLayout
Cell index: visPLORouteStyle

Scale cell (Character section)

See also Rows

Controls the font width. The default value for this cell is 100%.

Remarks

Set the percentage between 1% and 99% to decrease the font width. Set it between 101% and 600% to increase the font width.

You can also set the value of this cell by using the **Text** dialog box (click **Text** on the **Format** menu).

To reference the Scale cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.FontScale[*i*]

where i = <1>, 2, 3...

To reference the Scale cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2...

Cell index: visCharacterFontScale

SelectMode cell (Group Properties section)

See also Rows

Determines how you select a group shape and its members.

Value	Selection mode	Automation constant
0	Select the group shape only.	visGrpSelModeGroupOnly
1	Select the group shape first.	visGrpSelModeGroup1st
2	Select the members of the	visGrpSelModeMembers1st
	group first.	

Remarks

You can also set this value in the **Behavior** dialog box (with the group shape selected, click **Behavior** on the **Format** menu, and then click a mode in the **Selection** list under **Group Behavior**).

To reference the SelectMode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: SelectMode

To reference the SelectMode cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowGroup**

Cell index: visGroupSelectMode

ShapeFixedCode cell (Shape Layout section)

Rows

Specifies placement behavior for a placeable shape.

Value	Selection mode	Automation constant
&H1	Don't move this shape when shapes are laid out using the Lay Out Shapes command.	visSLOFixedPlacement
&H2	Don't move this shape and do not allow shapes that plow to be placed on top of it.	t visSLOFixedPlow
&H4	Don't move this shape and allow shapes that plow to be placed on top of it.	visSLOFixedPermeablePlow
&H20 (32)	Ignore connection point locations when being routed to.	s visSLOFixedConnPysIgnore
&H40 (64)	Only allow routing to sides with connection points.	visSLOFixedConnPtsOnly
&H80 (128)	Don't glue to the perimeter of this shape. Glue to the shape's alignment box instead.	vis SLOFixed NoFold To Shape

Remarks

You can also set the value of this cell on the **Placement** tab in the **Behavior**

dialog box (with the shape selected, click **Behavior** on the **Format** menu).

You can set any combination of these values for this cell. For example, you can enter the value 3 (&H3), which eliminates movement when you lay out shapes using the **Lay Out Shapes** command and when other placeable shapes are placed on or near the shape.

In versions earlier than Microsoft Visio 2000, you set this behavior using the ObjInteract cell in the Miscellaneous section.

To reference the ShapeFixedCode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShapeFixedCode

To reference the ShapeFixedCode cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowShapeLayout
Cell index: visSLOFixedCode

ShapePermeablePlace cell (Shape Layout section)

See also Rows

Determines whether placeable shapes can be placed on top of a shape when laying out shapes using the **Lay Out Shapes** command.

Value	Description
TRUE	Enable shapes to be placed on top of a shape.
FALSE	Do not enable shapes to be placed on top of a shape.

Remarks

You can also set the value of this cell on the **Placement** tab in the **Behavior** dialog box (with a shape selected, click **Behavior** on the **Format** menu, and then click the **Placement** tab).

In versions earlier than Microsoft Visio 2000, you set this behavior using the ObjInteract cell in the Miscellaneous section.

To reference the ShapePermeablePlace cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShapePermeablePlace

To reference the ShapePermeablePlace cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowShapeLayout
Cell index: visSLOPermeablePlace

ShapePermeableX cell (Shape Layout section)

See also Rows

Determines whether a connector can route horizontally through a placeable shape.

Value	Description
TRUE	Enable connectors to route horizontally through a placeable shape.
FALSE	Do not let connectors route horizontally through a placeable shape.

Remarks

You can also set the value of this cell on the **Placement** tab in the **Behavior** dialog box (with a shape selected, click **Behavior** on the **Format** menu, and then click the **Placement** tab).

In versions earlier than Microsoft Visio 2000, you set this behavior using the ObjInteract cell in the Miscellaneous section.

To reference the ShapePermeableX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShapePermeableX

To reference the ShapePermeableX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowShapeLayout

Cell index: visSLOPermX

ShapePermeableY cell (Shape Layout section)

See also Rows

Determines whether a connector can route vertically through a shape.

Value	Description
TRUE	Enable connectors to route vertically through a shape.
FALSE	Do not let connectors route vertically through a shape.

Remarks

You can also set the value of this cell on the **Placement** tab in the **Behavior** dialog box (with a shape selected, click **Behavior** on the **Format** menu, and then click the **Placement** tab).

In versions earlier than Microsoft Visio 2000, you set this behavior using the ObjInteract cell in the Miscellaneous section.

To reference the ShapePermeableY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShapePermeableY

To reference the ShapePermeableY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowShapeLayout**

Cell index: visSLOPermY

ShapePlaceFlip cell (Shape Layout section)

Rows

Determines how a placeable shape flips and/or rotates on the page when you are laying out shapes using the **Lay Out Shapes** command.

Value	Description	Automation constant
0	Use page default.	visLOFlipDefault
1	Flip horizontal.	visLOFlipX
2	Flip vertical.	visLOFlipY
4	Flip in 90 degree increments between 0 and 270.	visLOFlipRotate
8	Do not flip.	visLOFlipNone

Remarks

The value in the ShapePlaceFlip cell helps orient a placeable shape toward the next placeable shape it is connected to.

To set this behavior for *all* the shapes on the drawing page, use the PlaceFlip cell in the Page Layout section.

To reference the ShapePlaceFlip cell by name from another formula, or from a program using the <u>Cells</u>, use:

Cell name: ShapePlaceFlip

To reference the ShapePlaceFlip cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowShapeLayout

Cell index: visSLOPlaceFlip

ShapePlowCode cell (Shape Layout section)

Rows

Determines whether this placeable shape moves away when you drop another placeable shape near this shape on the drawing page.

Value	Description	Automation constant
0	Plow as page specifies.	visSLOPlowDefault
1	Plow no shapes.	visSLOPlowNever
2	Plow every shape.	visSLOPlowAlways

Remarks

You can also set the value of this cell for a particular shape on the **Placement** tab in the **Behavior** dialog box (with a shape selected, click **Behavior** on the **Format** menu, and then click the **Placement** tab).

To set this behavior for *all* the shapes on the drawing page, use the PlowCode cell in the Page Layout section.

To reference the ShapePlowCode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShapePlowCode

To reference the ShapePlowCode cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowShapeLayout

Cell index: visSLOPlowCode

ShapeRouteStyle cell (Shape Layout section)

Rows

Determines the routing style and direction for a selected connector on the drawing page.

Value	Routing style	Direction	Automation constant
0	Use page default	None	visLORouteDefault
1	Right angle	None	visLORouteRightAngle
2	Straight	None	visLORouteStraight
3	Organization chart	Top to bottom	visLORouteOrgChartNS
4	Organization chart	Left to right	${\bf vis LORoute Org Chart WE}$
5	Flowchart	Top to bottom	vis LOR out eFlow chart NS
6	Flowchart	Left to right	${\bf vis LOR oute Flow chart WE}$
7	Tree	Top to bottom	visLORouteTreeNS
8	Tree	Left to right	${\bf vis LORoute Tree WE}$
9	Network	None	visLORouteNetwork
10	Organization chart	Bottom to top	visLORouteOrgChartSN
11	Organization chart	Right to left	visLORouteOrgChartEW
12	Flowchart	Bottom to top	${\bf vis LOR oute Flow chart SN}$
13	Flowchart	Right to left	${\bf vis LOR oute Flow chart EW}$
14	Tree	Bottom to top	${\bf vis LORoute Tree SN}$
15	Tree	Right to left	${\bf vis LORoute Tree EW}$
16	Center to center	None	vis LOR out e Center To Center

17	Simple	Top to bottom	visLORouteSimpleNS
18	Simple	Left to right	${\bf vis LOR oute Simple WE}$
19	Simple	Bottom to top	visLORouteSimpleSN
20	Simple	Right to left	vis LOR oute Simple EW
21	Simple horizontal- vertical	None	visLORouteSimpleHV
22	Simple vertical- horizontal	None	visLORouteSimpleVH

Remarks

You can also set the value of this cell for a particular connector on the **Connector** tab in the **Behavior** dialog box (with a connector selected, click **Behavior** on the **Format** menu, and then click the **Connector** tab).

To set this behavior for *all* the connectors on a page, use the RouteStyle cell in the Page Layout section.

In versions earlier than Microsoft Visio 2000, you set this behavior using the ObjBehavior cell in the Miscellaneous section.

To reference the ShapeRouteStyle cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShapeRouteStyle

To reference the ShapeRouteStyle cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: **visRowShapeLayout**Cell index: **visSLORouteStyle**

Sharpen cell (Image Properties section)

See also Rows

Sharpens a bitmap image. The default value is 0%. Sharpening an image focuses it by increasing the contrast of adjacent pixels.

Remarks

To reference the Sharpen cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Sharpen

To reference the Sharpen cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowImage
Cell index: visImageSharpen

ShdwBkgnd cell (Fill Format section)

Rows

Determines the color used for the background (fill) of the shape's drop shadow fill pattern.

Remarks

To set the color, enter a number from 0 to 23, which is an index into a collection of colors. You can view the collection in the **Color Palette** dialog box (click **Color Palette** on the **Tools** menu).

To enter a custom color, use the RGB or HSL function. The value of a custom color is its RGB color, and RGB(r, g, b), rather than a number, will be shown in the ShapeSheet window. When used in numeric operations, custom colors have values of 24 and above.

You can set the transparency of the background color of the shape's drop shadow fill pattern in the ShdwBkgndTrans cell.

To reference the ShdwBkgnd cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShdwBkgnd

To reference the ShdwBkgnd cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowFill

Cell index: visFillShdwBkgnd

ShdwBkgndTrans cell (Fill Format section)

Determines the transparency level for the color used for the background (fill) of the shape's drop shadow fill pattern.

Value	Description
0 – 100	Represents the percentage of transparency. The default is 0% (completely opaque).

Remarks

Values are rounded to the nearest half percent. A value of 100% is completely transparent. Although a shadow with a completely transparent fill appears the same as a shadow with no fill on the drawing page, it will interact with other objects on the page in the same ways as if its transparency is 0%.

You can also set this value using the slider control in the **Shadow** dialog box (click **Shadow** on the **Format** menu). This value controls the value of both the background and foreground shadow transparencies. To set these values independently, you must enter them in the ShapeSheet window.

To reference the ShdwBkgndTrans cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShdwBkgndTrans

To reference the ShdwBkgndTrans cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowFill**

Cell index: visFillShdwBkgndTrans

ShdwForegnd cell (Fill Format section)

Rows

Determines the color used for the foreground (stroke) of the shape's drop shadow fill pattern.

Remarks

To set the color, enter a number from 0 to 23, which is an index into a collection of colors. You can view the collection in the **Color Palette** dialog box (click **Color Palette** on the **Tools** menu).

To enter a custom color, use the RGB or HSL function. The value of a custom color is its RGB color, and RGB(r, g, b), rather than a number, will be shown in the ShapeSheet window. When used in numeric operations, custom colors have values of 24 and above.

You can set the transparency of the foreground color of the shape's drop shadow fill pattern in the ShdwForegndTrans cell.

To reference the ShdwForegnd cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShdwForegnd

To reference the ShdwForegnd cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowFill

Cell index: visFillShdwForegnd

ShdwForegndTrans cell (Fill Format section)

Determines the transparency level for the color used for the foreground (stroke) of the shape's drop shadow fill pattern.

Value	Description
0 - 100	Represents the percentage of transparency. The default is 0% (completely opaque).

Remarks

Values are rounded to the nearest half percent. A value of 100% is completely transparent. Although a shadow with a completely transparent fill appears the same as a shadow with no fill on the drawing page, it will interact with other objects on the page in the same ways as if its transparency is 0%.

You can also set this value using the slider control in the **Shadow** dialog box (click **Shadow** on the **Format** menu). This value controls the value of both the background and foreground shadow transparencies. To set these values independently, you must enter them in the ShapeSheet window.

To reference the ShdwForegndTrans cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShdwForegndTrans

To reference the ShdwForegndTrans cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowFill

Cell index: visFillShdwForegndTrans

ShdwOffsetX cell (Page Properties section)

See also Rows

Determines the distance in page units that a shape's drop shadow is offset horizontally from the shape.

Remarks

This value is set in the **Page Setup** dialog box (click **Page Setup** on the **File** menu). This value is independent of the scale of the drawing. If the drawing is scaled, the shadow offset remains the same.

To reference the ShdwOffsetX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShdwOffsetX

To reference the ShdwOffsetX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowPage

Cell index: visPageShdwOffsetX

ShdwOffsetY cell (Page Properties section)

See also Rows

Determines the distance in page units that a shape's drop shadow is offset vertically from the shape.

Remarks

This value is set in the **Page Setup** dialog box (click **Page Setup** on the **File** menu). This value is independent of the scale of the drawing. If the drawing is scaled, the shadow offset remains the same.

To reference the ShdwOffsetY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShdwOffsetY

To reference the ShdwOffsetY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowPage

Cell index: visPageShdwOffsetY

ShdwPattern cell (Fill Format section)

See also Rows

Determines the fill pattern for a shape's shadow.

Value	Description
0	None (transparent fill)
1	Solid foreground color
2 - 40	Assorted patterns

Remarks

To set the fill pattern, enter a number from 0 to 40, which is an index into a collection of patterns. You can view the fill pattern collection in the **Fill** dialog box (click **Fill** on the **Format** menu).

To reference the ShdwPattern cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: ShdwPattern

To reference the ShdwPattern cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**

Row index: visRowFill

Cell index: visFillShdwPattern

Size cell (Character section)

See also Rows

Determines the size of the text in the shape's text block.

Remarks

The text's size is independent of the scale of the drawing. If the drawing is scaled, the text size remains the same.

To reference the Size cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Size[i]

where i = <1>, 2, 3...

To reference the Size cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2...

Cell index: visCharacterSize

Snap cell (Layers section)

See also Rows

Determines whether other shapes can snap to shapes assigned to the layer. Shapes assigned to the layer can snap to other shapes, but other shapes can't snap to them.

Value	Description
TRUE	Other shapes can snap to shapes on the layer.
FALSE	Other shapes cannot snap to shapes on the layer.

Remarks

You can also set the value of this cell using the **Snap** option in the **Layer Properties** dialog box (click **Layer Properties** on the **View** menu).

To reference the Snap cell by name from another formula, or from a program using the **Cells** property, use:

Layers.Snap[*i*] Cell name: where i = <1>, 2, 3...

To reference the Snap cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionLayer Row index: visRowLayer + i

where i = 0, 1, 2...

Cell index: visLayerSnap

SortKey cell (Custom Properties section)

See also

Evaluates to a string that influences the order in which items in the **Custom Properties** dialog box and window are listed.

Remarks

The calculation used to compare SortKey values is locale specific and case insensitive. If SortKey values are equal, the custom properties are listed in row order. A custom property with no sort key is listed before custom properties that contain a sort key.

Following is an example of using sort keys to display the custom properties in the Custom Properties dialog box and window in the order: Item Number, Quantity, Price.

Row, Label, and **SortKey** refer to cells in the custom property row. In this case, the custom property rows have been named.

Row	Label	SortKey
110 11	Lubei	SOLUTION

Prop.Item	Item Number	1
Prop.Price	Price	3
Prop.Quan	Quantity	2

To reference the SortKey cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Prop.Name.SortKey

where Prop. *Name* is the name of the custom property

row

To reference the SortKey cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionPropRow index: visRowProp + i

where i = 0, 1, 2...

Cell index: visCustPropsSortKey

Spacing cell (Character section)

See also Rows

Controls the amount of space between two or more characters. Space can be added or subtracted in 1/20th point increments.

Remarks

You can also set the value of this cell by using the **Text** dialog box (click **Text** on the **Format** menu).

To reference the Spacing cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Letterspace[*i*]

where i = <1>, 2, 3...

To reference the Spacing cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2...

Cell index: visCharacterspace

SpAfter cell (Paragraph section)

Rows

Determines the amount of space inserted after each paragraph in the shape's text block, in addition to any space from the SpLine cell and, if it is the last paragraph in a text block, the BottomMargin cell.

Remarks

This value is independent of the scale of the drawing. If the drawing is scaled, the Space After setting remains the same.

To reference the SpAfter cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Para.SpAfter[*i*]

where i = <1>, 2, 3...

To reference the SpAfter cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionParagraph**Row index: **visRowParagraph** + *i*

where i = 0, 1, 2...

Cell index: visSpaceAfter

SpBefore cell (Paragraph section)

Rows

Determines the amount of space inserted before each paragraph in the shape's text block, in addition to any space from the SpLine cell if it is the first paragraph in a text block, the TopMargin cell.

Remarks

This value is independent of the scale of the drawing. If the drawing is scaled, the Space Before setting remains the same.

To reference the SpBefore cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Para.SpBefore[*i*]

where i = <1>, 2, 3...

To reference the SpBefore cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionParagraph**Row index: **visRowParagraph** + *i*

where i = 0, 1, 2...

Cell index: visSpaceBefore

SpLine cell (Paragraph section)

See also Rows

Determines the distance between one line of text and the next, expressed as a percentage, where 100% is the height of a text line.

Value	Description
>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)

Remarks

To reference the SpLine cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Para. SpLine [i] where i = <1>, 2, 3...

To reference the SpLine cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionParagraph visRowParagraph** + i where i = 0, 1, 2...

Cell index: visSpaceLine

Strikethru cell (Character section)

See also Rows

Determines whether the text is formatted as strikethrough.

Value	Description
TRUE	Text is formatted as strikethrough.
FALSE	Text is not formatted as strikethrough.

Remarks

You can also set the value of this cell by using the **Text** dialog box (click **Text** on the **Format** menu).

To reference the Strikethru cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Strikethru[i] where i = <1>, 2, 3...

To reference the Strikethru cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2...

Cell index: visCharacterStrikethru

Style cell (Character section)

Rows

Shows the character formatting applied to a range of text in the shape's text block.

Style	Value	Automation constant
Bold	&H1	visBold
Italic	&H2	visItalic
Underline	&H4	visUnderLine
Small caps	&H8	visSmallCaps

Remarks

The Style cell contains formatting information applied to a sub-range of a shape's text if the Characters section contains multiple rows. Otherwise, it contains formatting information for all of the shape's text.

The value represents a binary number in which each bit indicates a character style. 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. You can test for a particular format using Boolean BIT* functions. See your programming documentation for details about these functions.

To reference the Style cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.Style[i]

where i = <1>, 2, 3...

To reference the Style cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2...

Cell index: visCharacterStyle

Example

Suppose the Color cell in the first row of a shape's Character section is set to this formula:

= IF(BITAND(Char.Style,1)=1,4,3)

Then if the first character of the shape's text is bold, the text covered by the first Character properties row will be blue (4); otherwise it will be green (3). This example assumes default colors are in effect.

The following is an example of setting the Style cell in a program. The first statement references the Style cell by name, and the second statement references the Style cell by index. Both statements apply italics to the text covered by the second row of a shape's Character section.

shpObj.Cells("Char.Style[2]").ResultIU = visItalic shpObj.CellsSRC(visSectionCharacter, visRowCharacter visCharacterStyle).ResultIU = visItalic

SubAddress cell (Hyperlinks section)

Specifies a location within the target document to link to.

Remarks

For example, if the Address cell is "Drawing1.vsd", the SubAddress 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.microsoft.com/visio/", the value of this cell can be a named anchor within the document, such as "solutions".

You can also set the value of this cell using the **Hyperlinks** dialog box (click **Hyperlinks** on the **Insert** menu).

To reference the SubAddress cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Hyperlink.*Name*.SubAddress

where Hyperlink. *Name* is the row name

To reference the SubAddress cell by index from a program, use the **CellsSRC**

property with the following arguments:

Section index: visSectionHyperlink Row index: visRowHyperlink + i

where i = 0, 1, 2...

Cell index: visHLinkSubAddress

TextBkgnd cell (Text Block Format section)

Rows

Determines the text background color for a shape.

Remarks

The TextBkgnd cell can have any value from 0 through 24, or 255. The values 0 and 255 (**visTxtBlklOpaque**) both indicate a transparent text background. To select a color value, add 1 to the value displayed in the **Color Palette** dialog box (click **Color Palette** on the **Tools** menu).

To enter a custom color, use the RGB or HSL function plus one—for example, RGB(255,127,255)+1. The value of a custom color is its RGB color, and RGB(r, g, b)+1, rather than a number, will be shown in the ShapeSheet window. When used in numeric operations, custom colors have values of 25 and above.

You can set the transparency of the text background color in the TextBkgndTrans cell.

To reference the TextBkgnd cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TextBkgnd

To reference the TextBkgnd cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowText**

Cell index: visTxtBlkBkgnd

TextBkgndTrans cell (Text Block Format section)

Rows

Determines the transparency level for the background color of the shape's text block.

Value	Description
0 – 100	Represents the percentage of transparency. The default is 0% (completely opaque).

Remarks

Values are rounded to the nearest half percent. A value of 100% is completely transparent. Although a shape with a completely transparent text background appears the same as a shape with no text background on the drawing page, it will interact with other objects on the page in the same ways as if its transparency is 0%.

You can also set this value using the slider control in the **Text Format** dialog box (click **Text** on the **Format** menu, then select the **Text Block** tab).

To reference the TextBkgndTrans cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TextBkgndTrans

To reference the TextBkgndTrans cell by index from a program, use the

CellsSRC property with the following arguments:

Section index: visSectionObject

Row index: **visRowText**

Cell index: visTxtBlkBkgndTrans

TextDirection cell (Text Block Format section)

See also Rows

Determines the direction of the characters in a text block.

Value	Direction	Automation constant
0	Horizontal	visTxtBlkLeftToRight
1	Vertical	visTxtBlkTopToBottom

Remarks

In Microsoft Visio version 5.0 Japanese products, the value of this cell was stored in the VerticalText cell in the Miscellaneous section.

To reference the TextDirection cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TextDirection

To reference the TextDirection cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowText**

Cell index: visTxtBlkDirection

TheData cell (Events section)

See also Rows

Reserved for future use.

TheText cell (Events section)

Rows

An event cell that is evaluated when a shape's text or text composition changes.

Remarks

Event cells are evaluated only when the event occurs, not upon formula entry. You can use the TheText cell to trigger recalculations, for example, to recalculate the text width and height with the TEXTWIDTH() and TEXTHEIGHT() functions.

To reference the TheText cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TheText

To reference the TheText cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowEvent**

Cell index: visEvtCellTheText

Tip cell (Controls section)

See also Rows

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

Remarks

To reference the Tip cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Controls.Prompt[i]

where i = <1>, 2, 3...

To reference the Tip cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionControls**

Row index: $\mathbf{visRowControl} + i$

where i = 0, 1, 2...

Cell index: visCtlTip

TopMargin cell (Text Block Format section)

See also Rows

Determines the distance between the top border of the text block and the first line of text it contains. The default is 4.0000 point. This value is independent of the scale of the drawing. If the drawing is scaled, the top margin remains the same.

Remarks

To reference the TopMargin cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TopMargin

To reference the TopMargin cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowText

Cell index: visTxtBlkTopMargin

Transparency cell (Character section)

Rows

Determines the transparency level for a range of a shape's text color.

Value	Description
0 – 100	Represents the percentage of transparency. The default is 0% (completely opaque).

Remarks

Values are rounded to the nearest half percent. A value of 100% is completely transparent. Although a shape with completely transparent text appears the same as a shape with no text on the drawing page, it will interact with other objects on the page in the same ways as if its transparency is 0%.

You can also set this value using the slider control in the **Text Format** dialog box (click **Text** on the **Format** menu, then select the **Font** tab).

The Transparency cell contains formatting information applied to a sub-range of a shape's text if the Characters section contains multiple rows. Otherwise, it contains formatting information for all of the shape's text.

To reference the Transparency cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Char.ColorTrans[i] where i = <1>, 2, 3...

To reference the Transparency cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionCharacter**Row index: **visRowCharacter** + *i*

where i = 0, 1, 2...

Cell index: visCharacterColorTrans

Transparency cell (Image Properties section)

Determines the transparency level for a layer color.

Value	Description
0 – 100	Represents the percentage of transparency. The default is 0% (completely opaque).

Remarks

Values are rounded to the nearest half percent. A value of 100% is completely transparent. Although a layer with completely transparent color appears the same as a layer with no color on the drawing page, it will interact with other objects on the page in the same ways as if its transparency is 0%. You can also set this value using the slider control in the **Layer Properties** dialog box (click **Layer properties** on the **View** menu).

To reference the Transparency cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Transparency

To reference the Transparency cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionObject**Row index: **visRowImage**

Cell index: visImageTransparency

Transparency cell (Layers section)

Rows

Determines the transparency level for a layer color.

Value	Description
0 – 100	Represents the percentage of transparency. The default is 0% (completely opaque).

Remarks

Values are rounded to the nearest half percent. A value of 100% is completely transparent. Although a layer with completely transparent color appears the same as a layer with no color on the drawing page, it will interact with other objects on the page in the same ways as if its transparency is 0%. You can also set this value using the slider control in the **Layer Properties** dialog box (click **Layer properties** on the **View** menu).

To reference the Transparency cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Layers.ColorTrans[i] where i = <1>, 2, 3...

To reference the Transparency cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionLayer**Row index: **visRowLayer** + *i*

where i = 0, 1, 2...

Cell index: visLayerColorTrans

TxtAngle cell (Text Transform section)

See also Rows

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

Remarks

To reference the TxtAngle cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TxtAngle

To reference the TxtAngle cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowTextXForm
Cell index: visXFormAngle

TxtHeight cell (Text Transform section)

See also Rows

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

= Height * 1

This formula evaluates to the height of the shape.

Remarks

To reference the TxtHeight cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TxtHeight

To reference the TxtHeight cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowTextXForm
Cell index: visXFormHeight

TxtLocPinX cell (Text Transform section)

See also Rows

Determines 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

This formula evaluates to the horizontal center of the text block.

Remarks

To reference the TxtLocPinX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TxtLocPinX

To reference the TxtLocPinX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowTextXForm
Cell index: visXFormLocPinX

TxtLocPinY cell (Text Transform section)

See also Rows

Determines 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

This formula evaluates to the vertical center of the text block.

Remarks

To reference the TxtLocPinY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TxtLocPinY

To reference the TxtLocPinY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowTextXForm
Cell index: visXFormLocPinY

TxtPinX cell (Text Transform section)

See also Rows

Determines 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

This formula evaluates to the horizontal center of the shape.

Remarks

To reference the TxtPinX cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TxtPinX

To reference the TxtPinX cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowTextXForm
Cell index: visXFormPinX

TxtPinY cell (Text Transform section)

See also Rows

Determines 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

This formula evaluates to the vertical center of the shape.

Remarks

To reference the TxtPinY cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TxtPinY

To reference the TxtPinY cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowTextXForm
Cell index: visXFormPinY

TxtWidth cell (Text Transform section)

See also Rows

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

= Width * 1

This formula evaluates to the width of the shape.

Remarks

To reference the TxtWidth cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: TxtWidth

To reference the TxtWidth cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowTextXForm
Cell index: visXFormWidth

Type cell (Custom Properties section)

Specifies a data type for the custom property value.

Value	Description	Automation constant
0	String. This is the default.	visPropTypeString
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.	visPropTypeListFix
2	Number. Includes date, time, duration, and currency values as well as scalars, dimensions, and angles. Specify a format picture in the Format cell.	visPropTypeNumber
3	Boolean. Displays FALSE and TRUE as items users can select from a drop-down list box in the Custom Properties dialog box.	visPropTypeBool
4	Variable list. Displays the list items in a drop-down combo box in the Custom	visPropTypeListVar

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.

Date or time value. Displays days, **visPropTypeDate** months, and years, or seconds, minutes, and hours, or a combined date and time value. Specify a format picture in the

Format cell.

Duration value. Displays elapsed time. **visPropTypeDuration**Specify a format picture in the Format cell.

Currency value. Uses the system's current **visPropTypeCurrency**Regional Settings. Specify a format
picture in the Format cell.

Remarks

To reference the Type cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Prop.*Name*.Type

where Prop. Name is the row name

To reference the Type cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionProp**Row index: **visRowProp** + *i*

where i = 0, 1, 2...

Cell index: visCustPropsType

Type cell (Text Fields section)

Rows

Specifies a data type for the text field value.

Value	Description
0	String.
2	Number. Includes date, time, duration, and currency values as well as scalars, dimensions, and angles. Specify a format picture 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.

Remarks

You can also set the value of this cell using the **Field** dialog box (with a shape selected, click **Field** on the **Insert** menu).

To reference the Type cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Fields.Type[i] where i = <1>, 2, 3...

To reference the Type cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionTextField

Row index: visRowField + i

where i = 0, 1, 2...

Cell index: visFieldType

Type / C cell (Connection Points section)

See also

Determines the connection point type.

Value	Type	Automation constant
0	Inward	visCnnctTypeInward
1	Outward	visCnnctTypeOutward
2	Inward & Outward	visCnnctTypeInwardOutward

Remarks

You can also set the connection point type by choosing the **Connector** tool, selecting a shape, and then right-clicking a connection point. To do this, you need to run in developer mode (on the **Tools** menu, click **Options**, click the **Advanced** tab, and then select the **Run in developer mode** check box).

To reference the Type / C cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Connections. Type[i]

where
$$i = <1>, 2, 3...$$

To reference the Type / C cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionConnectionPts**Row index: **visRowConnectionPts** + *i*

where i = 0, 1, 2...

Cell index: **visCnnctType** (non-extended rows)

visCnnctC (extended rows)

For information about non-extended and extended rows, see Connections.Row row.

UICategory cell (Text Fields section)

See also Rows

Determines the category of an inserted field in versions of Microsoft Visio earlier than Visio 2000.

Remarks

This cell does not appear in the ShapeSheet window. Use this cell if you need to deal with backward capability issues such as saving a Visio version 2000 drawing in Visio version 5.0 file format.

To reference the UICategory cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Fields.UICat[*i*]

where i = <1>, 2, 3...

To reference the UICategory cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionTextField

Row index: visRowField + i

where i = 0, 1, 2...

Cell index: visFieldUICategory

UICode cell (Text Fields section)

See also Rows

Determines the code of an inserted field in versions of Microsoft Visio earlier than Visio 2000.

Remarks

This cell does not appear in the ShapeSheet window. Use this cell if you need to deal with backward capability issues, such as saving a Visio version 2000 drawing in Visio version 5.0 file format.

To reference the UICode cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Fields.UICod[i]

where i = <1>, 2, 3...

To reference the UICode cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionTextField

Row index: $\mathbf{visRowField} + i$

where i = 0, 1, 2...

Cell index: visFieldUICode

UIFormat cell (Text Fields section)

See also Rows

Determines the format of an inserted field in versions of Microsoft Visio earlier than Visio 2000.

Remarks

This cell does not appear in the ShapeSheet window. Use this cell if you need to deal with backward capability issues, such as saving a Visio version 2000 drawing in Visio version 5.0 file format.

To reference the UIFormat cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Fields.UIFmt[*i*]

where i = <1>, 2, 3...

To reference the UIFormat cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionTextField

Row index: visRowField + i

where i = 0, 1, 2...

Cell index: visFieldUIFormat

UpdateAlignBox cell (Miscellaneous section)

See also Rows

Recalculates the selection rectangle whenever a control handle is moved.

Remarks

To reference the UpdateAlignBox cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: UpdateAlignBox

To reference the UpdateAlignBox cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowMisc

Cell index: visUpdateAlignBox

Value cell (Custom Properties section)

See also

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

Remarks

Formulas entered in this cell are overridden by values entered in the **Custom Properties** dialog box. This is true even if you use the GUARD function to protect the formula.

To reference the Value cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Prop.Name.Value

where Prop. Name is the row name

To reference the Value cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionProp**Row index: **visRowProp** + *i*

where i = 0, 1, 2...

Cell index: visCustPropsValue

Value cell (Text Fields section)

Rows

Contains the function for a field.

Remarks

You can set the value of this cell using the **Field** dialog box (click **Field** on the **Insert** menu). To see a list of functions, see the Functions list in this Developer Reference.

Formulas entered in this cell are overridden by values entered in the **Custom Properties** dialog box. This is true even if you use the GUARD function to protect the formula.

To reference the Value cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Prop.*Name*.Value

where Prop. Name is the row name

To reference the Value cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionProp**Row index: **visRowProp** + *i*

where i = 0, 1, 2...

Cell index: visCustPropsValue

Value cell (User-Defined Cells section)

See also

Specifies a value for the corresponding user-defined cell.

Remarks

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

To reference the Value cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: User.Name.Value

where User. *Name* is the row name

To reference the Value cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionUser**Row index: **visRowUser** + *i*

where i = 0, 1, 2...

Cell index: visUserValue

VerticalAlign cell (Text Block Format section)

See also Rows

Determines the vertical alignment of text within the text block.

Value	Description	Automation constant
0	Top	visVertTop
1	Middle	visVertMiddle
2	Bottom	visVertBottom

Remarks

To reference the VerticalAlign cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: VerticalAlign

To reference the VerticalAlign cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: **visRowText**

Cell index: visTxtBlkVerticaAlign

Visible cell (Layers section)

See also Rows

Specifies whether shapes belonging to the layer are visible on the drawing page.

Value	Description
TRUE	Shapes are visible.
FALSE	Shapes are hidden.

Remarks

This cell corresponds to the **Visible** option in the **Layer Properties** dialog box (click **Layer Properties** on the **View** menu).

To reference the Visible cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Layers.Visible[i] where i = <1>, 2, 3...

To reference the Visible cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionLayer**Row index: **visRowLayer** + *i*

where i = 0, 1, 2...

Cell index: visLayerVisible

WalkPreference cell (Glue Info section)

Rows

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. By default, both endpoints of the 1-D shape move to horizontal connection points.

Value	Description	Automation constant
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-	visWalkPrefBegNS
2	to-side connections). 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).	

Remarks

This cell has no effect on dynamic connectors. A dynamic connector's behavior is determined by its routing style. See the RouteStyle cell for the default routing style for dynamic connectors on a page, or the ShapeRouteStyle for the routing style of a particular dynamic connector.

To reference the WalkPreference cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: WalkPreference

To reference the WalkPreference cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject

Row index: visRowMisc
Cell index: visWalkPref

Width cell (Shape Transform section)

See also Rows

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

=
$$SQRT((EndX - BeginX) \land 2 + (EndY - BeginY) \land 2)$$

Remarks

To reference the Width cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Width

To reference the Width cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowXFormOut
Cell index: visXFormWidth

X cell (Connection Points section)

See also

Represents the *x*-coordinate for a connection point in local coordinates.

Remarks

To reference the X cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Connections.Xi

where i = <1>, 2, 3...

To reference the X cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionConnectionPts**Row index: **visRowConnectionPts** + *i*

where i = 0, 1, 2...

Cell index: visX

X cell (Controls section)

See also Rows

Represents the *x*-coordinate that indicates the location of a shape's control handle in local coordinates.

Remarks

To reference the X cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Controls.X*i*

where i = <1>, 2, 3...

To reference the X cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionControls**Row index: **visRowControls** + *i*

where i = 0, 1, 2...

Cell index: visCtlX

X cell (Geometry section)

See also

Represents an *x*-coordinate on a shape in local coordinates. This table describes the X cell based on the row in which it's located.

Row	Description
<u>MoveTo</u>	If the MoveTo row is the first row in the section, the X cell represents the <i>x</i> -coordinate of the first vertex of a path. If the MoveTo row appears between two rows, the X cell represents the <i>x</i> -coordinate of the first vertex after the break in the path.
<u>LineTo</u>	The <i>x</i> -coordinate of the ending vertex of a straight line segment.
<u>ArcTo</u>	The x -coordinate of the ending vertex of an arc.
<u>EllipticalArcTo</u>	The <i>x</i> -coordinate of the ending vertex of an elliptical arc.
<u>PolylineTo</u>	The <i>x</i> -coordinate of the ending vertex of a polyline.
NURBSTo	The <i>x</i> -coordinate of the last control point of a nonuniform rational B-spline (NURBS).
<u>SplineStart</u>	The x -coordinate of a spline's second control point.
<u>SplineKnot</u>	The <i>x</i> -coordinate of a control point.

InfiniteLine An *x*-coordinate of a point on the infinite line. Ellipse The *x*-coordinate of the center of the ellipse.

Remarks

To reference the X cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometryi.X*j*

where *i* and j = <1>, 2, 3...

Geometryi.X1 (InfiniteLine and Ellipse rows)

where i = <1>, 2, 3...

To reference the X cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: visRowVertex + j

where j = 0, 1, 2...

visRowVertex (InfiniteLine and Ellipse rows)

Cell index: visX (MoveTo, LineTo, ArcTo, EllipticalArcTo,

NURBSTo, Polyline, SplineStart, and SplineKnot

rows)

visInfiniteLineX1 (InfiniteLine row)
visEllipseCenterX (Ellipse row)

X Behavior cell (Controls section)

See also Rows

Controls the type of behavior the *x*-coordinate of the control handle will exhibit after the handle is moved.

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

left edge, handle is not visible.
hidden

8 Offset from Same as 3, but the control visCtlOffsetMidHidden center, hidden handle is not visible.

9 Offset from Same as 4, but the control visCtlOffsetMaxHidden right edge, handle is not visible.
hidden

Remarks

To reference the X Behavior cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Controls.XCon[*i*]

where i = <1>, 2, 3...

To reference the X Behavior cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionControls**Row index: **visRowControls** + *i*

where i = 0, 1, 2...

Cell index: visCtlXCon

X Dynamics cell (Controls section)

See also Rows

Represents the *x*-coordinate for a control handle's anchor point in local coordinates.

Remarks

The anchor point is used for rubber-banding during dynamics.

To reference the X Dynamics cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Controls.Xdyn[i]

where i = <1>, 2, 3...

To reference the X Dynamics cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionControls**Row index: **visRowControls** + *i*

where i = 0, 1, 2...

Cell index: visCtlXDyn

XGridDensity cell (Ruler & Grid section)

See also Rows

Specifies the type of horizontal grid to use.

Value	Description	Automation constant
0	Fixed	visGridFixed
2	Coarse	visGridCoarse
4	Normal (default)	visGridNormal
8	Fine	visGridFine

Remarks

This cell corresponds to the horizontal **Grid spacing** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the XGridDensity cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: XGridDensity

To reference the XGridDensity cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visXGridDensity

XGridOrigin cell (Ruler & Grid section)

See also Rows

Specifies the horizontal coordinate of the grid origin.

Remarks

This cell corresponds to the horizontal **Grid origin** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the XGridOrigin cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: XGridOrigin

To reference the XGridOrigin cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visXGridOrigin

XGridSpacing cell (Ruler & Grid section)

Rows

Specifies the distance between horizontal lines in a fixed grid (XGridDensity = 0).

Remarks

This cell corresponds to the horizontal **Minimum spacing** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the XGridSpacing cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: XGridSpacing

To reference the XGridSpacing cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visXGridSpacing

XRulerDensity cell (Ruler & Grid section)

See also Rows

Specifies the horizontal subdivisions on the ruler for the page.

Value	Description	Automation constant
0	Fixed	visRulerFixed
8 (&H8)	Coarse	visRulerCoarse
16 (&H10)	Normal (Default)	visRulerNormal
32 (&H20)	Fine	visRulerFine

Remarks

This cell corresponds to the horizontal **Subdivisions** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the XRulerDensity cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: XRulerDensity

To reference the XRulerDensity cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visXRulerDensity

XRulerOrigin cell (Ruler & Grid section)

See also Rows

Specifies the zero point on the *x*-axis ruler for the page.

Remarks

This cell corresponds to the horizontal **Ruler zero** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the XRulerOrigin cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: XRulerOrigin

To reference the XRulerOrigin cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visXRulerOrigin

Y cell (Connection Points section)

See also

Represents the *y*-coordinate for a connection point in local coordinates.

Remarks

To reference the Y cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Connections. *Yi*

where i = <1>, 2, 3...

To reference the Y cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionConnectionPts**Row index: **visRowConnectionPts** + *i*

where i = 0, 1, 2...

Cell index: visY

Y cell (Controls section)

See also Rows

Represents the *y*-coordinate that indicates the location of a shape's control handle in local coordinates.

Remarks

To reference the Y cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Controls.Yi

where i = <1>, 2, 3...

To reference the Y cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionControls**Row index: **visRowControls** + *i*

where i = 0, 1, 2...

Cell index: visCtlY

Y cell (Geometry section)

See also

Represents a *y*-coordinate on a shape in local coordinates. This table describes the Y cell based on the row in which it's located.

Row	Description
<u>MoveTo</u>	If the MoveTo row is the first row in the section, the Y cell represents the <i>y</i> -coordinate of the first vertex of a path. If the MoveTo row appears between two rows, the Y cell represents the y-coordinate of the first vertex after the break in the path.
<u>LineTo</u>	The <i>y</i> -coordinate of the ending vertex of a straight line segment.
<u>ArcTo</u>	The <i>y</i> -coordinate of the ending vertex of an arc.
<u>EllipticalArcTo</u>	The <i>y</i> -coordinate of the ending vertex of an elliptical arc.
<u>PolylineTo</u>	The <i>y</i> -coordinate of the ending vertex of a polyline.
<u>NURBSTo</u>	The <i>y</i> -coordinate of the last control point of a nonuniform rational B-spline (NURBS).
<u>SplineStart</u>	The <i>y</i> -coordinate of a spline's second control point.
<u>SplineKnot</u>	The <i>y</i> -coordinate of a control point.

InfiniteLine A *y*-coordinate of a point on the infinite line. Ellipse The *y*-coordinate of the center of the ellipse.

Remarks

To reference the X cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Geometryi.Y*j*

where *i* and j = <1>, 2, 3...

Geometryi.Y1 (InfiniteLine and Ellipse rows)

where i = <1>, 2, 3...

To reference the X cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionFirstComponent** + *i*

where i = 0, 1, 2...

Row index: visRowVertex + j

where j = 0, 1, 2...

visRowVertex (InfiniteLine and Ellipse rows)

Cell index: visY (MoveTo, LineTo, ArcTo, EllipticalArcTo,

NURBSTo, Polyline, SplineStart, and SplineKnot

rows)

visInfiniteLineY1 (InfiniteLine row)
visEllipseCenterY (Ellipse row)

Y Behavior cell (Controls section)

See also Rows

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	Automation constant
0	Proportional	The control handle can be moved, and it also moves in proportion with the shape when it is stretched.	visCtlProportional
1	Proportional locked	The control handle moves in proportion with the shape, but the control handle itself cannot be moved.	visCtlLocked
2		The control handle is offset a constant distance from the bottom of the shape.	visCtlOffsetMin
3	Offset from center	The control handle is offset a constant distance from the center of the shape.	visCtlOffsetMid
4	Offset from top edge	The control handle is offset a constant distance from the top of the shape.	visCtlOffsetMax
5	Proportional, hidden	Same as 0, but the control handle is not visible.	visCtlProportionalHidden
6	Proportional locked, hidden	Same as 1, but the control handle is not visible.	visCtlLockedHiddenv

7	Same as 2, but the control handle is not visible.	visCtlOffsetMinHidden
8	Same as 3, but the control handle is not visible.	visCtlOffsetMidHidden
9	Same as 4, but the control handle is not visible.	visCtlOffsetMaxHidden

Remarks

To reference the Y Behavior cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Controls.Ycon[*i*]

where i = <1>, 2, 3...

To reference the Y Behavior cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionControls**Row index: **visRowControls** + *i*

where i = 0, 1, 2...

Cell index: visCtlYCon

Y Dynamics cell (Controls section)

See also Rows

Represents the *y*-coordinate for a control handle's anchor point in local coordinates. The anchor point is used for rubber-banding during dynamics.

Remarks

To reference the Y Dynamics cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: Controls.YDyn[i]

where i = <1>, 2, 3...

To reference the X Dynamics cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: **visSectionControls**Row index: **visRowControls** + *i*

where i = 0, 1, 2...

Cell index: visCtlYDyn

YGridDensity cell (Ruler & Grid section)

See also Rows

Specifies the type of vertical grid to use.

Value	Description	Automation constant
0	Fixed	visGridFixed
2	Coarse	visGridCoarse
4	Normal (default)	visGridNormal
8	Fine	visGridFine

Remarks

This cell corresponds to the vertical **Grid spacing** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the YGridDensity cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: YGridDensity

To reference the YGridDensity cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visYGridDensity

YGridOrigin cell (Ruler & Grid section)

See also Rows

Specifies the vertical origin of the grid.

Remarks

This cell corresponds to the vertical **Grid origin** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the YGridOrigin cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: YGridOrigin

To reference the YGridOrigin cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visYGridOrigin

YGridSpacing cell (Ruler & Grid section)

Rows

Specifies the distance between vertical lines in a fixed grid (YGridDensity = 0).

Remarks

Corresponds to the vertical **Minimum spacing** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the YGridSpacing cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: YGridSpacing

To reference the YGridSpacing cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visYGridSpacing

YRulerDensity cell (Ruler & Grid section)

See also Rows

Specifies the vertical subdivisions on the ruler for the page.

Value	Description	Automation constant
0	Fixed	visRulerFixed
8 (&H8)	Coarse	visRulerCoarse
16 (&H10)	Normal (Default)	visRulerNormal
32 (&H20)	Fine	visRulerFine

Remarks

This cell corresponds to the vertical **Subdivisions** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the YRulerDensity cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: YRulerDensity

To reference the YRulerDensity cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visYRulerDensity

YRulerOrigin cell (Ruler & Grid section)

See also Rows

Specifies the zero point on the *y*-axis ruler for the page.

Remarks

This cell corresponds to the vertical **Ruler zero** option in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

To reference the YRulerOrigin cell by name from another formula, or from a program using the <u>Cells</u> property, use:

Cell name: YRulerOrigin

To reference the YRulerOrigin cell by index from a program, use the <u>CellsSRC</u> property with the following arguments:

Section index: visSectionObject
Row index: visRowRulerGrid
Cell index: visYRulerOrigin

ABS function

Returns the absolute value of a number.

Syntax

ABS(number)

Example

ABS(-1.24)

Returns 1.24.

ACOS function

See also

Returns the arccosine of number, for example, the angle whose cosine is *number*.

Syntax

ACOS(number)

Remarks

The input value must be in the range $-1 \le$ number ≤ 1 , or a #NUM! error is returned. The resulting angle is in the range $0 \le$ angle \le PI radians ($0 \le$ angle ≤ 180 degrees).

Example

ACOS(0)

Equals 90 deg.

AND function

See also

Returns TRUE (1) if all of the logical expressions supplied are TRUE. If any of the logical expressions are FALSE or 0, the AND function returns FALSE (0).

Syntax

AND(logical expression1,logical expression2,...,logical expressionN)

logical A combination of constants, operators, functions, and expression references to ShapeSheet cells that results in a value. Any

expression that evaluates to a non-zero value is considered to be

TRUE.

Example

AND(Height > 1, PinX > 1)

Returns TRUE if both expressions are TRUE. Returns FALSE if either expression is FALSE.

ANG360 function

See also

Normalizes an angle's range to be $0 \le \text{result} \le 2\text{PI radians}$ ($0 \le \text{result} \le 360$ deg.).

Syntax

ANG360(angle)

Remarks

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.

Example 1

ANG360(395 deg.)

Equals 35 deg.

Example 2

ANG360(-9.8 rad.)

Equals 2.7664 rad.

Example 3

ANG360(45)

Equals 58.31 deg. (1.0177 rad.)

ANGLETOLOC function

See also

Returns a transformed angle in the destination shape's local coordinate system. It converts an angle from local coordinates in a source shape to the local coordinates in a destination shape. You can use the ANGLETOLOC function to set local angle cells in a shape in terms of an angle from another coordinate space.

Syntax

ANGLETOLOC(srcAngle, srcRef, dstRef)

srcAngle An angle in the source coordinate system.

srcRef A reference to a cell in the source object, such as a shape,

group, page, and so on.

dstRef A reference to a cell in the destination object, such as a shape,

group, page, and so on.

Remarks

This function works even when the source and destination shapes are within groups. It also adjusts for rotation and flips in the intermediate transformation.

The source and destination coordinates must be on the same page.

ANGLETOPAR function

See also

Returns a transformed angle in the destination shape's parent coordinate system. It converts an angle from local coordinates in a source shape to the parent coordinates in a destination shape.

Syntax

ANGLETOPAR(srcAngle, srcRef, dstRef)

srcAngle An angle in the source coordinate system.

srcRef A reference to a cell in the source object, such as a shape,

group, page, and so on.

dstRef A reference to a cell in the destination object, such as a shape,

group, page, and so on.

Remarks

This function works even when the source and destination shapes are within groups. It also adjusts for rotation and flips in the intermediate transformation.

The source and destination coordinates must be on the same page.

If the destination is a page, which has no parent, the result is expressed in page's local coordinates.

ASIN function

See also

Returns the arcsine of a number, for example, the angle whose sine is *number*.

Syntax

ASIN(number)

Remarks

The input value must be in the range $-1 \le number \le 1$, or a #NUM! error is returned. The resulting angle is in the range $-PI/2 \le number \le PI/2$ radians (-90 <= angle <= 90 degrees).

Example

ASIN(1)

Equals 90 deg.

ATAN function

See also

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

Syntax

ATAN(number)

Example

ATAN(1)

Equals 45 deg.

ATAN2 function

See also

Returns the angle between the vector represented by x,y and the direction of the x-axis. The result is a number in the current unit of measure for angles.

Syntax

ATAN2(y,x)

Remarks

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 Microsoft Visio, ATAN2(0,0) returns 0. To force the result of ATAN2 into a different angular measurement, use the DEG or RAD function.

The ATAN2 function is the antifunction of the TAN function. The ATAN2 function 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 "opposite side" and x is the "adjacent side," so the function could be written as ATAN2(opposite,adjacent).

Example 1

ATAN2(1.25,2.25)

Returns 29.0456 degrees.

Example 2

ATAN2(1,SQRT(3))

Returns 30 degrees.

Example 3

ATAN2(1,1)

Returns 45 degrees.

BITAND function

See also

Returns a 16-bit 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.

Syntax

BITAND(binarynumber1,binarynumber2)

Example

BITAND(12,6)

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

BITNOT function

See also

Returns a 16-bit binary number in which each bit is set to 1 only if the corresponding bit in *binary number* is 0. Otherwise, the bit is set to 0.

Syntax

BITNOT(binary number)

Example

BITNOT(6)

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

BITOR function

See also

Returns a 16-bit binary number in which each bit is set to 1 if the corresponding bit in either *binary number1* or *binary number2* is 1. The bit is set to 0 only if the corresponding bit is 0 in both *binary number1* and *binary number2*.

Syntax

BITOR(binary number1,binary number2)

Example

BITOR(12,6)

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

BITXOR function

See also

Returns a 16-bit binary number in which each bit is set to 1 if the corresponding bit in either but not both *binary number1* and *binary number2* is 1. Otherwise, the bit is set to 0.

Syntax

BITXOR(binary number1,binary number2)

Example

BITXOR(12,6)

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

BKGPAGENAME function

See also

Returns a background page name as a string.

Syntax

BKGPAGENAME (langID_opt)

langID_opt

Optional argument. Use to specify a language for the string the function returns. Use 0 (default value) to specify the local language. Use 750 to specify universal language.

Remarks

If the page for which you are using the function doesn't have a background page, the string "<no background>" is returned.

If you pass an illegal language code, the local language is used.

BLUE function

See also

Returns the blue component of a color. The return value is an integer in the range of 0 to 255, inclusive. The input is an index of a color in the document's color table, an expression that resolves to a custom color (like RGB or HSL), or a reference to a cell that contains a color index or color result. The function returns 0 for invalid input.

Syntax

BLUE(expression)

Example 1

BLUE(Sheet.4!FillForegnd)

Returns the blue component of Sheet.4's fill foreground color.

Example 2

BLUE(13)

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

Example 3

BLUE(RGB(10, 20, 30))

Returns 30.

CALLTHIS function

Calls a procedure in a Microsoft Visual Basic for Applications (VBA) project.

Syntax

CALLTHIS("procedure",["project"],[arg1,arg2,...])

procedure The name of the procedure to call; required string.

project Optional string. The project that contains the procedure.

An argument of any type that can include number, string, date,

and currency values; passed as parameters to the procedure.

Remarks

In the VBA project, *procedure* is defined as follows:

procedure(shpObj As Visio.shape [arg1 As type, arg2 As t

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 this procedure and pass it a reference to my shape."

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.

Numbers in *arg1*, *arg2*... 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.

The second comma in the CALLTHIS function is optional. It corresponds to the number of additional parameters added to your procedure. If you do not use any additional parameters, except shpObj as Visio.Shape, do not add the second comma—use CALLTHIS("",). If you add two additional parameters, for example, use CALLTHIS("",,).

The CALLTHIS function 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.

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

Note VBA code invoked as a result of the Visio instance evaluating a CALLTHIS function in a formula should not close the document containing the cell using the function because an application error results and Visio terminates.

If you need to close the document containing the cell that uses the CALLTHIS function, use one of the following techniques:

Close the document from non-VBA code.

Close the document from a project other than the one that is closing.

Post window messages to close windows into the document rather than closing the document.

Example 1

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

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

Example 2

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

Calls the procedure named q located in a module and passes the cell's height in centimeters and Sheet.2's width in internal units.

Example 3

Use the following procedure in the **ThisDocument** class module.

Sub A(shpObj As Visio.Shape)

MsgBox "Click OK."

End Sub

Sub B(shpObj As Visio.Shape, strA As String)

MsgBox strA

End Sub

Sub C(shpObj As Visio.Shape, strA As String, strB As Str MsgBox strA & strB

End Sub

Use any of the following syntax in a shape's EventDblClick cell with the procedures above.

CALLTHIS("ThisDocument.A",)

CALLTHIS("ThisDocument.B",,"Click")

CALLTHIS("ThisDocument.C",,"Click", "OK.")

CATEGORY function

See also

Returns the text from the **Category** field of a document's properties (click **Properties** on the **File** menu).

Syntax

CATEGORY ()

CEILING function

See also

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

Syntax

CEILING(number, multiple)

Remarks

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.

Example 1

CEILING(3.7)

Equals 4.

Example 2

CEILING(-3.7)

Equals -4.

Example 3

CEILING(3.7, 0.25)

Equals 3.75.

CHAR function

See also

Returns the ANSI character for a number.

Syntax

CHAR(number)

Remarks

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.

COMPANY function

See also

Returns the text from the **Company** field of a document's properties (click **Properties** on the **File** menu).

Syntax

COMPANY ()

COS function

Returns the cosine of an angle.

Syntax

COS(angle)

Remarks

Uses radians unless a unit of measure is specified with angle.

Example

COS(45 degrees)

Returns 0.7071.

COSH function

Returns the hyperbolic cosine of an angle.

Syntax

COSH(angle)

Remarks

Uses radians unless a unit of measure is specified with angle.

Example

COSH(45 degrees)

Returns 1.3246.

CREATOR function

See also

Returns the text from the **Author** field of a document's properties (click **Properties** on the **File** menu).

Syntax

CREATOR ()

CY function

Returns a currency value.

Syntax

CY(value, cyID)

value Optional. A number or a string that includes currency-specific

formatting. If not specified, the currency value is formatted according to the currency style in the system's Regional

Settings.

cyID Optional. A numeric currency ID or a three-character quoted

string for the ISO 4217 abbreviation.

Remarks

To specify a different currency, you must include a valid *cyID*. For a list, see <u>About currency constants</u>.

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 *cyID* returns an error.

Example 1

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

CY(999998.993)

Returns \$999,998.99

Example 2

CY(12345678.993, "USD")

Returns \$12,345,678.99

Example 3

CY(12345678.993, "DEM")

Returns 12,345,678.99 DEM

Example 4

CY(12345678.7832, "XXX")

Returns 12,345,678.78

DATA1 function

See also

Returns the text from the **Data 1** field in the **Special** dialog box (click **Special** on the **Format** menu) for a shape as a string.

Syntax

DATA1 ()

Remarks

If the field is empty, an empty string is returned.

DATA2 function

See also

Returns the text from the **Data 2** field (click **Special** on the **Format** menu) for a shape as a string.

Syntax

DATA2 ()

Remarks

If the field is empty, an empty string is returned.

DATA3 function

See also

Returns the text from the **Data 3** field (click **Special** on the **Format** menu) for a shape as a string.

Syntax

DATA3 ()

Remarks

If the field is empty, an empty string is returned.

DATE function

Returns the date represented by *year*, *month*, and *day* formatted according to the short date style in the system's Regional Settings. The values for *year*, *month*, and *day* reflect the Gregorian calendar.

Syntax

DATE(year,month,day)

Example 1

DATE(1999,6,7)

Returns the value representing 6/7/1999.

Example 2

DATE(1999,6,7) + 4 ed.

Returns the value representing 6/11/1999.

Example 3

FORMAT(DATE(1999,10,14),"C")

Returns the value representing Tuesday, October 14, 1999.

DATETIME function

Returns the date and time value represented by *datetime* or *expression*.

Syntax

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

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. The locale identifier to be used in

evaluating a nonlocal datetime. The locale identifier is a

number described in the system header files.

Remarks

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 in the system's current Regional Settings.

The DATETIME function also accepts a single number value for *expression* where the integer portion of the result represents the number of days since December 30, 1899, and the decimal portion represents the fraction of a day since midnight.

Example 1

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

Returns the value representing 6/4/1997.

Example 2

FORMAT(DATETIME("5/20/1997 14:30:45"),"C")

Returns the string "Tuesday, May 20, 1997 2:30:45 PM."

Example 3

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

Returns the value representing 7/19/2001 1:30 PM (assuming the current year is 2001).

Example 4

DATETIME(35580.6337)

Returns the value representing 5/30/1997 3:12:32 PM.

DATEVALUE function

Returns the date value represented by *datetime* or *expression*.

Syntax

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

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. Specifies the locale identifier to be used in

evaluating a nonlocal datetime. The locale identifier is a

number described in the system header files.

Remarks

Any time component in *datetime* or *expression* is discarded.

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.

The DATEVALUE function also accepts a single number value for *expression* where the integer portion of the result represents the days since December 30, 1899.

Example 1

DATEVALUE(NOW())+5 ed.

Returns the date five days from now.

Example 2

DATEVALUE("7/19/1995 12:30")

Returns the date.

Example 3

DATEVALUE("May 33, 1997")

Returns a #VALUE! error.

Example 4

DATEVALUE(35580.6337)

Returns 5/30/1997.

DAY function

Returns an integer, 1 to 31, representing the day in *datetime* or *expression*. The DAY function uses the Gregorian calendar.

Syntax

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

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. Specifies the locale identifier to be used in

evaluating a nonlocal datetime. The locale identifier is a

number described in the system header files.

Remarks

Any time component in *datetime* or *expression* is discarded.

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

The DAY function also accepts a single number value for *expression* where the integer portion of the result represents the number of days since December 30, 1899.

Example 1

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

Returns 30.

Example 2

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

Returns 6.

Example 3

DAY(35580.6337)

Returns 30.

DAYOFYEAR function

Returns an integer, 1 to 366, representing the sequential day of the year in *datetime* or *expression*. The DAYOFYEAR function uses the Gregorian calendar.

Syntax

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

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. Specifies the locale identifier to be used in

evaluating a nonlocal datetime. The locale identifier is a

number described in the system header files.

Remarks

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, the function returns an error.

The DAYOFYEAR function also accepts a single number value for *expression* where the integer portion of the result represents the number of days since December 30, 1899.

Example 1

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

Returns 150.

Example 2

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

Returns 157.

Example 3

DAYOFYEAR(35580.6337)

Returns 150.

DECIMALSEP function

See also

Returns the decimal separator string for the current user locale.

Syntax

DECIMALSEP()

Example

SETF(GETREF(user.size), user.wholePart & DecimalSep() & user.fracPart)

DEFAULTEVENT function

See also

Performs the default event associated with the object.

Syntax

DEFAULTEVENT()

Remarks

Object	Default event
Shape	Edit text.
Group	Do nothing.
OLE	Do primary verb.
Other	Do nothing.

In earlier versions of Microsoft Visio, this function appears as _DEFAULTEVENT and the default event for a group is to open a group window.

DEG function

See also

Converts the value of an angle from radians to degrees.

Syntax

DEG(angle)

Example

DEG(PI()/4)

Returns 45.

DEPENDSON function

See also

Creates a cell reference dependency.

Syntax

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

Remarks

This function always returns FALSE. It 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's PinX or PinY cells change.

DESCRIPTION function

See also

Returns the text from the **Description** field of a document's properties (click **Properties** on the **File** menu).

Syntax

DESCRIPTION ()

DIRECTORY function

See also

Returns the full path of the directory in which a document is saved as a string, or an empty string if the document hasn't been saved.

Syntax

DIRECTORY ()

DOCCREATION function

Returns the date and time a document was created as a serial value.

Syntax

DOCCREATION ()

Remarks

Use the FORMAT function to format the value.

DOCLASTPRINT function

Returns the date and time a document was last printed as a serial value.

Syntax

DOCLASTPRINT ()

Remarks

Use the FORMAT function to format the value.

DOCLASTSAVE function

Returns the date and time a document was last saved as a serial value.

Syntax

DOCLASTSAVE ()

Remarks

Use the FORMAT function to format the value.

DOOLEVERB function

See also

Executes a verb for the OLE object.

Syntax

DOOLEVERB("verb")

Remarks

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

Example

DOOLEVERB("edit")

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

EVALTEXT function

See also

Evaluates the text in *shapename* as if it were a formula and returns the result.

Syntax

EVALTEXT(shapename!theText)

shapename!theText A cell that is triggered when the associated shape's text composition changes. Shapename! can be used to refer to the text of a shape other than the current shape.

Remarks

If there is no text, the result is zero. If the text cannot be evaluated, the function returns an error.

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.

FIELDPICTURE function

Returns a format picture string that matches the Microsoft Visio internal text field format code.

Syntax

FIELDPICTURE(code)

code

A text field format code.

Remarks

Format picture strings are used in the FORMAT function to define the expansion of values to dates, times, numbers, and unit labels.

Example

FIELDPICTURE(0)

Returns the format picture string "esc(0)", which specifies a number with one decimal place and lowercase unit description when used in the FORMAT function.

FILENAME function

See also

Returns a document's file name as a string.

Syntax

FILENAME ()

FLOOR function

See also

Rounds a number toward 0 (zero), to the next integer, or to the next instance of *multiple*.

Syntax

FLOOR(number, multiple)

Remarks

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.

Example 1

FLOOR(3.7)

Returns 3.

Example 2

FLOOR(-3.7)

Returns -3.

Example 3

FLOOR(3.7, 0.5)

Returns 3.5.

FORMAT function

Returns the result of *expression* as a string formatted according to format picture.

Syntax

FORMAT(expression, "formatpicture")

Remarks

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*.

Example 1

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

Returns "0.250 cm."

Example 2

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

Returns "0.25 CM."

Example 3

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

Returns "0.3 cm."

FORMATEX function

Returns the result of *expression* evaluated in *inputunit* as a string formatted according to *formatpicture* expressed in *outputunit*.

Syntax

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

expression A combination of constants, operators, functions, and

references to ShapeSheet cells that results in a value.

inputunit Optional. A numerical value or valid abbreviated unit of

measure (in, in., inch, etc.).

outputunit Optional. A numerical value or valid abbreviated unit of

measure (in, in., inch, etc.).

Remarks

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 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.

GETREF function

References a cell and doesn't recalculate the formula when the referenced cell changes.

Syntax

GETREF(cellname)

Example

SETF(GETREF(PinX),5.1)

Sets the formula of the PinX cell to 5.1.

GETVAL function

See also

Gets the value of a cell and doesn't recalculate the formula when the cell's value changes.

Syntax

GETVAL(cellname)

Example

GETVAL(PinX) + GETVAL(PinY) + Width

Returns the sum of the value of the PinX, PinY, and Width cells.

GOTOPAGE function

Displays the page whose name is *pagename* in the currently active window.

Syntax

GOTOPAGE("pagename")

Remarks

If a window is already displaying the page, that window becomes active. If *pagename* does not exist, the application attempts to navigate to http://pagename/. If Microsoft Visio is acting as an in-place server, the GOTOPAGE function has no effect.

You can use the HYPERLINK function to navigate to any DOS, UNC, or URL path.

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

GRAVITY function

Calculates a text block's correct angle of rotation for the indicated shape rotation to prevent the text from turning upside down.

Syntax

GRAVITY(angle,limit1,limit2)

angle The shape's angle.

limit1 Optional. Default limit is 90 degrees.*limit2* Optional. Default limit is 270 degrees.

Remarks

The GRAVITY function 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.

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.

Example 1

GRAVITY(Angle)

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

Example 2

GRAVITY(2)

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

Example 3

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

Returns 0 deg.

Example 4

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

Returns 0 deg.

GREEN function

See also

Returns the green component of a color. The return value is an integer in the range of 0 to 255, inclusive. The input is an index of a color in the document's color table, an expression that resolves to a custom color (like RGB or HSL), or a reference to a cell that contains a color index or color result. The function returns 0 for invalid input.

Syntax

GREEN(expression)

Remarks

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, it returns 0 (black).

Example 1

GREEN(Sheet.4!FillForegnd)

Returns the green component of Sheet.4's fill foreground color.

Example 2

GREEN(11)

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

Example 3

GREEN(RGB(10, 20, 30))

Returns 20.

GUARD function

Protects *expression* from deletion and change by actions performed in the drawing window, for example, moving, sizing, grouping, or ungrouping shapes.

Syntax

GUARD(expression)

Remarks

The cells most often affected by the GUARD function 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.

Example

GUARD(TEXTWIDTH(TheText) + 0.5 in)

This expression in the Width cell of a shape's Shape Transform 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 cell that gets triggered when the associated shape's text composition changes.

HELP function

See also

Opens a WinHelp or HTML Help file and displays the **Search** dialog box with the selected *keyword*. Or it displays the pop-up help topic associated with *number*, a numeric ID referenced in the MAP section of the WinHelp project file (.hpj) or the HTML Help project file (.hhp).

Syntax

HELP("filename.hlp!keyword") or HELP("filename.chm!keyword")
HELP("filename.hlp!#number") or HELP("filename.chm#number")

Remarks

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

Example 1

HELP("visio.hlp!shapesheet") or HELP("visio.chm!shapesheet")

Opens the file Visio.hlp or Visio.chm and displays the topic whose keyword is "shapesheet."

Example 2

HELP("shape.hlp!#9000") or HELP("shape.chm#9000")

Opens the file Shape.hlp or Shape.chm and displays the topic associated with the ID 9000.

HOUR function

See also

Returns an integer, 0 to 23, representing the hour of the day of *datetime* or *expression*.

Syntax

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

datetime A string commonly recognized as a date and time or a reference

to a cell containing a date and time.

expression An expression that yields a date and time.

lcid Optional. A locale identifier to be used in evaluating a nonlocal

datetime. The locale identifier is a number described in the

system header files.

Remarks

The date component in *datetime* and *expression* is discarded.

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

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

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

Example 1

HOUR("3:45 PM")

Returns 15.

Example 2

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

Returns 15.

Example 3

HOUR(0.5)

Returns 12.

Example 4

HOUR("5/30/1997")

Returns 0.

HSL function

See also

Returns a value representing an index in the document's color palette. It specifies a color by its hue, saturation, and luminosity components.

Syntax

HSL(hue,saturation,luminosity)		
hue	The color's hue, expressed as a number in the range 0 to 239, inclusive, or an expression that evaluates to such a number.	
saturation	The color's saturation, expressed as a number in the range 0 to 240, inclusive, or an expression that evaluates to such a number.	
luminosity	The color's luminosity, expressed as a number in the range 0 to 240, inclusive, or an expression that evaluates to such a number.	

Remarks

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.

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

Example 1

HSL(160,240,120)

Returns the index for the color blue.

Example 2

HSL(HUE(FillForegnd),SAT(FillForegnd),MIN(LUM(FillForegnd)+100,240))

Returns the index for a color that mirrors the fill foreground color with increased luminosity.

HUE function

See also

Returns the value of a color's hue component.

Syntax

HUE(expression)

Remarks

The return value is a number in the range 0 to 239, inclusive. The input is an index of a color in the document's color table, an expression that resolves to a custom color (like RGB or HSL), or a reference to a cell that contains a color index or color result. The function returns 0 for invalid input.

Example 1

HUE(Sheet.4!FillForegnd)

Returns the hue of Sheet.4's fill foreground color.

Example 2

HUE(7)

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

Example 3

HUE(HSL(10, 20, 30))

Returns 10.

HYPERLINK function

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

Syntax

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

address A full path or a relative path based on the **Hyperlink base** field

specified in the **Properties** dialog box (click **Properties** on the

File menu).

subaddress Optional string. Specifies a location within address to link to.

For example, if *address* is a Microsoft Visio file, *subaddress* can be a page name; if a Microsoft 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 Optional string. Passes information used in resolving the URL,

such as the coordinates of an image map.

window Optional argument. Specifies whether the hyperlink is opened

in a new window. The default value is FALSE.

frame Optional string. Specifies the name of a frame to target when

Visio is open as an Active document in an ActiveX browser, such as Microsoft Internet Explorer 3.0 or later. The default is

an empty string.

Remarks

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.

You can use the GOTOPAGE function to navigate to pages of a document.

Example 1

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

Example 2

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

Example 3

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

Example 4

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

HYPERLINKBASE function

See also

Returns the text from the **Hyperlink base** field of a document's properties (click **Properties** on the **File** menu).

Syntax

HYPERLINKBASE ()

ID function

See also

Returns a shape's internal ID as a number.

Syntax

ID ()

IF function

See also

Returns *valueiftrue* if *logicalexpression* is TRUE. Otherwise, it returns *valueiffalse*.

Syntax

IF(logicalexpression,valueiftrue,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 function

See also

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

Syntax

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

index The location that you want to find.*list* The list in which you want to search.

delimiter The string to use as a delimiter within list. A delimiter string

can be more than one character in length and include multibyte

characters. The default is a semicolon.

errorvalue Optional. A user-specified value to return if the index is out of

range. The default is an empty string.

Remarks

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, Microsoft Visio returns an empty string or the optional token provided as the *errorvalue* argument.

Example 1

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

Returns "goat".

Example 2

INDEX(54,";1;2;3;",,"ERROR")

Returns "ERROR."

INT function

See also

Rounds a number down to the next integer.

Syntax

INT(number)

Example 1

INT(7.2)

Returns 7.

Example 2

INT(-7.2)

Returns -8.

INTERSECTX function

See also

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.

Syntax

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

<i>x</i> 1	The <i>x</i> -coordinate of a point on the first line.
<i>y</i> 1	The <i>y</i> -coordinate of a point on the first line.
angle1	The value of the Angle cell for the first line.
<i>x</i> 2	The <i>x</i> -coordinate of a point on the second line.
<i>y</i> 2	The <i>y</i> -coordinate of a point on the second line.
angle2	The value of the Angle cell for the second line.

Remarks

Microsoft Visio uses this function in the PinX cell of a shape glued to a rotated guide.

If the lines don't intersect, the function returns a divide-by-zero error (#DIV/0!), 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 function

See also

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.

Syntax

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

<i>x</i> 1	The x -coordinate of a point on the first line.
y1	The <i>y</i> -coordinate of a point on the first line.
angle1	The value of the Angle cell for the first line.
<i>x</i> 2	The <i>x</i> -coordinate of a point on the second line.
<i>y</i> 2	The <i>y</i> -coordinate of a point on the second line.
angle2	The value of the Angle cell for the second line.

Remarks

Microsoft Visio uses this function in the PinY cell of a shape glued to a rotated guide.

If the lines don't intersect, the function returns a divide-by-zero error (#DIV/0!), 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 function

See also

Rounds a number up to the next integer.

Syntax

INTUP(number)

Example 1

INTUP(3.2)

Returns 4.

Example 2

INTUP(-3.2)

Returns -3.

Example 3

INTUP(3)

Returns 3.

ISERR function

Returns TRUE if the value of *cellreference* is any error type except #N/A; otherwise, it returns FALSE. The ISERR function is used in formulas that refer to another cell.

Syntax

ISERR(cellreference)

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 function

See also

Returns TRUE if the value of *cellreference* is error type #N/A! (not available); otherwise, it returns FALSE. The ISERRNA function is used in formulas that refer to another cell.

Syntax

ISERRNA(cellreference)

Example 1

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

Returns FALSE because the value returned is available.

Example 2

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 function

Returns TRUE if the value of *cellreference* is any error type; otherwise, it returns FALSE. The ISERROR function is used in formulas that refer to another cell.

Syntax

ISERROR(cellreference)

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 function

See also

Returns TRUE if the value of *cellreference* 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.

Syntax

ISERRVALUE(cellreference)

Remarks

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.

Example 1

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

Returns 2 because the value returned is a #VALUE! error, and the expression instructs Microsoft 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	5 + 7
	(ISERRVALUE(Scratch.A1),2,Scratch.A1)	

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

KEYWORDS function

See also

Returns the text from the **Keywords** field of a document's properties (click **Properties** on the **File** menu).

Syntax

KEYWORDS ()

LISTSEP function

See also

Returns the list separator string for the current user locale.

Syntax

LISTSEP()

Example

SETF(GETREF(user.extent), "MAX(Width" & ListSep() & "Height)")

LN function

See also

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

Syntax

LN(number)

Example

LN(42)

Returns 3.7377.

LOC function

See also

Takes a point defined in one shape's local coordinates and returns the equivalent point expressed in the local coordinates of the shape associated with the formula. Local coordinates are measured from the lower-left corner of the shape's selection rectangle. Both shapes must be on the same page.

Syntax

LOC(point)

Example

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

In this expression, PNT converts a set of local coordinates in Sheet.5 to a point. (Sheet.5 is another shape on the same drawing page.) LOC then converts that point to an equivalent point in the current shape's local coordinate system, relative 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 (click **Special** on the **Format** menu).

LOCTOLOC function

See also

Returns a transformed point in local coordinates in the destination coordinate system.

Syntax

LOCTOLOC(srcPoint, srcRef, dstRef)

srcPoint A point in local coordinates in the source coordinate system.

srcRef A reference to a cell in the source object.

dstRef A reference to a cell in the destination object.

Remarks

The LOCTOLOC function converts a point from local coordinates in a source shape to local coordinates in a destination shape. You can use this function to construct a shape, for example, in terms of a point from another coordinate space. You can also use this function to transform a local point to page coordinates, or vice versa.

This function works even when the source and destination shapes are within groups. It also adjusts for rotation and flips in the intermediate transformation.

The source and destination coordinates must be on the same page.

Example

LOCTOLOC(PNT(LocPinX, LocPinY), Width, ThePage!PageWidth)

Converts the local pin of the shape associated with the formula to a point on the page.

LOCTOPAR function

See also

Returns a transformed point in parent coordinates in the destination coordinate system.

Syntax

LOCTOPAR(srcPoint, srcRef, dstRef)

srcPoint A point in local coordinates in the source coordinate system.

srcRef A reference to a cell in the source object.

dstRef A reference to a cell in the destination object.

Remarks

Converts a point from local coordinates in a source shape to parent coordinates in a destination shape. You can use the LOCTOPAR function to set parent coordinates in cells, such as PinX, PinY, BeginX, and BeginY in a shape using another point from another coordinate system.

This function works even when the source and destination shapes are within groups. It also adjusts for rotation and flips in the intermediate transformation.

The source and destination coordinates must be on the same page.

If the destination is a page, which has no parent, the result is expressed in the page's local coordinates.

Example

LOCTOPAR(PNT(LocPinX, LocPinY), Width, Sheet.4!Width)

Converts the local pin of the shape associated with the formula to parent coordinates of Sheet.4.

LOG10 function

See also

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

Syntax

LOG10(number)

Example

LOG10(42)

Returns 1.6232.

LOOKUP function

See also

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*.

Syntax

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

key The string that you want to look up.*list* The list in which you want to search.

delimiter The string to use as a delimiter within list. A delimiter string

can be more than one character in length and may include

multibyte characters. The default is a semicolon.

Remarks

The LOOKUP function 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.

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.

Example 1

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

Returns 1.

Example 2

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

Returns 0.

Example 3

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

Returns 3.

LOTUSNOTES function

See also

Beginning with Microsoft Visio 2002, this function is obsolete.

Remarks

In versions earlier than Visio 2002, you could return information read from Lotus Notes into your formula.

LOWER function

See also

Returns a string converted to lowercase.

Syntax

LOWER(expression)

expression A string, a cell reference, or an expression; the result is

converted to a string which is then converted to lowercase.

Remarks

The case conversion is locale specific, based on the current user settings.

Example

LOWER("mIxEd CAse")

Returns "mixed case".

LUM function

See also

Returns the value of a color's luminosity component.

Syntax

LUM(expression)

Remarks

The return value is a number in the range 0 to 240, inclusive. The input is an index of a color in the document's color table, or a reference to a cell that contains a color index. The function returns 0 for invalid input.

Example 1

LUM(Sheet.4!FillForegnd)

Returns the luminosity of Sheet.4's fill foreground color.

Example 2

LUM(6)

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

Example 3

LUM(HSL(10, 20, 30))

Returns 30.

MAGNITUDE function

See also

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:

 $SQRT((constant A * A)^2 + (constant B * B)^2)$

Syntax

MAGNITUDE(constantA,A,constantB,B)

Example

MAGNITUDE(1, 3, 1, 4)

Returns 5.

MANAGER function

See also

Returns the text from the **Manager** field of a document's properties (click **Properties** on the **File** menu).

Syntax

MANAGER ()

MASTERNAME function

See also

Returns a sheet's master name as a string, or the string, "<no master>" if the sheet doesn't have a master.

Syntax

MASTERNAME (langID_opt)

langID_opt

Optional argument. Use to specify a language for the string the function returns. Use 0 (default value) to specify the local language. Use 750 to specify universal language.

Remarks

If you pass an illegal language code, the local language is used.

MAX function

See also

Returns the largest number from a list. Largest means closest to positive infinity.

Syntax

MAX(number1,number2,...,numberN)

Example

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

Returns 13 in.

MIN function

See also

Returns the smallest number from a list. Smallest means closest to negative infinity.

Syntax

MIN(number1,number2,...,numberN)

Example

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

Returns 20 cm.

MINUTE function

See also

Returns an integer, 0 to 59, representing the minutes component of *datetime* or *expression*.

Syntax

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

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. The locale identifier to be used in

evaluating a nonlocal datetime. The locale identifier is a

number described in the system header files.

Remarks

The date component in *datetime* and *expression* is discarded.

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

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

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

Example 1

MINUTE("7/7/1999 3:45:24 PM")

Returns 45.

Example 2

MINUTE(TIMEVALUE("Jan. 25, 1999 12:07:45")+6 em.)

Returns 13.

Example 3

MINUTE(0.575)

Returns 48.

MODULUS function

See also

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

Syntax

MODULUS(number, divisor)

Remarks

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

Example 1

MODULUS(5, 1.4)

Returns 0.8.

Example 2

MODULUS(5, -1.4)

Returns -0.6.

Example 3

MODULUS(-5, 1.4)

Returns 0.6.

Example 4

MODULUS(-5, -1.4)

Returns -0.8.

MONTH function

See also

Returns an integer, 1 to 12, representing a month.

Syntax

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

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. The locale identifier to be used in

evaluating a nonlocal datetime. The locale identifier is a

number described in the system header files.

Remarks

The time component of *datetime* or *expression* is discarded.

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

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

The MONTH function also accepts a single number value for *expression* where the integer portion of the result represents the number of days since December 30, 1899.

Example 1

MONTH("May 30, 1999 3:45:24 PM")

Returns 5.

Example 2

MONTH(DATEVALUE("May 30, 1999")+7 ed.)

Returns 6.

Example 3

MONTH(35580.6337)

Returns 5.

NA function

See also

Returns the error value #NA!. NA is the error value that means "no value available."

Syntax

NA()

Remarks

Use the NA function in a ShapeSheet cell where information is missing, so that the cell's default value is not used in calculations.

NAME function

See also

Returns a sheet's name as a string.

Syntax

NAME (langID_opt)

langID_opt

Optional argument. Use to specify a language for the string the function returns. Use 0 (default value) to specify the local language. Use 750 to specify universal language.

Remarks

If you pass an illegal language code, the local language is used.

NOT function

See also

Returns TRUE (1) if *logicalexpression* is FALSE. Otherwise, it returns FALSE (0).

Syntax

NOT(logicalexpression)

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 function

See also

Returns the current date and time value.

Syntax

NOW()

Remarks

NOW is automatically recalculated every minute.

Example 1

NOW()

Returns the current date and time, such as 9/27/1999 12:03:30 PM.

Example 2

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

Returns the current date and time formatted as 27 Sep., 1999 12:03.

Example 3

NOW()+2EW.

Returns the current date and time plus two elapsed weeks, such as 10/11/99 12:03:30 PM.

NURBS function

Returns a nonuniform rational B-spline (NURBS). This function is used in the E cell in the NURBSTo geometry rows.

Syntax

NURBS(knotLast, degree, xType, yType, x1, y1, knot1, weight1, ...)

knotLast The last knot.

degree The spline's degree.

xType Specifies how to interpret the x input data. If xType is 0, the

input *x*-data is interpreted as a percentage of Width. If *xType* is

1, the input *x*-data is interpreted as a local coordinate.

yType Specifies how to interpret the *y* input data. If *yType* is 0, the

input *y*-data is interpreted as a percentage of Height. If yType is 1, the input *y*-data is interpreted as a local coordinate. Specifies how to interpret the *x* and *y* input data, respectively. If *xType* or *yType* is 0, the input data is interpreted as a percentage of the height. If *xType* or *yType* is 1, the input data is interpreted as a

local coordinate.

x1 An *x*-coordinate. *y1* A *y*-coordinate.

knot1 A knot on the B-spline.*weight1* A weight on the B-spline.

Remarks

For every *x* argument, there must be a *y* argument, otherwise an error is returned.

You must specify an *x*, *y*, *knot*, and *weight* argument, otherwise an error is returned.

OPENFILE function

See also

Opens a Microsoft Visio document, if it's not already open, and activates the document window.

Syntax

OPENFILE("filename")

Remarks

Multiple OPENFILE function calls are queued and executed in order of evaluation. If the current Visio document is activated for visual (in-place) editing, a new Visio instance is launched with the requested file name.

This function always returns FALSE.

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

Example

OPENFILE("basic.vss")

Opens the specified stencil file Basic.vss in a new window, or activates the window if the stencil is already open.

OPENGROUPWIN function

See also

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

Syntax

OPENGROUPWIN()

Remarks

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

OPENPAGE function

See also

Displays a page name in new window.

Syntax

OPENPAGE("pagename")

Remarks

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

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

OPENSHEETWIN function

See also

Opens the ShapeSheet window in a new window.

Syntax

OPENSHEETWIN()

Remarks

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

OPENTEXTWIN function

See also

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

Syntax

OPENTEXTWIN()

Remarks

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

OR function

See also

Returns TRUE (1) if any of the logical expressions are TRUE.

Syntax

OR(logicalexpression1,logicalexpression2,...,logicalexpressionN)

Remarks

Any expression that evaluates to a non-zero value is considered TRUE. If all of the logical expressions are FALSE or equal 0, this function returns FALSE.

Example

OR(Height > 1, PinX > 1)

Returns TRUE (1) if either expression is TRUE. Returns FALSE (0) only if both expressions are FALSE.

PAGECOUNT function

See also

Returns the number of foreground pages in a document.

Syntax

PAGECOUNT ()

PAGENAME function

See also

Returns the page name as a string.

Syntax

PAGENAME (langID_opt)

langID_opt

Optional argument. Use to specify a language for the string the function returns. Use 0 (default value) to specify the local language. Use 750 to specify universal language.

Remarks

If you pass an illegal language code, the local language is used.

PageNumber function

Returns the number of the page that contains the shape.

Syntax

PageNumber()

Remarks

If the current page is a background page (unnumbered), or if an error occurs, the PageNumber function returns zero.

PAR function

See also

Returns the *x*,*y* coordinates of a point in the coordinate system of the shape's parent.

Syntax

PAR(point)

Remarks

In Microsoft Visio, a point is a single value that embodies a pair of x- and ycoordinates. 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.

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 or group that contains the current shape.

PI function

See also

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

Syntax

PI()

Example

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

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

PLAYSOUND function

See also

Plays a sound file or system sound.

Syntax

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

filename The name of the sound file you want to play.

alias A system sound represented by an alias.

isAlias Specifies whether the preceding expression is an alias or file

name; use a non-zero value to specify an alias.

beep Specifies whether Microsoft Visio beeps when sound can't be

played; use a non-zero number to beep.

synch Determines whether sounds are played asynchronously (0) or

synchronously (1).

Remarks

Sounds should usually be played asynchronously so Visio can continue processing while the sound is being played. To string several sounds together, play them synchronously or some might fail to play.

Example 1

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

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

Example 2

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

Plays the system exclamation sound asynchronously with no warning beep.

PNT function

See also

Returns the point represented by the coordinates x and y as a single value.

Syntax

PNT(x,y)

Remarks

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 function

See also

Returns the *x*-coordinate of a point.

Syntax

PNTX(point)

Example

PNTX(PNT(7,12))

Returns 7.

PNTY function

See also

Returns the *y*-coordinate of a point.

Syntax

PNTY(point)

Example

PNTY(PNT(7,12))

Returns 12.

POLYLINE function

See also

Returns a polyline. This function is used in the A cell of PolyLineTo geometry rows.

Syntax

POLYLINE(*xType*, *yType*, *x1*, *y1*...)

xType Specifies how to interpret the x input data. If xType is 0, the

input *x*-data is interpreted as a percentage of Width. If *xType* is

1, the input *x*-data is interpreted as a local coordinate.

yType Specifies how to interpret the *y*-input data. If *yType* is 0, the

input y-data is interpreted as a percentage of Height. If *yType* is

1, the input *y*-data is interpreted as a local coordinate.

*x*1 An *x*-coordinate.

y1 A *y*-coordinate.

Remarks

For every *x* argument, there must be a *y* argument, otherwise an error is returned.

Example

POLYLINE (0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0)

Returns a rectangle using Width x Height.

POW function

See also

Returns a number raised to the power of an exponent.

Syntax

POW(number, exponent)

Remarks

Both *number* and *exponent* may be non-integers, and they may be negative. If *number* is not 0 and *exponent* is 0, this function returns 1. If *number* is 0 and *exponent* is negative, this function returns 0.0. If both *number* and *exponent* are 0, or if *number* is negative and *exponent* is not an integer, this function returns 0.0.

Example

POW(5,2)

Returns 25.

RAD function

See also

Converts the value of an angle from degrees to radians.

Syntax

RAD(angle)

Example

RAD(45)

Returns 0.7854.

RAND function

See also

Returns a random floating point number from 0.0 to 1.0. It returns a different number each time the function is evaluated, which is once per minute according to the system clock.

Syntax

RAND()

Remarks

Use this function to create animation effects by setting shapes' properties to random values.

Example

RAND()

Returns a decimal fraction such as 0.3503.

RECTSECT function

See also

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

Syntax

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

width Width of the rectangle.height Height of the rectangle.

x An *x*-coordinate. *y* A *y*-coordinate.

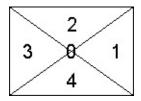
option Specifies how points that fall on the diagonals are treated. Set

the value to 0 to use the left and right sectors for points on a diagonal. Set the value to 1 to use the top and bottom sectors

for points on a diagonal.

Remarks

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 0 through 4 represent the center-point, right, top, left, and bottom respectively.



Example

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

Returns 4.

RED function

See also

Returns the red component of a color. The return value is an integer in the range of 0 to 255, inclusive. The input is an index of a color in the document's color table, an expression that resolves to a custom color (like RGB or HSL), or a reference to a cell that contains a color index or color result. The function returns 0 for invalid input.

Syntax

RED(expression)

Remarks

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, this function returns 0 (black).

Example 1

RED(22)

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

Example 2

RED(Char.Color)

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

Example 3

RED(RGB(10, 20, 30))

Returns 10.

REF function

The REF function returns the error value #REF!.

Syntax

REF()

Example

REF()

Returns #REF!.

RGB function

See also

Returns a value representing an index in the document's color palette. It 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.

Syntax

RGB(red,green,blue)

Remarks

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

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

Example 1

RGB(0,0,255)

Returns the index for the color blue.

Example 2

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

Returns the index for a color whose red component mirrors Sheet.1's fill foreground.

ROUND function

See also

Rounds a number to the precision represented by *numberofdigits*.

Syntax

ROUND(number,numberofdigits)

Remarks

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 an integer. If *numberofdigits* is less than 0, *number* is rounded by *numberofdigits* to the left of the decimal.

Example 1

ROUND(123.654,2)

Returns 123.65.

Example 2

ROUND(123.654,0)

Returns 124.

Example 3

ROUND(123.654,-1)

Returns 120.

RUNADDON function

See also

Executes an add-on or passes code to be executed to the Microsoft Visual Basic for Applications (VBA) project of the document containing the operand.

Syntax

RUNADDON("string")

string

The name of an add-on or a line of 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.

Remarks

If the project cannot execute *string*, Microsoft Visio runs the add-on named *string*. If no add-on named *string* can be found, Visio does nothing and reports no error.

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

Beginning with Microsoft Visio 2002, evaluation of RUNADDON does not create a new project. In earlier versions, a VBA project was created in the operand's document if no project existed already. For example, in Visio 2002, the evaluation of =RUNADDON("msgbox ""hello""") shows the message box if the document contains a project; otherwise, it does nothing. You can create a document project by opening the Visual Basic Editor when the document is open.

Note VBA code invoked as a result of a Visio instance evaluating a RUNADDON operand in a formula should not use the **Document.Close** method to close the document containing the function's cell because an application error results and Visio terminates.

If you need to close the document containing the cell that uses the RUNADDON

function, use one of the following techniques:

Close the document from non-VBA code.

Close the document from a project other than the one that is closing.

Post window messages to close windows into the document rather than closing the document.

Example 1

RUNADDON("Calendar.exe")

Launches the Calendar.exe add-on.

Example 2

RUNADDON("summation 1, 2, 3")

Calls the VBA procedure named summation and passes three arguments.

Example 3

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

Prints "xyz" in the VBA Immediate window.

Example 4

RUNADDON("ThisDocument.Save")

Saves the document containing the function.

RUNADDONWARGS function

Runs *filename* and passes the command line *arguments* to the program as a string.

Syntax

RUNADDONWARGS("filename","arguments")

Remarks

In practice, *arguments* should be 50 characters or fewer. Use the RUNADDONWARGS function to bind a program, such as an add-on, to an Action cell.

In earlier versions of Microsoft Visio, this function appears as _RUNADDONWARGS. Visio application versions 4.0 and later accept either style.

Note VBA code invoked as a result of the Visio instance evaluating a RUNADDONWARGS function in a formula should not close the document containing the cell using the function because an application error results and Visio terminates.

If you need to close the document containing the cell that uses the RUNADDONWARGS function, use one of the following techniques:

Close the document from non-VBA code.

Close the document from a project other than the one that is closing.

Post window messages to close windows into the document rather than closing the document.

Example

RUNADDONWARGS("GRAPHMKR.EXE","/GraphMaker=Stack")

Launches the add-on Graphmkr.exe and passes it the argument /GraphMaker=Stack.

SAT function

See also

Returns the value of a color's saturation component.

Syntax

SAT(expression)

Remarks

The return value is a number in the range 0 to 240, inclusive. The input is an index of a color in the document's color table, an expression that resolves to a custom color (like RGB or HSL), or a reference to a cell that contains a color index or color result. The function returns 0 for invalid input.

Example 1

SAT(Sheet.4!FillForegnd)

Returns the saturation of Sheet.4's fill foreground color.

Example 2

SAT(8)

Returns 240 if the document uses the default Microsoft Visio color palette, where dark red is the color at index 8.

Example 3

SAT(HSL(10, 20, 30))

Returns 20.

SECOND function

See also

Returns an integer, 0 to 59, representing the seconds component of *datetime* or *expression*.

Syntax

SECOND("datetime"|expression[,lcid])

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. The locale identifier to be used in

evaluating a nonlocal *datetime*. The locale identifier is a

number described in the system header files.

Remarks

The date component in *datetime* or *expression* is discarded.

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

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

Example 1

SECOND("05/30/1997 3:45:32 PM")

Returns 32.

Example 2

SECOND(TIMEVALUE("May 30, 1996 12:07:45")+7 es.)

Returns 52.

Example 3

SECOND(0.6337)

Returns 32.

SETF function

See also

Sets a cell's formula.

Syntax

```
SETF(GETREF(cell), formula)
SETF(GETREF(cell), "formula")
SETF(GETREF(cell), """formula""")
SETF("cell",formula)
SETF("cell","formula")
SETF("cell","""formula""")
```

Remarks

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.

The target cell must be specified using a GETREF() reference or as a string to avoid circularity. Using GETREF is preferred, because Microsoft Visio can adjust references when the shape is moved to a different document.

If *cell* is not specified using GETREF or as 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.

Example 1

SETF(GETREF(Scratch.A1), 1.5in.*6+1ft.)

Sets the formula of Scratch.A1 to 21 in.

Example 2

SETF(GETREF(Scratch.A1), "1.5in.*6+1ft.")

Sets the formula of Scratch. A1 to the expression 1.5 in.*6+1ft.

Example 3

SETF(GETREF(Scratch.A1), """Say """"ahh"""""")

Sets the formula of Scratch.A1 to the string "Say ""ahh"" which evaluates to Say "ahh".

SHAPETEXT function

See also

Retrieves the text from a shape.

Syntax

SHAPETEXT (*shapename!TheText,flag*) shapename!TheText A reference to the cell named TheText in the target shape. Shapename! is the name of the shape from which you want to retrieve the text. flag Optional. A bit that specifies the format of the text. The default flag (0) shows the text exactly as it is shown in the shape.

Remarks

You can use any combination of the following flags with the SHAPETEXT function.

Flag	Description
0	Show text exactly as shown in shape.
1	Include discretionary hyphens.
2	Don't include expanded text in fields.
4	Convert tabs to a single space.
8	Convert tabs to a set of spaces.
16	Convert carriage returns and line feeds to spaces.
32	Convert typographer quotes to regular quotes.
64	Convert adjacent white space to a single space.

Example 1

SHAPETEXT(sheetN!theText)

Returns the text of the shape named sheetN, exactly as it is shown in the shape.

Example 2

SHAPETEXT(theText)

Returns the text of the current shape exactly as it is shown in the shape.

Example 3

SHAPETEXT(theText, 84)

Returns the text of the current shape. It also converts adjacent white space to a single space (64), converts carriage returns and line feeds to spaces (16), and converts tabs to a single space (4). The sum of these flags is 84.

SIGN function

See also

Returns a value that represents the sign of a number.

Syntax

SIGN(number,fuzz)

number The number for which you want to determine the sign.

fuzz Optional. Specifies how close to zero the number must be in

order to be considered equal to zero.

Remarks

The SIGN function returns 1 if *number* is positive, 0 if *number* is zero, or -1 if *number* is negative.

Fuzz helps avoid floating-point roundoff errors when a calculation is almost zero. If you do not specify a fuzz value, Microsoft Visio uses 1E-9 (0.00000001). You may want to supply a different value when you scale drawings or when you want an exact comparison.

Example 1

SIGN(-5)

Returns -1.

Example 2

SIGN(0)

Returns 0.

Example 3

SIGN(0.00000000001)

Returns 0.

Example 4

SIGN(0.0000000001,0)

Returns 1.

SIN function

Returns the sine of an angle.

Syntax

SIN(angle)

Remarks

The SIN function uses radians unless a unit of measure is specified with angle.

Example

SIN(45 degrees)

Returns 0.7071.

SINH function

Returns the hyperbolic sine of an angle.

Syntax

SINH(angle)

Remarks

The SINH function uses radians unless a unit of measure is specified with angle.

Example

SINH(45 degrees)

Returns 0.8687.

SQRT function

See also

Returns the square root of a number.

Syntax

SQRT(number)

Remarks

If *number* is negative, the SQRT function returns the error value #NUM!.

Example

SQRT(2)

Returns 1.4142.

STRSAME function

Determines whether strings are the same. It returns TRUE if they are the same and FALSE if they aren't. To compare multi-byte strings or to do comparisons using case rules for a specific locale use the STRSAMEEX function.

Syntax

STRSAME ("string1", "string2", ignoreCase)

string1 The first string to compare.string2 The second string to compare.

ignoreCase Optional. A Boolean variable. TRUE to ignore the case and

FALSE to compare the case. The default is FALSE.

Example 1

STRSAME("cat","dog")

Returns FALSE.

Example 2

STRSAME("cat","cat")

Returns TRUE.

Example 3

STRSAME("cat", "cat", TRUE)

Returns TRUE.

Example 4

STRSAME("cat","CAT")

Returns FALSE.

Example 5

STRSAME("cat","CAT", TRUE)

Returns TRUE.

Example 6

STRSAME("cät","CAT", TRUE)

Returns FALSE.

Example 7

STRSAME("cät","CÄT", TRUE)

Returns TRUE.

STRSAMEEX function

See also

Determines whether strings are the same. It returns TRUE if they are the same and FALSE if they aren't. Use this function to compare multi-byte strings or to do comparisons using case rules for a specific locale.

Syntax

STRSAMEEX ("string1", "string2", localeID, flag)		
string1	The first string to compare.	
string2	The second string to compare.	
localeID	The locale ID code.	
flag	A bit that specifies the type of comparison.	

Remarks

You can use a combination of any of the following flags with the STRSAMEEX function.

Flag	Description
1	Ignore case.
2	Ignore non-spacing characters.
4	Ignore symbols.
4096	Treat punctuation the same as symbols.
65536	Don't differentiate between Hiragana and Katakana
	characters.
131072	Don't differentiate between a single-byte character and the same character as a double-byte character.

SUBJECT function

See also

Returns the text from the **Subject** field of a document's properties (click **Properties** on the **File** menu).

Syntax

SUBJECT ()

SUM function

See also

Returns the sum of a list of numbers.

Syntax

SUM(number1,number2,...,number14)

Example

SUM(5,7,12)

Returns 24.

TAN function

See also

Returns the tangent of an angle.

Syntax

TAN(angle)

Remarks

The TAN function uses radians unless a unit of measure is specified with angle.

Example

TAN(45 degrees)

Returns 1.

TANH function

Returns the hyperbolic tangent of an angle.

Syntax

TANH(angle)

Remarks

The TANH function uses radians unless a unit of measure is specified with *angle*.

Example

TANH(-45)

Returns -0.6558.

TEXTHEIGHT function

Returns the height of the composed text in a shape where no text line exceeds *maximumwidth*.

Syntax

TEXTHEIGHT(shapename!TheText,maximumwidth)

shapename! A name that refers to the text of a shape other than the current

shape. Default equals the current shape.

TheText A cell that gets triggered when the associated shape's text

composition changes.

maximumwidth Optional. The maximum width of the text block.

Remarks

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. This function is commonly used to adjust the height of a shape to fit the text it contains.

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 function

See also

Returns the width of the composed text in a shape.

Syntax

TEXTWIDTH(shapename!TheText,maximumwidth)

shapename! A name that refers to the text of a shape other than the current

shape. Default equals the current shape.

TheText A cell that gets triggered when the associated shape's text

composition changes.

maximumwidth Optional. The maximum width of the text block.

Remarks

The TEXTWIDTH function is commonly used to adjust the width of a shape to fit the text it contains.

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.

Example

TEXTWIDTH(TheText)

Returns the total length of the text in the current shape.

TIME function

See also

Returns the time represented by *hour*, *minute*, and *second*.

Syntax

TIME(hour,minute,second)

Example 1

TIME(15,30,30)

Returns the value representing 3:30:30 PM.

Example 2

FORMAT(TIME(15,30,30),"HH:mm")

Returns the value representing 15:30.

Example 3

TIME(15,30,30) + 8 eh.

Returns the value representing 11:30:30 PM.

TIMEVALUE function

See also

Returns the time value represented by *datetime* or *expression*.

Syntax

TIMEVALUE("datetime"|expression[,lcid])

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. The locale identifier to be used in

evaluating a nonlocal datetime. The locale identifier is a

number described in the system header files.

Remarks

Any date component in *datetime* or *expression* is discarded.

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

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

Example 1

TIMEVALUE("6:00 AM")

Returns the value representing 6:00 AM.

Example 2

TIMEVALUE("14:30")+4 eh.+30 em.

Returns the value representing 7:00:00 PM.

Example 3

TIMEVALUE("11 AM, July 1, 1997")

Returns the value representing 11:00 AM.

Example 4

TIMEVALUE(0.6337)

Returns the value representing 3:12:32 PM.

Example 5

TIMEVALUE("7:89")

Returns a #VALUE! error.

TITLE function

See also

Returns the text from the **Title** field of a document's properties (click **Properties** on the **File** menu).

Syntax

TITLE ()

TRUNC function

See also

Returns a number truncated to *numberofdigits*.

Syntax

TRUNC(number,numberofdigits)

Remarks

If *numberofdigits* is greater than 0, *number* is truncated to *numberofdigits* to the right of the decimal. If *numberofdigits* is 0, *number* is truncated to an integer. If *numberofdigits* is less than 0, *number* is truncated to *numberofdigits* to the left of the decimal.

Example 1

TRUNC(123.654,2)

Returns 123.65.

Example 2

TRUNC(123.654,0)

Returns 123.

Example 3

TRUNC(123.654,-1)

Returns 120.

TYPE function

See also

Returns an object's internal type as a number.

Syntax

TYPE ()

Remarks

The following table lists internal object types a numbers.

Object	Number
Page	70
Group	71
Shape	72
Style	74
Guide	77
Foreign	78
Document	79

TYPEDESC function

See also

Returns a descriptive string for an object's internal type.

Syntax

TYPEDESC ()

Remarks

The following table lists strings for internal objects.

Object	String
Page	"Page Sheet"
Group	"Group"
Shape	"Shape"
Style	"Style"
Guide	"Guide"
Foreign (metafile)	"Metafile"
Foreign (bitmap)	"Bitmap"
Foreign (OLE link)	"OLE Linked Object"
Foreign (OLE embedded object)	"OLE Embedded Object"
Foreign (OLE control)	"Control"
Document	"Document Sheet"
Unknown	"Shape"

UPPER function

See also

Returns a string converted to uppercase.

Syntax

UPPER(expression)

expression A string, a cell reference, or an expression; the result is

converted to a string which is then converted to uppercase.

Remarks

The case conversion is locale specific, based on the current user settings.

Example

UPPER("mIxEd CAse")

Returns "MIXED CASE".

USE function

Applies the line pattern, fill pattern, or line end called *name* to the shape when placed in the LinePattern, FillPattern, BeginArrow, or EndArrow cell. This function always returns 254.

Syntax

USE("name")

name

Any string that is a valid master name.

Remarks

If a master with *name* is present on a document 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 function

See also

Evaluates one of the two expressions depending on the value of *state*.

Syntax

USERUI(state, default expression, user expression)

Remarks

If *state* is 0, the USERUI function evaluates the *defaultexpression*. If *state* is 1, it evaluates the *userexpression*.

Example

USERUI(1,if(Width>6in.,6in.,Width),Width*0.75)

Evaluates the expression Width*.075 and returns the result.

WEEKDAY function

See also

Returns an integer, 1 to 7, representing the weekday in *datetime* or *expression*.

Syntax

WEEKDAY("datetime"|expression[,lcid])

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. The locale identifier to be used in

evaluating a nonlocal datetime. The locale identifier is a

number described in the system header files.

Remarks

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, the function returns a #VALUE! error.

The WEEKDAY function also accepts a single number value for *expression* where the integer portion of the result represents the number of days since December 30, 1899.

Example 1

WEEKDAY("May 30, 1999")

Returns 5.

Example 2

WEEKDAY(DATEVALUE("May 30, 1999")+2 ed.)

Returns 7.

Example 3

WEEKDAY(35880.6337)

Returns 5.

YEAR function

See also

Returns an integer representing the Gregorian year in *datetime* or *expression* formatted according to the short date style set by the system's current Regional Settings.

Syntax

YEAR("datetime"|expression[,lcid])

datetime Any string commonly recognized as a date and time or a

reference to a cell containing a date and time.

expression Any expression that yields a date and time.

lcid Optional argument. The locale identifier to be used in

evaluating a nonlocal datetime. The locale identifier is a

number described in the system header files.

Remarks

The time component in *datetime* or *expression* is discarded.

No rounding is done. If *datetime* is missing or cannot be interpreted as a valid date or time, returns an error.

The YEAR function also accepts a single number value for *expression* where the integer portion of the result represents the number of days since December 30, 1899.

Example 1

YEAR("10/27/1999 3:45:24 PM")

Returns 1999.

Example 2

YEAR(DATEVALUE("Dec. 25, 1998") + 7 ed.)

Returns 1999.

Example 3

YEAR(35580.6337)

Returns 1997.

ArcTo row (Geometry section)

Contains the *x*- and *y*-coordinates and bow of a circular arc.

An ArcTo row contains the following cells.

Cell	Description
<u>X</u>	The x -coordinate of the ending vertex of an arc.
$\underline{\mathbf{Y}}$	The y -coordinate of the ending vertex of an arc.
A	The distance from the arc's midpoint to the midpoint of its chord.

Remarks

Arcs drawn in Microsoft Visio are elliptical arcs, even if they are based on a circle. By default, drawn arcs are represented by an EllipticalArcTo row in a ShapeSheet window. To show an ArcTo row in a ShapeSheet window, you must draw an arc, and then change the EllipticalArcTo row type to an ArcTo row type; in effect you are changing an elliptical arc to a circular arc.

To change a row type, right-click a row, and then click **Change Row Type** on the shortcut menu.

Connections.Row row (Connection Points section)

See also

Contains the *x*- and *y*-coordinates, horizontal and vertical direction, and type for a single connection point on a shape. Coordinates of connection points are measured from the origin of the shape.

A Connections.Row row contains the following cells.

Cell	Description
X	The <i>x</i> -coordinate for a connection point in local coordinates.
Y	The <i>y</i> -coordinate for a connection point in local coordinates.
<u>DirX/A</u>	The <i>x</i> -component for the required alignment vector of a matching connection point. It is also used to orient the attached leg of a dynamic connector. This cell takes a floating point value.
DirY/B	The <i>y</i> -component for the required alignment vector of a matching connection point. It is also used to orient the attached leg of a dynamic connector. This cell takes a floating point value.
Type/C	The connection point type (0 = inward; 1 = outward; 2 = inward

+ outward).

A scratch cell that you can use for entering or testing formulas. To access this cell, right-click a row, and then click **Change Row Type** on the shortcut menu.

Remarks

Cells in the Connections.Row row are labeled DirX/A, DirY/B, and Type/C because these rows can be extended or non-extended rows.

Most connection points (all connection points created through the user interface) are non-extended and have DirX, DirY and Type cells. Their row type is **visTagCnnctPt** or **visTagCnnctNamed.**

In non-extended rows the DirX and DirY cells together define a direction vector that influences the rotation of shapes involved in connections using the connection point. If both are zero the point is directionless. Connection points are of type

Inward (0), which means that shapes glue to them. This is the default.

Outward (1), which means these connection points will glue to inward connection points.

Both inward and outward (2), in which case the direction is the inward direction, which is reversed if used as an outward connection.

Extended rows have A, B, C, and D cells and behave like directionless non-extended rows of type Inward. Extended rows are not commonly used, but you might use them to associate data with a connection point in the A, B, C and D, cells. Their row type is **visTagCnnctPtABCD** or **visTagCnnctNamedABCD**. Extended rows can be identified by the presence of a formula in the D cell.

You can reference connection row cells by their row name, which appears in a ShapeSheet window in red text. To change the row name, click it, and then type *Custom*, for example, to create the row name Connections.Custom. You can then reference the X cell using Connections.Custom.X, for example, or Connections.X1 if you want to use the row number.

The row name you enter must be unique within the section. When you create a

name for one row in the Connection Points section, Microsoft Visio names all the rows in the section with the default name, Connections.Row_*n*.

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

Ellipse row (Geometry section)

Contains the *x*- and *y*-coordinates of the ellipse's center point and two points on the ellipse.

An Ellipse row contains the following cells.

Cell	Description
X	The x -coordinate of the center point.
<u>Y</u>	The <i>y</i> -coordinate of the center point.
A	The x-coordinate of one point on the ellipse; paired with <i>y</i> -coordinate represented by the B cell.
<u>B</u>	The <i>y</i> -coordinate of one point on the ellipse; paired with x-coordinate represented by the A cell.
<u>C</u>	The <i>x</i> -coordinate of another point on the ellipse; paired with <i>y</i> -coordinate represented by the D cell.
D	The <i>y</i> -coordinate of another point on the ellipse; paired with <i>y</i> -coordinate represented by the C cell.

Remarks

A geometry section that contains an Ellipse or an InfiniteLine row should not

contain any other rows.

EllipticalArcTo row (Geometry section)

See also

Contains *x*- and *y*-coordinates of an elliptical arc's endpoint, *x*- and *y*-coordinates of the control points on the arc, angle from the *x*-axis to the ellipse's major axis, and ratio between the ellipse's major and minor axes.

An EllipticalArcTo row contains the following cells.

Cell	Description
<u>X</u>	The x -coordinate of the ending vertex on an arc.
Y	The <i>y</i> -coordinate of the ending vertex on an arc.
A	The <i>x</i> -coordinate of the arc's control point—a point on the arc. The control point is best located about halfway 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.
<u>B</u>	The <i>y</i> -coordinate of an arc's control point.
C	The angle of an arc's major axis relative to the <i>x</i> -axis of its parent

 $\underline{\mathbf{D}}$

The ratio of an arc's major axis to its minor axis. Despite the usual meaning of these words, the "major" axis does not have to be greater than the "minor" axis, so this ratio does not have to be greater than 1. Setting this cell to a value less than or equal to 0 or greater than 1000 can lead to unpredictable results.

Hyperlink.Row row (Hyperlinks section)

See also

Contains the information for a single hyperlink associated with a shape. A shape will contain one Hyperlink.Row row for each hyperlink.

A Hyperlink.Row row contains the following cells. For more details, see the specific cell topics.

Cell	Description
Description	A descriptive text string for a hyperlink.
<u>Address</u>	A URL address, DOS file name, or UNC path to which to
	jump.
SubAddress	A location within the target document to link to.
ExtraInfo	A string that passes information to be used in resolving a
	URL.
<u>Frame</u>	The name of a frame to target when Microsoft Visio is open as
	an Active document in an ActiveX container. The default is an
	empty string.
NewWindow	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.

<u>Default</u> The default hyperlink for a shape or page.

Remarks

You can assign meaningful names to hyperlink rows. To change the row name, click it, and then type Marketing, for example, to create the row name Hyperlink.Marketing. You can then reference these cells by their row name, which appears in the ShapeSheet window in red text. For example, in the above named row you can reference the Description cell using Hyperlink.Marketing.Description.

The row name you enter must be unique within the section.

InfiniteLine row (Geometry section)

Contains the *x*- and *y*-coordinates of two points on an infinite line.

An InfiniteLine row contains the following cells.

Cell	Description
X	An x -coordinate of a point on the infinite line; paired with y -coordinate represented by the Y cell.
Y	A <i>y</i> -coordinate of a point on the infinite line; paired with x -coordinate represented by the X cell.
<u>A</u>	An x -coordinate of a point on the infinite line; paired with y -coordinate represented by the B cell.
<u>B</u>	A <i>y</i> -coordinate of a point on an infinite line; paired with <i>x</i> -coordinate represented by the A cell.

Remarks

A geometry section that contains an Ellipse or InfiniteLine row should not contain any other rows.

LineTo row (Geometry section)

See also

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

A LineTo row contains the following cells.

Cell	Description
X	The x -coordinate of the ending vertex of a straight line
	segment.
Y	The <i>y</i> -coordinate of the ending vertex of a straight line segment.

MoveTo row (Geometry section)

See also

Contains the *x*- and *y*-coordinates of the first vertex of a shape, or represents the *x*- and *y*-coordinates of the first vertex after a break in a path.

A MoveTo row contains the following cells.

Cell	Description
X	If the MoveTo row is the first row in the section, the X cell represents the <i>x</i> -coordinate of the first vertex of a shape. If the MoveTo row appears between two rows, the X cell represents the <i>x</i> -coordinate of the first vertex after the break in the path.
Y	If the MoveTo row is the first row in the section, the Y cell represents the <i>y</i> -coordinate of the first vertex of a shape. If the MoveTo row appears between two rows, the Y cell represents the <i>y</i> -coordinate of the first vertex after the break in the path.

Remarks

The MoveTo row contains the *x*- and *y*-coordinates of the first vertex for the

shape if the MoveTo row is the first row in the section. 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.

A geometry section must begin with a MoveTo row, but you can also use the MoveTo row to represent a gap in the stroking of a shape's path. However, when the path is used to define the boundary of a filled region, this gap is interpreted as a straight line segment. To insert such a gap, insert a row between two rows and change the row type to MoveTo. If the MoveTo row is between two rows, it contains the *x*- and *y*-coordinates of the first vertex of the line after the break in the shape's path.

NURBSTo row (Geometry section)

See also

Contains the *x*- and *y*-coordinates, position of the second to last knot, position of the last weight, position of the first knot, position of the first weight, and the formula for a nonuniform rational B-spline (NURBS).

A NURBSTo row contains the following cells.

Cell	Description
X	The x -coordinate of the last control point of a NURBS.
$\underline{\mathbf{Y}}$	The <i>y</i> -coordinate of the last control point of a NURBS.
<u>A</u>	The second to the last knot of the NURBS.
<u>B</u>	The last weight of the NURBS.
<u>C</u>	The first knot of the NURBS.
<u>D</u>	The first weight of the NURBS.
<u>E</u>	A NURBS formula.

Remarks

NURBS is a common way to represent a curve mathematically. You can create a NURBS with the **Freeform** tool.

PolylineTo row (Geometry section)

Contains *x*- and *y*-coordinates of the last point of a polyline and a polyline formula.

A PolylineTo row contains the following cells.

Cell	Description
X	The x -coordinate of the ending vertex of a polyline.
<u>Y</u>	The <i>y</i> -coordinate of the ending vertex of a polyline.
<u>A</u>	The polyline formula.

Remarks

Lines represented as a Polyline row are equivalent to lines represented as a sequence of LineTo rows, but a Polyline row is more efficient. You can change a PolylineTo row to a LineTo row so you can easily see the shape geometry. To do this, right-click the PolylineTo row, and then click **Expand Row** on the shortcut menu.

To change a row type to a PolylineTo row, right-click the row, and then click **Change Row Type** on the shortcut menu.

Prop.Name row (Custom Properties section)

See also

Contains the information for a single custom property associated with a shape. A shape will contain one Prop.Name row for each custom property.

A Prop.Name row contains the following cells. For more details, see the specific cell topics.

Cell	Description
<u>Label</u>	The Label cell specifies the label that appears to users in the Custom Properties dialog box or window.
Prompt	The Prompt cell specifies descriptive or instructional text that appears to users in the Custom Properties dialog box or window when the property is selected.
<u>Type</u>	Specifies a data type for the custom property value: string (0), a fixed list (1), a number (2), a Boolean value (3), a variable list (4), a date or time (5), a duration (6), or a currency (7).
<u>Format</u>	Specifies the formatting of a custom property. For details see the Format cell.

<u>Value</u> Contains the property's value as entered in the **Custom**

Properties dialog box or window.

SortKey Specifies a key by which items in the **Custom Properties**

dialog box or window are listed.

<u>Invisible</u> Specifies whether the custom property is visible in the

Custom Properties dialog box or window. TRUE = not

visible; FALSE = visible.

Ask Determines whether a user is gueried to enter custom

property information for a shape when an instance is

created or the shape is duplicated or copied.

Remarks

You can reference these cells by their row name, which appears in a ShapeSheet window in red text. To change the row name, click it, and then type Price, for example, to create the row name Prop.Price. You can then reference the Label cell using Prop.Price.Label.

The row name you enter must be unique within the section.

SplineKnot row (Geometry section)

Contains *x*- and *y*-coordinates for a spline's control point and a spline's knot.

A SplineKnot row contains the following cells.

Cell	Description
X	The <i>x</i> -coordinate of a control point.
<u>Y</u>	The <i>y</i> -coordinate of a control point.
A	One of the spline's knots (other than the last one or the first two).

Remarks

Microsoft 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 MoveTo row, to indicate the first control point of the spline. The preceding row can be a LineTo, ArcTo, NURBSTo, PolylineTo, or EllipticalArcTo row if the spline follows a segment of that type.

SplineStart row (Geometry section)

Contains *x*- and *y*-coordinates for a spline's second control point, its second knot, its first knot, the last knot, and the degree of the spline.

A SplineStart row contains the following cells.

Cell	Description
X	The x -coordinate of a spline's second control point.
<u>Y</u>	The y -coordinate of a spline's second control point.
<u>A</u>	The second knot of the spline.
<u>B</u>	The first knot of a spline.
<u>C</u>	The last knot of a spline.
<u>D</u>	The degree of a spline (an integer from 1 to 25).

Remarks

Microsoft 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 MoveTo row, to indicate the first control point of the spline. The preceding row can be a LineTo, ArcTo, NURBSTo, PolylineTo, or EllipticalArcTo row if the spline

follows a segment of that type.

User.Row row (User-Defined Cells section)

See also

Contains the value and descriptive prompt for any user-defined cells in your solution. A shape will contain one User.Row row for each user-defined cell.

A User.Row row contains the following cells. For more details, see the cell topics.

Cell	Description
<u>Value</u>	The Value cell specifies a value for the corresponding user-defined cell.
<u>Prompt</u>	The Prompt cell specifies a descriptive prompt or comment for the user-defined cell.

Remarks

User-defined cells can be used for entering formulas or constants that are referred to by other cells or add-ons. Values in user-defined cells are portable, that is, if a shape that refers to a user-defined cell in one shape is copied to another shape that does not have the same user-defined cell, the cell is added to

the shape.

You can reference these cells by their row name, which appears in a ShapeSheet window in red text. You can assign meaningful names to User.Row rows. To change the row name, click it, and then type *Offset*, for example, to create the row name User.Offset. You can then reference the Prompt cell using User.Offset.Prompt.

The row name you enter must be unique within the section.

1-D Endpoints section

See also Rows

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.

Actions section

See also Rows

Defines custom command names that appear on an object's shortcut menu and specifies the actions that the commands take.

Alignment section

See also Rows

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.

Character section

See also Rows

Shows the formatting attributes for the shape's text, such as font, color, text style, case, position relative to the baseline, and point size.

Remarks

Each character style in a shape has a corresponding row in the Character section as viewed in the ShapeSheet window. The name of the row represents the number of characters the character style applies to.

Text formats can be set with the **Text** dialog box (click **Text** on the **Format** menu), by applying a text style, or by making an entry in a Character section cell.

Connection Points section

See also

Contains a row and cells for each connection point defined for the shape.

Remarks

You can define connection points in the Connection Points section in the ShapeSheet window or by using the **Connection Point** tool on the **Standard** toolbar.

If your shape has no Connection Points section you can insert a Connection Points section in the ShapeSheet window. On the **Insert** menu, click **Section**, and then click **Connection points**. To add connection points to an existing Connection Points section, right-click a row and click **Insert Row** on the shortcut menu.

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, and then click **Change Row Type** on the shortcut menu.

Controls section

See also Rows

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.

Custom Properties section

See also

Contains one row for each custom property and cells for associating data with a shape.

Remarks

Custom properties can be defined using the **Custom Properties** window or the cells in the Custom Properties section.

In the ShapeSheet window, you can define custom properties in the Custom Properties section. On the **Insert** menu, click **Section**, then click **Custom properties**. You can add as many Prop.Name rows as you need, assign meaningful names to the rows, and set cell values.

In the drawing window, you can define custom properties in the **Custom Properties** window. On the **View** menu, click **Custom Properties Window**. To access the **Define Custom Properties** dialog box, right-click the **Custom Properties** window, and then click **Define Properties** on the shortcut menu.

Document Properties section

See also Rows

Contains cells for a document that control preview quality, scope, and output format. You can also set these values using the **Properties** dialog box. Or, you can right-click a drawing in the **Drawing Explorer**, and then click **Properties** on the shortcut menu.

Events section

See also Rows

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

Fill Format section

See also Rows

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** dialog box, by applying a fill style, or by making an entry in a Fill Format cell.

Foreign Image Info section

See also Rows

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

Remarks

In versions of Visio products earlier than Visio version 2000, this section was named Image Info.

Geometry section

See also

Contains rows that list 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.

The Geometry section contains the following row types. For details, see the row topics.

Row	Description
<u>MoveTo</u>	Move to a coordinate.
<u>LineTo</u>	Draw a line to a coordinate.
<u>ArcTo</u>	Draw a circular arc to a coordinate.
EllipticalArcTo	Draw an elliptical arc to a coordinate.
<u>PolylineTo</u>	Draw a polyline, or consecutive lines, to a coordinate.
<u>NURBSTo</u>	Draw a nonuniform rational B-spline (NURBS) to a
	coordinate.
<u>SplineStart</u>	Start a spline.
<u>SplineKnot</u>	Draw a spline segment to a knot coordinate.
<u>InfiniteLine</u>	Draw an infinite line from one coordinate to another.
<u>Ellipse</u>	Draw an ellipse from a center coordinate and a

major/minor axis.

Remarks

To change a row type in this section, right-click the row, and then click **Change Row Type** on the shortcut menu.

Glue Info section

See also Rows

Contains formulas generated for a 1-D shape when the 1-D shape is glued to other shapes.

Group Properties section

See also Rows

Contains cells for a group that control how you add shapes to a group, move members of a group, and select groups. You can also set these values on the **Behavior** tab in the **Behavior** dialog box (click **Behavior** on the **Format** menu).

Hyperlinks section

See also

Contains cells for creating multiple jumps between a shape or drawing page and another drawing page, another file, or a Web site.

Remarks

To define a hyperlink, you can enter values in the cells of this section, or use the **Hyperlinks** dialog box (click **Hyperlinks** on the **Insert** menu).

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

Image Properties section

See also Rows

Contains the gamma, brightness, contrast, blur, sharpen, and denoise values for bitmaps (an image that consists of pixels) and appears in the ShapeSheet window only for such objects.

Layer Membership section

See also Rows Cells

Contains one row that lists each layer to which the shape is assigned.

Remarks

The layer assignment is shown as an index to the list of layers on the page. The layer index corresponds to the order of the layers in the **Layer Properties** dialog box (on the **View** menu, click **Layer Properties**). 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.

Layers section

See also Rows

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. Its cells correspond to the options in the **Layer Properties** dialog box (click **Layer Properties** on the **View** menu).

Line Format section

See also Rows

Contains cells that control line attributes for a shape, such as pattern, weight, and color. They determine whether the line ends are formatted (for example, with an arrowhead), the size of line end formats, radius of the rounding circle applied to the line, and line cap style (round or square).

Remarks

You can set line formats by using the **Line** dialog box, by applying a line style, or by entering a formula in a Line Format cell.

Miscellaneous section

Rows

Contains various attributes of shapes and groups, such as control selection highlighting and visibility.

Remarks

You can set the value of many of the cells in this section in the ShapeSheet window, or on the **Behavior** tab in the **Behavior** dialog box in the drawing window (click **Behavior** on the **Format** menu, and then click settings on the **Behavior** tab). You can set other group properties using cells in the Group Properties section.

In Microsoft Visio version 5.0 products, this section included ObjInteract and ObjBehavior cells. The behavior in the ObjInteract cell is now mapped to the following cells in the Shape Layout section: ShapeFixedCode, ShapePermeableX, ShapePermeableY, and ShapePermeablePlace. The behavior in the ObjBehavior cell is now mapped to the ConFixedCode and ShapeRouteStyle cells.

In Visio version 5.0 Japanese products, this section included a VerticalText cell. In Visio version 2000 products and later, the value of this cell is stored in the TextDirection cell in the Text Block Format section.

Page Layout section

See also Rows

Contains cells that control the page settings for shapes and connectors, such as the spacing between all shapes on the page, spacing between all connectors on the page, and routing style for all connectors on the page.

Remarks

To override the page defaults for a specific shape, use the cells in the Shape Layout section.

You can also set the values of the cells in this section on the **Layout and Routing** tab in the **Page Setup** dialog box (click **Page Setup** on the **File** menu).

Page Properties section

See also Rows

Contains cells that control page attributes, such as the page width, height, and scale.

Remarks

You can set the values of page properties by making an entry in the Page Properties section cell, or using the **Page Setup** dialog box in the drawing window (click **Page Setup** on the **File** menu).

Paragraph section

See also Rows

Shows the paragraph formatting attributes for the shape's text, such as indents, line spacing, bullets, and horizontal alignment of paragraphs.

Remarks

Paragraph formats can be set by making an entry in a Paragraph section cell by applying a text style, or by entering values in the **Text** dialog box (click **Text** on the **Format** menu).

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

Protection section

Rows

Shows the current lock settings set with the **Protection** command plus a few additional locks that can be set only in a ShapeSheet window.

Remarks

Locking helps prevent inadvertent changes to the shape but does not prevent Microsoft 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, use the GUARD function.

Ruler & Grid section

See also Rows

Shows the current settings of the page's rulers and grid.

Remarks

This section is a page property that can be viewed only in the ShapeSheet window for a page. The cells it contains correspond to the settings in the **Ruler & Grid** dialog box (click **Ruler & Grid** on the **Tools** menu).

Scratch section

Rows Cells

A work area for entering and testing formulas that can be referred to by other cells.

Remarks

You can add this section using the **Insert Section** dialog box. Make sure the ShapeSheet window is the active window, then on the **Insert** menu, click **Section**.

The Scratch section is typically used to isolate repeated complex calculations. If your solution has a well-defined purpose, it's wiser to use a cell in the User-Defined Cells section for clarity because User cells can be named.

Cells in the Scratch section use units in two different ways. X and Y cells use drawing units; A through D cells don't use units. (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 Microsoft 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.

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

Cell	If you enter	Visio treats it as	Result
X	PNT(3,4)	PNTX(PNT(3,4))	3.0000 in.
Y	PNT(3,4)	PNTY(PNT(3,4))	4.0000 in.
A-D	PNT(3,4)	PNT(3,4)	PNT(3.0000 in., 4.0000
			in.)

Shape Layout section

See also Rows

Contains cells that control shape placement and connector routing settings.

Remarks

You can also set the values of the shape-specific cells in this section using the **Placement** tab in the **Behavior** dialog box (click **Behavior** on the **Format** menu). You can set the values of the connector-specific cells in this section using the **Connector** tab in the **Behavior** dialog box. (To see the **Connector** tab, select the connector in the drawing window, and then click **Behavior** on the **Format** menu.)

You can use values in the Shape Layout section to override default values contained in the Page Layout section.

Shape Transform section

See also Rows

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.

Style Properties section

See also Rows

Includes cells that control style behavior, such as whether a style includes text, line, and fill attributes.

Remarks

You can also set these attributes using the **Define Styles** dialog box (click **Define Styles** on the **Format** menu).

Tabs section

See also Rows

Contains cells for shapes or styles that control tab stop position and alignment.

Remarks

You can also set these values using the **Tabs** tab in the **Text** dialog box (click **Text** on the **Format** menu) or using the **Text Ruler**. To show the **Text Ruler**, right-click the text while in text-editing mode, and then click **Text Ruler** on the shortcut menu.

You can set tab stops for specific characters by selecting the text, and then setting the tab stop. When you set tab stops for specific characters, a new row is added in the ShapeSheet window. The name of the row represents the number of characters the tab stop contains.

Text Block Format section

See also Rows

Contains cells that control the alignment, margins, and default tab stop position of text in a shape's text block.

Remarks

You can also set these text attributes using the **Text** dialog box (click **Text** on the **Format** menu).

Text Fields section

See also Rows

Displays functions and formulas inserted in the shape's text using the **Field** dialog box.

Remarks

The only way to add this section is to insert a field into a shape using the **Field** dialog box (click **Field** on the **Insert** menu).

Each text field in a text block has a corresponding row in the ShapeSheet window. The name of the row represents the number of characters the field contains.

Text Transform section

See also Rows

Contains positioning information about a shape's text block.

Remarks

To add this section, use the **Insert Section** dialog box (with the ShapeSheet window active, click **Section** on the **Insert** menu). Setting values in the Text Transform section is comparable to working with the text block using the **Text Block** tool.

User-Defined Cells section

See also

A work area for entering formulas in user-specific cells that can be referred to by other cells and add-on tools.

Remarks

Like the Scratch section, the User-Defined Cells section can be used by your solution in many different ways. The advantage of the User-Defined Cells section is that cells can be named—cells in the Scratch section cannot be named.

If your solution has a well-defined purpose it is suggested that you use cells in the User-Defined Cells section for clarity because User cells can be named. The Scratch section is typically used to isolate repeated complex calculations.

To add this section, use the **Insert Section** dialog box (with the ShapeSheet window active, click **Section** on the **Insert** menu).