GS-Calc 6.0 - Help Contents

Using GS-Calc 6.0

Using COM Automation

Automation methods and properties

As a COM automation server, GS-Calc enables you to create, open and edit its documents directly from your own apps. Creating the default automation object means opening a single GS-Calc document. You can use the following methods and properties (*):

Construction
ProgID

Initialization, Opening, Saving and Termination
Load
LoadDatabase
LoadTextFile
LoadExcelFile
SetDocumentState / GetDocumentState
SetPassword
Save
SaveAs
SaveExcelFile
SetModifiedRange
SetUserControl / GetUserControl
Show / Hide
ResetDocument
QuitApplication

Editing and Formatting
AddDropDownList / RemoveDropDownList
Delete
DeleteAll
Evaluate
GetType
GetLastColumn
GetLastRow
InsertColumnAt
InsertRowAt
IsCellEmpty
IsColumnEmpty
IsRowEmpty
PutNumber / GetNumber
PutText / GetText
PutFormula / GetFormula
Recalculate
RecalculateAll
RemoveColumns
RemoveRows
SetColumnWidth / GetColumnWidth
SetCurrentColumn / GetCurrentColumn
SetCurrentRow / GetCurrentRow
SetFirstColumn / GetFirstColumn
SetFirstRow / GetFirstRow
SetFormat / GetFormat / RemoveFormat
SetRowHeight / GetRowHeight
SetZoomFactor / GetZoomFactor
Sort

Window Size and Position
MaximizeAppWindow
MaximizeDocWindow
MinimizeAppWindow
MinimizeDocWindow
RestoreAppWindow
RestoreDocWindow
SetTreePaneWidth / GetTreePaneWidth
TileDocWindows

Printing and Page Setup
Print
PrintAll
PrintCellRange
PrintPageRange
SetPaGeHeaders / GetPageHeaders
SetPageOptions / GetPageOptions
SetPageMArgins / GetPageMargins
ShowGridLines / HideGridLines

Manipulating Worksheets
CopyWorksheet
DeleteItem
ExpandFolder / CollapseFolder
GetCurrentItem
GetFirstItem
GetItemID
GetItemPath
GetNextItem
IsFolder
InsertFolder
InsertWorksheet
IsSelected
IsWorksheet
SelectItem
SetItemName / GetItemName

(*) All declarations correspond to helper functions generated by VC++ (based on the type library).

Sample code

Editing documents

Press ENTER or any letter/digit to edit the contents of the current cell. To accept new data press ENTER once again or use one of the following cursor keys: UP ARROW, DOWN ARROW, PAGE UP (+CTRL), PAGE DOWN (+CTRL). Pressing ESC will return the original value. You can enter up to 1024 characters. To add an address of some cell to the currently edited cell, press SHIFT and click that cell with the left mouse button.

If you want to view and scroll different parts of your document independently, use the **Split** command from the **Window** menu to split the worksheet window.

To scroll the document contents, you can use the following keys: UP ARROW, DOWN ARROW, PAGE UP (+CTRL), PAGE DOWN (+CTRL), LEFT ARROW (+CTRL), RIGHT ARROW (+CTRL), HOME, END, CTRL+HOME (scrolls to the intersection of the first column and first row), CTRL+END (scrolls to the intersection of the last column and the last row).

You can use a wheel-mouse to perform (auto-) scrolling or zooming. In Windows 2000 or ME pressing the x-buttons switches among available worksheets forward and backward.

You can select cells using mouse or the SHIFT key. If you want to select a group of objects (OLE objects, charts, text frames), you must press and hold down the SHIFT key and click the desirable objects.

GS-Calc enables you to store unlimited numbers of worksheets in one document. You can organize and manage them in a variety of ways to make it easy to locate the worksheet you want. After activating the worksheets' tree pane you can use standard cursor keys to browse available worksheets. The LEFT ARROW and RIGHT ARROW keys are used to collapse and expand the selected folder. To edit the name of the selected item press ENTER or double-click it. Pressing any letters causes searching for an item whose name begins with these letters. Selected worksheets, highlighted by a small blue arrow, are automatically opened and displayed in the worksheet pane.

Notes

- The "first/last row" and "first/last column" terms mean the first and last row and column containing a not empty cell.
- The folder containing the default worksheet is always expanded after loading the given document
- You can change the appearance of the worksheet tree expanding or collapsing its folders, scrolling the current selection and changing width of the tree pane. However, such changes are not treated as operations that modify the document contents. You have to use the **Save** command explicitly before closing the document.
- Some default document settings that can be modified in Windows registry database include: the default display scale, default print scale and default font. GS-Calc uses the following key: **HKEY_CURRENT_USER\Software\JPS Development\GS-Calc\Settings**

Related Topics

Entering text, numbers, dates and formulas
Adding, deleting, renaming and moving folders and worksheets
Functions
Operators

Entering text, numbers, dates and formulas

GS-Calc automatically recognizes the type of the entered data. The cell type (**Text**, **Number** or **Formula)** is displayed in the first row of the document window. By default, text cells are left-aligned. If you want a given formula or number to be treated as text, you can enter the apostrophe (') as the first character (e.g. '3/4+1). Numbers with more than 15 digits are displayed in exponential format unless you use a user-defined style. You may use both the point (.) and the system-defined decimal separator when entering decimal places. If a number can't be fully displayed in a cell, the "#" characters are displayed.

The currency format used by GS-Calc is determined by Regional Settings in Control Panel.

Date/time values can be entered as numbers or as formatted date/time strings. The (positive or negative) date/time serial number consists of an integer portion representing the number of days since 12.30.1899, midnight and a fractional portion representing hours (1 day -> 1, 1 hour -> 1/24, 1 minute -> 1/24/60, etc). For example: "1/01/2099" = 72686, "1/1/1680" = -80531, "6:00 PM" = 0.75.
GS-Calc uses the short date format defined in Windows Regional Settings. For example, if you define that format as "M/d/yyyy" and "h:mm:ss tt", the following expressions will be valid: "1/01/2099", "1/1/1999", "11:00 AM 1/1/1680", "1/01/99 11:00 AM", "11:00:45 PM". For "MMM-d-yyyy" this can be: "Jan-1-2099", "May-30-1999", etc. The valid date range is 1 January 100 to 31 December 9999.

Optionally, formulas can begin with = or + (but it is not necessary). To display and print formulas instead of their values, clear the "**Display values of formulas**" check box in the **Options** dialog box.
If you enter one or more formulas containing recurrent expressions (for example, B1 into the A1 cell and A1 into B1), the program displays a list of cells causing such an iterative recalculation as a status bar message. The recurrence level can be changed in the **Options** dialog box.

When entering formulas you can use absolute or relative cell addressing. If you copy formulas from one cell to another, GS-Calc modifies relative cell addresses. For example, "A1" copied from B1 to B2 will appear as "A2" in B2. To avoid this, enter a dollar sign ($) before columns (to preserve the column number) or rows (to suppress the row change), e.g. A1, $A1, A$1. Clearing the "**Modify cell addresses when copying formulas**" check box in the **Options** dialog box has the same effect.

You can use the "Sum" and "Product" toolbar commands to quickly calculate one of these values for a given cell range. The following rules are used: (1) if the last row of the selected range is empty or if the last column is not empty, the respective values are inserted in the last row; (2) if the last column is empty, the respective values are inserted in the last column.

Examples

Numbers	**Text**	**Formulas**
5	abc	A1+A2
5.56	GSCALC	A1:B5
12.45e+12	Table	$A1+A$2+A3
-1995	'123	A1/A2+FACT(A3)
23%	'A1/45	"ab" &
	"123"	IF(A1="cd";A2;A3)
		FVAL(1000;3%;A1)
		FVAL(1000;0,03;A1)

Related Topics

Using definable number styles
Functions
Operators

Using drag-drop functions

Use the Drag-Drop functions to copy cells and objects and copy or move entire worksheets or folders containing worksheets.

Copying cells and objects

To copy data using drag-drop functions:
1. Select a cell, range or object(s) you want to copy.
2. Press and hold down the CTRL key.
3. Drag the selection to the desirable location.

You can copy data within the same document, different GS-Calc documents or different applications. By default, GS-Calc creates an embedded object. If you want to paste a link into another application press and hold down the SHIFT key.

Related Topic

Using OLE and DDE

Using OLE and DDE

As an OLE client and server, GS-Calc supports editing objects in-place, linking objects and drag-drop functions. The size of the metafile (picture) representation of an inserted GS-Calc object is always limited to (approx.) 2000x2000 pixels.
If a GS-Calc document is activated in-place, the tree pane is not displayed - you must open it in the full-window mode to browse all worksheets. Changing the size of an in-place-edited object increases or decreases the number of visible cells. To change the displayed cell range for a linked GS-Calc object (in an OLE client application), modify the column and row numbers included in its name.

By default, if you copy a chart to another program, GS-Calc creates an OLE object which consists of the smallest range of cells covering both the chart and the cells with its data. If that chart is displayed in the full-window mode, the created picture will contain only that chart.

Double-click an OLE object to execute the first command from the **Edit/Object** menu. If you press and hold down the SHIFT key, the subsequent command (if any) from the same menu will be performed. If you use commands from the **Object** menu instead of double-clicking, you will save some memory if the current undo level is greater than 0 (the copy of the original object is not created), but you will not be able to undo any changes of this object.

Note: GS-Calc doesn't support inserting GS-Calc objects into its own documents.

You can use the DDE commands to enter data into the GS-Calc document. The DDE "execute" command should have the following form:
[insert(file_name,column,row,value]
Each parameter should be enclosed in quotation marks and no spaces are allowed between parameters. For example:
[insert("d:\tmp\analyze.gsc","2","11","4.5")]
The *value* parameter can represent a number or a text string. After accepting a new value GS-Calc performs recalculation and updates charts and associated OLE server items.

To open or print a document, you can also use the following commands
[open(file_name)]
[print(file_name)]
[printto(file_name,printer_name,driver_name,port_name)]

During the DDE initiation use the "gscalc" and "system" text IDs to create application and item atoms.

Importing and exporting text, xBase, Excel and Access files

GS-Calc enables you to use several of the most popular database/spreadsheet formats. To save your data to a file in one of the supported formats, use the Save Copy As command and choose the desirable format from the **File Type** drop-down list in the displayed **Save Copy As** dialog box. To open a given file, choose the exact file type in the **Open** dialog box.

Text (*.txt, *.csv, *.tab, ...)

You can define any character as the cell separator in a text file. Saved cell values containing that separator or double quotation marks (") will be enclosed in double quotation marks (the inner quotation marks are doubled). You can specify whether the created text file should contain formulas or their values, plain numbers or formatted numbers and whether the entire current folder should be saved. If you save multiple worksheet, they will be separated by some special (longer) separator so that they can be re-loaded later correctly. When opening text files you can specify whether all cells should be treated as text labels or whether GS-Calc should parse numbers and/or formulas. Additionally, GS-Calc can split the data into multiple worksheet if the number of columns exceeds a certain value (by default, it's 512).

dBase III+, dBase IV, FoxPro 2.x, Clipper (*.dbf, *dbt, *.fpt) (*)

Before creating a new xBase database you can define the xBase record structure specifying field types (Character, Numeric, Float, Logical, Date, Memo), lengths and numbers of decimal places. When opening an xBase file you can specify which fields and which records should be displayed in GS-Calc. Database records marked as "deleted" are displayed as highlighted worksheet rows. If it's necessary, data can be converted from or to the OEM character set. GS-Calc performs text length validation as database text fields are modified.
Existing xBase files are opened and edited "in-place", without changing the database structure. The **Save** command saves the current selection only. If you don't select any cells, only one field of a single record pointed by the current cell will be updated. New records are always added at the end of the edited database. To remove records from the edited record set, use the Remove Rows command from the **Tools** menu.

Excel 97, Excel 2000 (*.xls) (*)

Before opening an Excel workbook you can specify which worksheets should be imported. GS-Calc converts all necessary formatting information including fonts, cell formats, background colors, number styles, column widths, row heights and page settings. The current version of GS-Calc does not convert formulas - instead, their current values are imported and exported.
By default, all text strings are exported as compressed Unicode strings. If you're using a non-English versions of Windows and/or entering some non-ascii characters and if they aren't later correctly displayed by Excel, you must change the value of the **SaveUnicodeStrings** entry in the following key in the Windows registry database:
HKEY_CURRENT_USER\Software\JPS Development\GS-Calc\Settings
If it's 1, strings containing non-ascii characters will be saved as uncompressed Unicode strings (all other strings will still be saved as "compressed").

Access 95-2000 (*.mdb) (*)

You can define the record structure specifying field types and lengths. Access databases are opened in the read/write mode - to open a database placed on a CD or any other read-only media, you must first copy it on your hard disk and remove the read-only file attribute. When opening an Access file, you can specify which fields and which records should be displayed. GS-Calc performs text length validation as text fields are modified.
Existing Access files are opened and edited "in-place", without changing the database structure. The **Save** command saves the current selection only. If you don't select any cells, only one field of a single record pointed by the current cell will be updated. To remove records from the edited record set, use the Remove Rows command from the **Tools** menu.

(*) All brands and names are the property of their respective owners.

Creating and editing charts

Use the Insert Chart command to insert a new chart. The **Chart Properties** dialog box enables you to choose the chart type, data ranges and other options affecting its appearance:
Type
Data
Axes
Grid
Description
Display
3D View

Charts are updated after every action that can change the document contents. To edit an existing chart, double-click it or use the **Properties** command from the **Edit** menu. To change the size of a chart, click it and drag one of the eight points displayed around it.

Related Topic

Using OLE and DDE

Changing the default number format

GS-Calc uses the decimal separator and the group separator defined in Regional Settings. You can change them using Control Panel.

Changing the default currency, date and time formats

GS-Calc uses the formats defined in Regional Settings. You may change them using Control Panel.

Changing the default font type

GS-Calc saves some parameters in the registry database. It uses the following key name:

HKEY_CURRENT_USER\Software\JPS Development\GS-Calc\Settings

Change value of the **DefaultFont** entry in this key to change the default font used by GS-Calc. The default name is "Arial".

Printing worksheets

When you print a single worksheet after choosing the **Print** command from the **File** menu, you can print the entire worksheet, specified pages or the current selection: a range, OLE objects, charts or text frames.
If you use the **Print All** command, GS-Calc prints all pages of all worksheets in the current folder and its subfolders using their individual page/printout settings.
Using the **Page Setup** command you can change page margins, specify page layout, headers and footers etc. Page settings are defined separately for each worksheet. To duplicate them within the current folder and its subfolders, use the **Duplicate Data** command from the **Tools** menu.

Related Topic

Defining headers and footers

Searching for worksheets or cells

1. To search for worksheets, activate the tree pane, use the **Find Worksheet/Cell** command from the **Edit** menu and enter a search filter in the **Find** dialog box.

(a) If you clear the **Use simple text search** checkbox:

> Filters can be any regular spreadsheet expressions returning numbers. If, for a given worksheet, the expression returns a value other than zero and "error", the worksheet will be added to the found list.

(b) If you check the **Use simple text search** checkbox:

> Filters can be any text strings. They are compared against the (unformatted) contents of all cells in a worksheet. The specified text can be a substring of the cell contents. The comparison is not case-sensitive.

You can sort the list of found worksheets by name, size and the modification date.

2. To search for cells containing a given substring, activate the worksheet pane and use the **Find Worksheet/Cell** command from the **Edit** menu. If you select any cells before using this command, the searching will be limited to those cells.

Related Topic

Search expressions

Changing column widths and row heights

To change column width, drag the boundary on the right side of the column heading until the column is the width you want.
To change row height, drag the bottom boundary of the row heading until the row is the height you want.

If you want to specify the exact width or height in pixels or if you want to change the width or height for multiple rows or columns, use the **Change Width/Height** command from the **Tools** menu.

Adding, deleting, renaming and moving folders and worksheets

1. To add a new worksheet or folder use the **New Folder** or **New Worksheet** command from the **Insert** menu. After inserting a new tree item press ENTER to accept its default/suggested name or enter your own text.

2. To copy an existing folder or worksheet(s), do one of the following:
(a) Point to a folder or worksheet, press and hold down the left mouse button and the CTRL key, and drag that item to a new location. The highlighted items show the current target location. You can copy single worksheets, empty folders or folders containing other worksheets or nested folders.
(b) Point to a folder or worksheet and use the **Copy** command from the **Edit** menu. Point to the target location and use the **Paste** command. You can copy single worksheets, empty folders or folders containing other worksheets or nested folders.

3. To move a folder or worksheet to the desirable location, repeat the 2 (a), except for pressing the CTRL key. If the "target" location is placed within the moved folder, GS-Calc performs the "copy" operation.

4. To rename a worksheet or folder, select it and press the ENTER key or double-click it.

Using definable number styles

Apart from some common styles like currency or percent, you can define your own format. To specify a new style, select the **New** item from the **Style** list in the **Cell Properties/Format** dialog box. In the **New Style** dialog box enter your format definition for positive (incl. zero) and negative numbers. The format specification must has the following form:

[text] % [flags] [width] [.precision] type [text]

Examples

Format	**Number**	**How it's displayed**
%f	123	123.000000
%06.0f	123	000123
%09.2f	123	000123.00
$ %.2f	123.357	$ 123.36
%.f	1234567891011121314	1234567891011121400
%e	123	1.230000e+002
%.3e	123	1.230e+002
%3.e	123	1e+002
%+g	123	+123
(%g)	-123	(-123)
Negative	-123	Negative
abc	123	abc

The optional fields, control the formatting, as follows:

text

This can be any text - not containing the "%" characters - that will be displayed along with (or instead of) numbers.

type

e Signed value having the form [–]*d***.***dddd* **e** [*sign*]*ddd* where *d* is a single decimal digit, *dddd* is one or more decimal digits, *ddd* is exactly three decimal digits, and *sign* is + or –.

E Identical to the **e** format except that **E** rather than **e** introduces the exponent.

f Signed value having the form [–]*dddd***.***dddd*, where *dddd* is one or more decimal digits. The number of digits before the decimal point depends on the magnitude of the number, and the number of digits after the decimal point depends on the requested precision.

g Signed value printed in **f** or **e** format, whichever is more compact for the given value and precision. The **e** format is used only when the exponent of the value is less than –4 or greater than or equal to the precision argument. Trailing zeros are truncated, and the decimal point appears only if one or more digits follow it.

G Like the **g** format, except that **E**, rather than **e**, introduces the exponent (where appropriate).

flags

– Left align the result within the given field width.

+	Prefix the output value with a sign (+ or –) if the output value is of a signed type
0	If *width* is prefixed with **0**, zeros are added until the minimum width is reached. If 0 and – appear, the **0** is ignored.
blank (' ')	Prefix the output value with a blank if the output value is signed and positive; the blank is ignored if both the blank and + flags appear.
#	When used with the **e**, **E**, or **f** format, the **#** flag forces the output value to contain a decimal point in all cases. When used with the **g** or **G** format, the **#** flag forces the output value to contain a decimal point in all cases and prevents the truncation of trailing zeros.

width

The *width* argument is a nonnegative decimal integer controlling the minimum number of characters printed. If the number of characters in the output value is less than the specified width, blanks are added to the left or the right of the values, depending on whether the – flag is specified. If *width* is prefixed with 0, zeros are added until the minimum width is reached. The width specification never causes a value to be truncated.

precision

It specifies a nonnegative decimal integer, preceded by a period (**.**) which specifies the number of decimal places. The precision specification can cause rounding of a value.
If ***Type*** is **e** or **E**, the last printed digit is rounded. Default precision is 6. If ***precision*** is 0 or the period (.) appears without a number following it, no decimal point is printed.
If ***Type*** is **f**, If a decimal point appears, at least one digit appears before it. The value is rounded to the appropriate number of digits. Default precision is 6. If ***precision*** is 0, or if the period (.) appears without a number following it, no decimal point is printed.
If ***Type*** is **g** or **G**, the ***precision*** specifies the maximum number of significant digits printed. By default six significant digits are printed. Any trailing zeros are truncated.

Exchanging data between worksheets

In addition to copying and drag-drop functions, you can use the following functions to access cells from other worksheets:

CELL_EX
HLOOKUP_EX
FOLDER_COUNT
GETEXVAL
INDEX_EX
MATCH_EX
MAX_EX
MIN_EX
SHEET_COUNT
SUM_EX
VLOOKUP_EX

Related Topics

Copying cells and objects
Adding, deleting, renaming and moving folders and worksheets
Using Drag-Drop functions

Commands

File menu
Edit menu
View menu
Insert menu
Format menu
Tools menu
Window menu
Help menu

File menu

New
Open...
Save
Save As
Print
Print All
Print Preview
Page Setup
Password
Summary Info
Send
Exit

Edit menu

Undo
Redo
Repeat
Cut
Copy
Paste
Paste Special
Paste Link
Delete
Find Worksheet/Cell
Replace
Convert Formulas
Object Properties
Links

View menu

Status Bar
Standard Toolbar
Format Toolbar
Font Toolbar
Zoom
Grid
Full Screen
Synchronize Views

Insert Menu

Object
Chart
Text Frame
Data Series
Date/Time
Formula
New Folder
New Worksheet

Format menu

Cell Properties
Bold
Italic
Underline
Normal
Font

Tools menu

Sort
Analyze menu
Insert Column
Remove Columns
Insert Row
Remove Rows
Duplicate
Move to Foreground
Move to Background
Recalculate
Recalculate All
Options

Analyze menu

Equation System
Linear Optimization
Inverse Matrix
Determinant
Integration

Window menu

Split
Cascade
Tile Horizontally
Tile Vertically
Arrange Icons

Help menu

Help Topics
About GS-Calc

New command (File menu)

Use this command to create a new document in GS-Calc.

Open command (File menu)

Use this command to open an existing document in a new window. You can open multiple documents at once. Use the Window menu to switch among the multiple open documents.

Notes

You must select a proper file type from the **File Types** list in the **Open** dialog box.

Close command (File menu)

Use this command to close all windows containing the active document. GS-Calc suggests that you save changes to your document before you close it. If you close a document without saving, you lose all changes made since the last time you saved it.

Save command (File menu)

Use this command to save the active document to its current name and directory. If you want to change the name and directory of an existing document before you save it, choose the **Save As** command.
If you are editing a database, this command name is replaced by "Update Records".

Save As command (File menu)

Use this command to save and name the active document.

Page Setup command (File menu)

Use this command to change page layout.

Summary Info command (File menu)

Use this command to display a dialog box containing information about the current document.

Summary Info dialog box

GS-Calc displays information on the current document: the number of cells with particular contents, size of the document file and the date of last saving.

Summary Info dialog box

GS-Calc displays information on the current document: the number of cells with particular contents, size of the document file and the date of last saving.

Password command (File menu)

Use this command to change the current password.

1,2,3... command (File menu)

Use the numbers and filenames listed at the bottom of the File menu to open the last four documents you closed. Choose the number that corresponds with the document you want to open.

Send command (File menu)

Use this command to send the active document through electronic mail. This command presents a mail window with the active document attached to it. You may then fill out the To: field, Subject: field, etc., and add text to the body of the message if you wish. When you are finished you may click the "Send" button to send the message.

Exit command (File menu)

Use this command to end your GS-Calc session.

Undo command (Edit menu)

Use this command to reverse the last editing action. The name of the command changes, depending on what the last action was.

Redo command (Edit menu)

Use this command to redo the reversed editing action. The name of the command changes, depending on what the last action was.

Repeat command (Edit menu)

Use this command to repeat the last editing action. The name of the command changes, depending on what the last action was. This command is inactive for some types of actions.

Cut command (Edit menu)

Use this command to remove the currently selected data from the document and put it on the clipboard.

Cutting data to the clipboard replaces the contents previously stored there.

To cut:
(1) the cell contents - select cells and use this command; you can choose what should be cut:
format,
cells containing text,
cells containing numbers,
cells containing formulas,
all cells;

(2) object - select the object and use this command; if you want to select a group of objects hold down the SHIFT key.

Copy command (Edit menu)

Use this command to copy selected data onto the clipboard.

Copying data to the clipboard replaces the contents previously stored there.

Paste command (Edit menu)

Use this command to insert a copy of the clipboard contents at the insertion point. This command is unavailable if the clipboard is empty.

Delete command (Edit menu)

Use this command to remove the currently selected data from the document.

To delete:
(1) the cell contents - select cells and use this command; you can choose what should be cut:
format,
cells containing text,
cells containing numbers,
cells containing formulas,
all cells;

(2) object - select the object and use this command; if you want to select a group of objects hold down SHIFT key.

Formula command (Insert menu)

Use this command to paste a formula into the current cell. After choosing this command GS-Calc displays the **Paste Formula** dialog box which enables you to retrieve all available formulas (shortcut: right mouse button + SHIFT).

Find Worksheet/Cell command (Edit menu)

Use this command to search for worksheets or to search the active document for text. The search mode depends on whether the tree pane is active.

Related Topic

Searching for worksheets and cells

Find dialog box

Related Topic

Find command

Replace command (Edit menu)

Use this command to find and replace text in the active document. While comparing text GS-Calc uses unformatted cell contents.

If you select any cells before using this command, the searching will be limited to those cells.

Replace dialog box

Related Topic

Replace command

Convert Formulas command (Edit menu)

Use this command to convert the selected formulas to numbers or text strings.

Object Properties command (Edit menu)

Use this command to change properties of the selected OLE object. You can change its type, the way it is displayed, its scaling and link.

Standard Toolbar command (View menu)

Use this command to display and hide the **Standard Toolbar**, which includes buttons for some of the most common commands in GS-Calc, such as **File Open**. A check mark appears next to the menu item when the toolbar is displayed.

Standard Toolbar

The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in GS-Calc.

Format Toolbar command (View menu)

Use this command to display and hide the Format Toolbar, which includes buttons for some of the most common commands in GS-Calc. A check mark appears next to the menu item when the Toolbar is displayed.

Format Toolbar

The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in GS-Calc.

Font Toolbar (View menu)

Use this command to display and hide the Format Toolbar, which includes buttons for some of the most common commands in GS-Calc. A check mark appears next to the menu item when the Toolbar is displayed.

Status Bar command (View menu)

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed.

Status Bar

The status bar is displayed at the bottom of the GS-Calc window. To display or hide the status bar, use the **Status Bar** command in the **View** menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them. If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right areas of the status bar indicate:

Indicator	**Description**
AUTO	Formulas are recalculated automatically.
TRANS	Modifying formulas after changing its location.
CAP	Caps Lock is on.
NUM	Num Lock is on.
SCRL	Scroll Lock is on.

Zoom command (View menu)

Use this command to enlarge the view of the current document. Valid scaling factors are from 30 to 400%.

Grid command (View menu)

Use this command to hide or show the grid of the active document.

Full Screen command (View menu)

Use this command to turn on or off the full screen mode. When it's on, documents are displayed on the entire available screen area.

Synchronize Views (View menu)

Use this command to enable automatic scrolling of columns or rows in the second worksheet view. If the second view pane is minimized, GS-Calc changes its height to approx. a half of the height of the visible area.

Chart command (Insert menu)

Use this command to insert a new chart. The **Chart Properties** dialog box enables you to choose the chart type, data ranges and other options affecting its appearance:

Type
Data
Axes
Grid
Description
Display
3D View

Charts are updated after every action that can change the document contents. To edit an existing chart double-click it.

Related Topic

Creating and editing charts

Data Series (Insert menu)

Use this command to insert a series of numbers or a text into the selected cells.

Text Frame command (Insert menu)

Use this command to insert a new text frame into the active document.

Date/Time command (Insert menu)

Use this command to insert a date and/or time value into the selected cells.

New Folder command (Insert menu)

Use this command to insert a new folder into the worksheet tree.

Related Topic

Adding, deleting, renaming and moving folders and worksheets

New Worksheet (Insert menu)

Use this command to insert a new folder into the worksheet tree.

Related Topic

Adding, deleting, renaming and moving folders and worksheets

Cell Properties command (Format menu)

Use this command to change the properties of the selected cells. You may change the style, alignment, displaying options, colors, frame and shadows.

Bold command (Format menu)

Use this command to select bold format.

Italic command (Format menu)

Use this command to select italic format.

Underline command (Format menu)

Use this command to select underline format.

Strikeout command (Format menu)

Use this command to select strikeout format.

Normal command (Format menu)

Use this command to select normal format.

Font command (Format menu)

Use this command to display the **Font** dialog box which enables you to specify font name, size, style, color and effects.

Font dialog box

Related Topic

Font command

Sort command (Tools menu)

Use this command to sort the selected cells. The first column of the current range is treated as the sort key. If you don't select any cells, the program will sort the current column.

Equation System command (Tools menu)

After using this command GS-Calc displays the **Equation System** dialog box which enables you to solve linear equations.

Linear Optimization command (Tools menu)

After using this command GS-Calc displays the **Linear Optimization** dialog box which enables you to find optimal solutions using linear programming.

Inverse Matrix command (Tools menu)

After using this command GS-Calc displays the **Inverse Matrix** dialog box which enables you to inverse matrixes.

Determinant command (Tools menu)

After using this command GS-Calc displays the **Determinant** dialog box which enables you to calculate determinants of matrixes.

Integration command (Tools menu)

After using this command GS-Calc displays the **Integration** dialog box which enables you to perform integration of functions.

Analyze dialog box

Equation System
Linear Optimization
Inversion of Matrix/Determinant
Integration

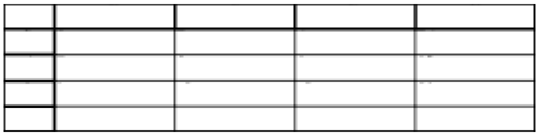
Equation System dialog box

To solve an equation system:

1. Select cells containing variables and click the **Variables** button.
2. Select cells containing constants and click the **Constants** button. The number of these cells must be equal to the number of rows selected in step **1**.
3. Select cells which will contain the result and click the **Result** button. The number of these cells must be equal to the number of columns selected in step **1**.
4. Click the **OK** button.

Example

the equation system
x + 2*y + 4*z = 24
2*x + z = 13
5x - y - z = 15
can be expressed in the following way:



Variables - B3:D5
Constants - E3:E5
Results - B6:D6

Linear Optimization dialog box

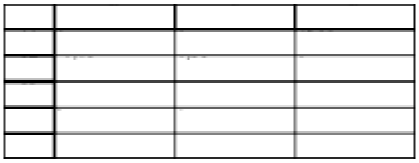
To find a solution using linear optimization:

1. Select cells containing variables and click the **Variables** button.
2. Select cells containing constants and click the **Constants** button. The number of these cells must be equal to the number of rows selected in step **1**.
3. Select cells containing optimized function factors and click the **Function** button. The number of these cells must be equal to the number of columns selected in step **1**.
4. Select cells which will contain found variables and function value and click the **Result** button. The number of these cells must be equal to the number of columns selected in step **1** plus one for the function result.
5. Click the **OK** button.
Note: The result may include a few symmetric solutions.

When finding the solution GS-Calc maximizes the given function.

Example

Problem
x <= 4500
-0.65x + 0.35y <= 0
5x + 8y -> max
can be expressed in the following way:



Variables - B11:C12
Constants - D11:D12
Function - B14:C14
Results - B15:D15

Inversion of Matrix dialog box

To find an inverted Matrix

1. Select cells containing the matrix and click the **Data** button.
2. Select cells which will contain the result. The number of these cells must be equal to the number of cells selected in step **1**.
3. Click the **OK** button.

To find a determinant

1. Select cells containing the matrix and click the **Data** button.
2. Select the cell which will contain the computed value and click the **Result** button.
3. Click the **OK** button.

Integration dialog box

To perform integration:

1. Select cells containing parameters and click the **Parameters** button. The range must contain the following values:

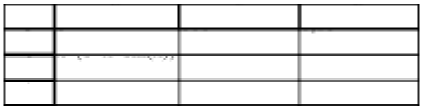
the top integration limit,
the bottom integration limit,
the precision as a percentage.

The integrated function can have no values in the limit points.

2. Select the cell containing the function and click the **Function** button. This function may consist of any formulas and numbers. The variable must be expressed as the "x" character.

4. Select the cell which will contain the result and click the **Result** button.

Example



Parameters - B5:D5 (integration from 1 to 100 at 0.01% relative precision)
Function - B6
Result - B7

Insert Column command (Tools menu)

Use this command to insert a new column. Inserting column may result in modifying formulas.

Example

If you insert the D column, the formula E1+E2 from the A1 cell will be replaced by F1+F2.

Remove Columns command (Tools menu)

Use this command to delete selected columns. Deleting columns may result in modifying formulas.
Example:
If you delete the D column, the formula E1+E2 from the A1 cell will be replaced by D1+D2.

Insert Row command (Tools menu)

Use this command to insert a new row. Inserting row may result in modifying formulas.

Example

If you insert the 10th row, the formula E100+E200 from the A1 cell will be replaced by E101+E201.

Remove Rows command (Tools menu)

Use this command to delete selected rows. Deleting rows may result in modifying formulas.

Example

If you delete the 10th row, the formula E100+E200 from the A1 cell will be replaced by E99+E199.

Change Width/Height (Tools menu)

Use this command to change the width or height of selected columns or rows.

Related Topic

Changing column widths and row heights

Duplicate command (Tools menu)

Use this command to copy data of the selected type from the current worksheet to all other worksheets in the current folder and its subfolders. This is an easy way to make the same: column widths, row heights, formats, selection, current view coordinates or page settings.

Move To Foreground command (Tools menu)

Use this command to move the selected object(s) before any other objects.

Move To Background (Tools menu)

Use this command to move the selected object(s) behind any other objects.

Recalculate command (Tools menu)

Use this command to recalculate all formulas within the active worksheets.

Recalculate All (Tools menu)

Use this command to recalculate all worksheets in the active document.

Options command (Tools menu)

Use this command to display the **Options** dialog box which enables you to change the program settings.

Cascade command (Window menu)

Use this command to arrange multiple opened windows in an overlapped fashion.

Tile Horizontal command (Window menu)

Use this command to vertically arrange multiple opened windows in a non-overlapped fashion.

Tile Vertical command (Window menu)

Use this command to arrange multiple opened windows side by side.

Split command (Window menu)

Use this command to split the active window into panes. You may then use the mouse or the keyboard arrows to move the splitter bars. When you are finished, press the mouse button or enter to leave the splitter bars in their new location. Pressing ESC keeps the splitter bars in their original location.

Arrange Icons command (Window menu)

Use this command to arrange the icons for minimized windows at the bottom of the main window. If there is an open document window at the bottom of the main window, then some or all of the icons may not be visible because they will be underneath this document window.

1,2,3... command (Window menu)

GS-Calc displays a list of currently open document windows at the bottom of the Window menu. A check mark appears in front of the document name of the active window. Choose a document from this list to make its window active.

Help Topics command (Help menu)

Use this command to display the opening screen of Help. From the opening screen, you can jump to step-by-step instructions for using GS-Calc and various types of reference information.

About GS-Calc command (Help menu)

Use this command to display the copyright notice and version number of your copy of GS-Calc.

Context Help command

Use the Context Help command to obtain help on some portion of GS-Calc. When you choose the Toolbar's Context Help button, the mouse pointer will change to an arrow and question mark. Then click somewhere in the GS-Calc window, such as another Toolbar button. The Help topic will be shown for the item you clicked.

Title Bar

The title bar is located along the top of a window. It contains the name of the application and document. To move the window, drag the title bar.

Scroll Bars

Displayed at the right and bottom edges of the document window. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the document. You can use the mouse to scroll to other parts of the document.

Format Toolbar

Use the **Format Toolbar** buttons to set or remove various formats for the selected cell(s). You can also do the same using the **Cell Properties** dialog box.

Editing documents

Editing documents

Editing documents

Editing documents

Editing documents

Editing documents

Editing documents

Editing documents

Editing documents

Installing GS-Calc

System requirements: Windows 9x/ME/2000

GS-Calc installs the following files:
gscalc.exe
gscalc.hlp
gscalc.cnt
gscalc.tlb
sample.gsc
readme.txt
order.htm (in the eveluation version only)
uninst_c.exe

Note:
To use MS Access 2000 (*) format in Windows 95/98, you may need to install "Microsoft Data Access Components". If MDAC hasn't been already installed in your system by another product, you can download the **mdac_type.exe** file from http://www.microsoft.com/data .

(*) All brands and names are the property of their respective owners.

Customer support

If have any problems with this product or if you have any suggestions, please contact:

E-mail: support@jps-development.com
Web: http://www.jps-development.com

JPS Development
ul. Chopina 13
58-100 Swidnica, Poland

tel.: +48-74-853-75-81
fax: +48-74-853-75-81

No information

No information on this topic.

No information

No information on this dialog box.

X-Y variant
Specifies whether an X-Y variant of the selected chart should be used. X-Y charts can present numeric relations between values on the X- and the Y-axis. If you check this option, categories describing the Y-axis will be treated as values of arguments for the given function values (data series). The x-values don't have to be sorted and they all must represent unique numbers.
Note: Bar and pie charts can't be used as X-Y charts.

Keep the X-Y aspect ratio
Specifies whether the aspect ratio between the X and the Y axes (that use the same units) should be preserved. For example, if the X-axis contains values between 0 and 50, and the Y-axis between 0 and 100, the displayed X-axis will be two times shorter than the Y-axis.
This option is ignored if the "Data \ Use independent data series ranges..." option is check.

Horizontal chart / Vertical chart
Specifies the chart orientation. This option doesn't apply to pie charts.

1., 2., 3. Data area
You can specify up to three different data areas (cell ranges) containing data series in rows, columns or single series. Initially, only the 1. range is used and its size depends on the cell range you select before inserting a new chart: (**1**) if the range contains more columns than rows, the "series in rows" option is used by default and the program assumes categories are in the first row, legend labels in the first column (the top-left cell is discarded) or (**2**) if the range contains more rows than columns, the "series in columns" option is used by default and the program assumes categories are in the first column, legend labels in the first row (the top-left cell is discarded).
To select a new range, select the desirable cells within the current document and click the given button.
Note:
You can use up to 16 series at the same time. Pie charts use only the first of the specified series.
GS-Calc uses formats of the respective document cells when displaying the x- and y-values describing charts. For example:
- if the x-values should be displayed as dates, the first cell of the range containing category values should have the date format set,
- if the y-labels of the 2nd series should be displayed as currency, the first cell of the range containing 2nd series should have the currency format set.

Use independent Y-axis ranges for each series
Specifies whether all defined data series should use independent / different Y-axis ranges. If you check this option, each thick on the Y-axis will be described by a number of labels equal to the number of series. The ranges can be calculated automatically for each series or can have predefined values (see: **Chart Properties / Axes**)

Categories / X-values
To select a new range containing category labels or x-values, select the desirable cells within the current document and click this button.

Legend
To select a new range containing legend labels, select the desirable cells within the current document and click this button.

Range of values on the X-axis
Specifies how the range of values on the X-axis should be calculated. You can use one of the following three options:
Default (from zero or the smallest negative x-value to the biggest x-value number),
Automatic (the smallest range of values containing all x-values),
From To (the start and end x-value defined by the user).
The *Automatic* and *From To* options apply to X-Y charts only. In the **Data Range** dialog box you can enter both numbers and formulas.

Range of values on the Y-axis
Specifies how the range of values on the Y-axis should be calculated. You can use one of the following three options:
Default (from zero or the smallest value in a given series to the biggest value in that series),
Automatic (the smallest range of values containing all data series values),
From To (the start and end value defined by the user).
In the **Data Range** dialog box you can enter both numbers and formulas.

X-axis scale / Y-axis scale
Specifies the X-axis and Y-axis scale. Non-linear x-scales apply to X-Y charts only.

Ticks on the X-axis / Ticks on the Y-axis
Specifies the number of ticks on the X-axis and Y-axis. The number of x-ticks can be defined for X-Y charts only.

X-labels, Y-labels, Z-labels
Specifies whether axis labels should be displayed (and how).

X-label font, Y-label font, Z-label font
Specifies fonts used to display x-, y- and z-labels.

Major gridlines
Specifies whether major gridlines should be displayed.

Color / Type
Specifies the color and type of major gridlines.

Minor gridlines
Specifies whether minor gridlines should be displayed.

Color / Type
Specifies the color and type of minor gridlines.

Number of minor x-lines
Specifies the number of minor vertical gridlines between each two major lines.

Number of minor y-lines
Specifies the number of minor horizontal gridlines between each two major lines.

X-Y, Y-Z, X-Z chart planes
Specifies which chart planes should be displayed and their colors.
Note: 2D charts use only X-Y planes and no planes are displayed on pie charts.

Title / Subtitle
Specifies the chart title and subtitle, their alignment and fonts.

X-axis title / Font
Specifies the X-axis title and its font.

Y-axis title / Font
Specifies the Y-axis title and its font.

Z-axis title / Font
Specifies the Z-axis title and its font.

Legend / Font
Specifies the chart legend and its font.

Fixed chart size
Specifies whether a fixed chart size should be used.

Full-window display
Specifies whether the selected chart should occupy the entire upper view pane (if its displayed in default document windows), the entire printed page (if its printed) or the entire OLE object window (if the document is activated in-place as an OLE object). A chart displayed in the full upper pane is not displayed/printed in the second view pane.

Display mode
Specifies the current display / print mode. When the "wire" mode is selected, the major gridline color is used to draw chart edges.

Bar width (2D/3D), Band width, Line width (2D), Pie thickness, Pie split, Line thickness (3D)
Specifies the size of various chart elements. The *pie thickness* and *split* parameters are relative to the current pie radius.

Background
Specifies the chart background color.

Data series colors
Specifies colors used for the subsequent data series.

Rotate X-Axis, Y-Axis, Z-Axis
Click these three button pairs to rotate the desirable (local) axis.

Reset
Use this button to displayed an "untransformed" chart.

Step
Specifies the type and size of the rotation step.

Perspective
Specifies whether perspective should be used.

Horz. lamp position, Vert. lamp position
Specifies the horizontal and vertical position of the light source. This option applies to 3D charts only.

Specifies the name of the opened table and its fields. A check mark in front of a field name indicates that the field will be included in the loaded data.

Fields of the selected table.

Specifies the field name.

Specifies the field type. If you change the initial setting, GS-Calc will perform necessary conversion.

Specifies the length of text fields and (for dBase-compatible formats) numeric fields.

Specifies the number of decimal places for numeric fields in a dBase-compatible file.

Specifies the list of the currently defined fields. To change a field name or length select that field and edit its description.

Specifies whether the program should convert text from the default Windows to OEM character set.

Specifies the list of imported fields.

Specifies the range of records to import.

Specifies whether the program should convert text from the OEM to Windows character set.

Specifies columns or rows that should be deleted.

Ascending sorting of selected data.

Descending sorting of selected data.

Specifies whether the reverse order of words should be used when comparing text strings. For example, comparing two names:

Tom A. Jones
Mark Smith

GS-Calc will use the following keys:

Jones A. Tom
Smith Mark

Specifies the sort keys. You can define more than one key to perform two- or three-level sorting.

Use this option to insert number sequences. You can specify the first number and how subsequent numbers should be calculated. The size of the series depends on the current selection.

Use this option to insert random number series. You can specify the distribution type for the generated series and additional parameters specific for each distribution. The size of the series depends on the current selection.
Note: To obtain random numbers spaced evenly within a given range, use the default **Uniform** type and specify that range in the **Options** dialog box.

Use this option to insert series of text strings, numbers or formulas. The size of the series depends on the current selection.

Specifies the number of empty cells between each two inserted elements of the data series.

Specifies the horizontal alignment for the selected cells.

Specifies the vertical alignment for the selected cells.

Specifies the style for the selected cells.

Specifies the number of decimal places for displayed numbers.

Specifies whether separators in numbers in numbers should be displayed.

Specifies whether text should be wrapped in cells.

Specifies whether the selected cells should be protected.

Specifies whether the contents of the selected cells should be hidden.

Specifies the cell background color for the selected cells.

Specifies the cell background pattern for the selected cells.

Turns on/off the current cell background for the selected cells.

Turns on/off the current font color for the selected cells.

Specifies whether the selected font shadow should be used.

Specifies the font shadow color.

Specifies the horizontal and vertical offset of a shadow. These values are treated as a percentage of the average character width and height.

Specifies whether the chosen cell edges should displayed for the selected cells.

Specifies the cell edges for the selected cells.

Specifies the cell edge thickness for the selected cells.

Specifies the frame color.

Specifies whether you want to use round frames.

Specifies whether you want to display a frame around the current selection.

Specifies the group of formulas.

Displays all formulas from the selected group. To paste one of them into the active document double-click it.

Displays descriptions of formulas.

Specifies the current password. The password can consist of 15 characters.

Specifies the current password. The password can consist of 15 characters.

Specifies whether you want the password to be checked only when saving the document. Otherwise it will be checked each time you try to open the document.

Specifies the current password.

Displays the list of installed printers.

Specifies whether you want to print to a file.

Specifies what should be printed: (1) all pages of the current worksheet, (2) selected pages and page ranges or (3) selected cells or objects (charts, OLE objects, text frames).
You can enter up to 499 charaters in the "Pages" field.

Specifies the number of copies of each printed page.

Specifies whether you would like to collate copies of printed pages.

Specifies whether printing should begin with the last page.

Specifies the current page size.

Specifies the current paper source.

Specifies the page orientation.

Specifies how many pages should be printed on one sheet.

Specifies page margins. Typically, their minimal values depend on the selected printer, page format and orientation.

Specifies whether printed pages should be centered within the printable area of a sheet.

Specifies page headers and footers. They can contain up to 499 characters including the following special codes:

&f	Filename
&w	Worksheet name
&s	File date
&z	File time
&d	Current system date
&e	Current system date (long format)
&t	Current system time
&a	Sheet number
&p	Page number (can contain multiple numbers if more than one page per sheet is printed)
&l	Left aligned (starting from the next character and up to the end or to &r/&c characters)
&r	Right aligned (starting from the next character and up to the end or to &l/&c characters)
&c	Centered (starting from the next character and up to the end or to &l/&r characters)

Specifies the position of headers and footers. The entered values are treated as a distance from the top and bottom paper edges. Typically, their minimal values depend on the selected printer, page format and orientation.

Specifies the font for headers/footers.

Specifies the magnification factor for the printed pages. The "100%" value corresponds with the screen in the 640x480 resolution. The "Scale to fit to page" option enables you to automatically shrink the page size so that the entire worksheet can be printed on one page.

Specifies whether headers and footers should be printed on the first page (that is, the page with the "1" number; not necessarily the first printed page).

Specifies the number of the first page.

Specifies whether and how row headings should be printed.

Specifies whether and how column headings should be printed.

Specifies which rows should not be printed. You can enter up to 499 charaters.

Specifies which columns should not be printed. You can enter up to 499 charaters.

Specifies whether gridlines should be printed.

Specifies whether cell backgrounds should be printed.

Specifies whether objects should be printed. If you clear that option, the printout will contain blank rectangles in place of inserted charts, OLE objets and text frames.

To choose a new data range select cells and click the desirable button.

Specifies the author name.

Specifies the notes which will be stored with the active document file.

Specifies the title of the active document.

Specifies the subject of the active document.

Specifies the document size.

Displays the date of the last saving.

Displays the total number of folders.

Displays the total number of worksheets.

Displays the size of the current worksheet.

Displays the date when the current worksheet was modified.

Displays the number of cells containing text, formulas and numbers.

Specifies the recurrence level. This is the number of circular recalculations the program performs after finding the recurrence references. For example, if A1 contains A1+1 and this value is equal to 10, the final value (in A1) will be 10. The list of cells causing such iterations is displayed as a status bar message.

Automatic recalculation

Specifies whether all formulas should be recalculated after each action which may change the document contents. If this option is not selected, formulas are recalculated after using the **Recalculate** command from the **Tools** menu.

After pressing Enter move right/down

Specifies how the current selection should be scrolled after pressing the Enter key.

Display values of formulas

Specifies whether to display computed values of formulas.

Use only True-Type fonts to format cells

Specifies whether the program should use only True-Type fonts when formatting. If you don't select this option, you will be able to apply any currently installed Windows font.

Modify cell addresses when formulas are copied

Specifies whether cell addresses should be modified when formulas are copied. For example, the formula "C5+1" in the A1 cell will be convert to "D6+1" after it's copied to the B1 cell. This option doesn't affect row or column numbers beginning with the "$" operator.

Automatically convert constant expressions

Specifies whether formulas having constant values (i.e. not containing cell addresses) should be automatically converted to numbers or text values.

Use AutoComplete in drop-down lists

Specifies whether the AutoComplete feature should be used when you edit cells with attached drop-down lists. If you select this option, GS-Calc will search the drop-down list for text beginning with the characters you've already entered, then - if it finds such an list item - the remaining characters are automatically added to the edited text.

Specifies the undo level. You may enter values from 0 to 20000.

Specifies the default folder that will be initially used in the "File Open" and "File Save As" dialog boxes.

Specifies the displaying scale.

Specifies the character set you want to use when opening or saving the text file. If you choose DOS character set, GS-Calc will convert text from the DOS to the ANSI set.

Select the type of the cell separator you want to use when opening or saving a text file.

Specifies whether strings read from a text file should be recognized as numbers and/or formulas. If you select the "None" option, all not empty cells will contain text labels. Using this option results in much faster loading text files (similarly, the "Numbers" option is faster than the "Formulas and numbers" option).

Specifies whether formulas or their values should be saved to text files.

Specifies whether formatted (containing separators and currency or percent symbols) numbers should be saved to text files.

Specifies whether all worksheet from the current folder and its subfolders should be saved to a text file.

Operators

Operator	**Operation**	**Comments**	**Precedence**
=	Equal	Compares numbers or text strings (the comparison is not case-sensitive). Example: A1 = 4 , B2 = "abc"	7
<	Less than	Compares numbers or text strings (the comparison is not case-sensitive). Example: A1 < 4 , B2 < "abc"	7
>	Greater than	Compares numbers or text strings (the comparison is not case-sensitive). Example: A1 > 4 , B2 > "abc"	7
<=	Less than or equal	Compares numbers or text strings (the comparison is not case-sensitive). Example: A1<= 4 , B2 <= "abc"	7
>=	Greater than or equal	Compares numbers or text strings (the comparison is not case-sensitive). Example: A1 >= 4 , B2 >= "abc"	7
<>	Not equal	Compares numbers or text strings (the comparison is not case-sensitive). Example: A1 <> 4 , B2 <> "abc"	7
+	Addition	Adds numbers.	6
-	Subtraction	Subtracts numbers.	6
&	String concatenation	Merges text strings. Example: A1 & "abc" , "a" & "b"	6
*****	Multiplication	Multiplies numbers.	5
/	Division	Divides numbers.	5
^	To the power of	Calculates the power of.	4
-	Negative	Changes the sign of a number. Example: -A1	3
:	Range of cells	Creates a range. Example: A1:B5	2
%	Percent	Specifies a number entered as a percentage. Example: 12%	1

Specifies whether a drop-down list should be used when editing the selected cells.

Specify the name of the drop-down list to be used when editing the selected cells.

Click this button to manage lists available with the current document.

Specifies whether new values entered in the worksheet cells should be added to the selected list. Each unique value is added only once.

Drag the slider to the new position to change the size of buttons. This may be helpful if the range displayed as the button text is particularly long.

Shows the state of the currently performed operation.

Click this button to stop the current operation.

Specifies the text to be used as a search filter.

If you check the **Use simple text search** checkbox, this can be any text that will be compared against unformatted cell contents of the subsequent worksheets.

If the checkbox is cleared, this must be a valid spreadsheet expression (returning a number value) that can be evaluated for the data of the retrieved worksheets. If the returned value is different from zero and error, the worksheet is added to the found list. In the simplest form it can be a constant expression, for example:
1
which causes finding all worksheets.

Examples

A1=1998
B2:B10 >= 1000
(C2="New York") * (C5="John")
HAS_OBJECT(2) + HAS_OBJECT(3)
MATCH("Alien"; A2:D7; 0)

Specifies whether the searching should be limited to the current folder and its subfolders.

Specifies how to search for worksheets.

If you check the **Use simple text search** checkbox, this can be any text that will be compared against unformatted cell contents of the subsequent worksheets.

If the checkbox is cleared, this must be a valid spreadsheet expression (returning a number value) that can be evaluated for the data of the retrieved worksheets. If the returned value is different from zero and error, the worksheet is added to the found list. In the simplest form it can be a constant expression, for example:
1
which causes finding all worksheets.

Examples

A1=1998
B2:B10 >= 1000
(C2="New York") * (C5="John")
HAS_OBJECT(2) + HAS_OBJECT(3)
MATCH("Alien"; A2:D7; 0)

Select this button to found the next matching worksheet. If found, it is added to the **Found Worksheets** list, then selected within the tree pane and opened.
The list of found worksheets can be sorted by name, size and date.

Select this button to find all matching worksheets. They are added to the **Found Worksheets** list. All previous items are removed.
The list of found worksheets can be sorted by name, size and date.

Click this button to display or hide the list of found worksheets.

Displays the list of found worksheets. To open the given worksheet, double-click it or use the **Open** button. The list can be sorted by name, size and date.

Click this button to open the selected worksheet.

Specifies the list name.

Specify list items. When editing, separate them using the Enter key.

Click the **Add**, **Rename** and **Remove** buttons to add, rename or remove lists. You may create up to 65,535 lists in one document. They are shared across all worksheets in the given document.

Specify the category of the data to be copied to all other worksheets in the current folder and its subfolders. This is an easy way to make the same: column widths, row heights, formats, selection, current view coordinates or page settings.

Specifies whether the long date format defined in Regional Settings should be used.

Specifies whether the chosen date/time should be inserted as a date/time number or as a formatted string.

Specifies whether the current time should be included in the inserted value.

Specifies whether the current date should be included in the inserted value.

Specifies which worksheets should be imported.

Use this option to change the width of columns within a given range. If you check the "Fit to size" check box, GS-Calc will automatically resize each of them to fit all not empty cells they contain.

Use this option to change the height of rows within a given range. If you check the "Fit to size" check box, GS-Calc will automatically resize each of them to fit all not empty cells they contain.

Text

Specifies the text displayed inside the frame.

Horz./Vert. alignment

Specifies text alignment.

Transparent text

If you check this box, the text rectangle will be filled with the current background color.

Change font

Use this button to change the current font.

Border type
Specifies the type of the frame border.
Border color
Specifies the color of the border.
Background color
Specifies the background color.
Background image
Specifies the background image. This can be a JPEG image, bitmap or fractal. If you specify the "JPEG/Bitmap - Link" item, the chosen image will be loaded from the supplied location each time you open the current worksheet.
GS-Calc can compress some 32 and 24-bit bitmaps inserted into *.gsc documents. If a bitmap is compressed, after saving the current document the "Image Properties" dialog box will display both the uncompressed and compressed image size.
Properties
Use this button to display image properties or to specify fractal parameters.

Specifies which toolbar you would like to modify.

Restores the original/default toolbar.

Displays the list of all available buttons for a given toolbar. Click the checkbox or press the spacebar to add or remove desirable buttons.

Describes an action associated with the selected button.

Functions

mathematical functions
statistical functions
lookup and reference functions
financial functions
date/time functions
text functions
logical functions
special functions

Mathematical functions

ABS
AND
BJ0
BJ1
BJN
BY0
BY1
BYN
CEIL
COMBNR
COMBNS
DEGREES
EVEN
EXP
EXPE
EXPM
FACT
GCD
GRANDTOTAL
INTERPOL
LOG
LOG10
LOGX
MAX
MAX_EX
MIN
MIN_EX
MOD
NOT
ODD
OR
PI
POW10
PRODUCT
PRODUCTIF
RADIANS
RAND
ROUND
ROUNDE
ROUNDO
ROUNDX
SCM
SGN
SHIFTL
SHIFTR
SQR
SQRT
SUBTOTAL
SUM
SUM2
SUM3
SUMIF
SUM2IF

SUM3IF
SUM_EX
TRUNC
XOR

Hyperbolic functions

ARCOSH
ARSINH
ARTGH
COSH
COSECH
SECH
SINH
TANH

Trygonometrical functions

ACOS
ASIN
ATAN
COS
SIN
TAN

Statistical functions

AVE
AVE2
AVEG
AVEH
CHI2
STD
CORREL
ENTROPY
ERF
ERFV
EST
ESTN

Text functions

BIN
CHAR
CODE
ETEXT
EXACT
FIND
HEX
LEFT
LENGTH
LOWER
LTRIM
MERGE
MID
NUM2STR
OCT
PROPER
REPEAT
REPLACE
RIGHT
RTRIM
STEXT
TEXT
TRAIL
TRIM
UPPER
VALUE

Financial functions

CTERM
FV
FVAL
PMT
PV
RATE
SLN
SYD
TERM

Date/Time functions

DATE
DATEDIF
DATEVALUE
DAY
DAYOFWEEK
DAYOFYEAR
FORMATDATE
FORMATTIME
HOUR
MINUTE
MONTH
NOW
SECOND
TIME
TIMEVALUE
TODAY
YEAR

Lookup and Reference functions

ADDRESS
AREA
CELL
CELL_EX
COLS
COUNT
COUNTV
FOLDER_COUNT
GETEXVAL
HLOOKUP
HLOOKUP_EX
INDEX
INDEX_EX
MATCH
MATCH_EX
SHEET_COUNT
OBJECT_COUNT
OFFSET
RANGE
ROWS
VLOOKUP
VLOOKUP_EX

Logical functions

HAS_OBJECT
IF
ISERROR
ISFILE
ISNUMBER
ISOPEN
ISSTRING
TEST

Special functions

OPENDOC
OPENSHEET
MESSAGE
PLAYSOUND

ABS(Number)

Returns the absolute value of **Number**.

Example: ABS(-10) = 10

SGN(Number)

Returns -1 if **Number** < 0, 0 if **Number** = 0 and 1 if **Number** > 0.

Example: SGN(10) = 1

AND(Number1; Number2)

Returns the bitwise AND of **Number1** and **Number2**.

Example: AND(12; 4) = 4

NOT(Number)

Returns the bitwise negation of *Number*.

Example: NOT(10) = -11

OR(Number1; Number2)

Returns the bitwise OR of *Number1* and *Number2*.

Example: OR(8; 4) = 12

XOR(Number1; Number2)

Returns the bitwise XOR of *Number1* and *Number2*.

Example: XOR(12; 4) = 8

ARCOSH(Number)

Returns the inverse hyperbolic cosine of *Number*.

Example: ARCOSH(1.5) = 0.96242365

ARSINH(Number)

Returns the inverse hyperbolic sine of *Number*.

Example: ARSINH(1.5) = 1.19476322

ARTGH(Number)

Returns the inverse hyperbolic tangent of *Number*.

Example: ATGH(0.5) = 0.54930614

COSECH(Number)

Returns the hyperbolic cosecant of *Number*.

Example: COSECH(1.5) = 0.46964244

COSH(Number)

Returns the hyperbolic cosine of *Number*.

Example: COSH(1.5) = 2.35240962

SECH(Number)

Returns the hyperbolic secant of *Number*.

Example: SECH(1.5) = 0.42509603

SINH(Number)

Returns the hyperbolic sine of *Number*.

Example: SINH(1.5) = 2.12927946

TANH(Number)

Returns the hyperbolic tangent of *Number*.

Example: TANH(1.5) = 0.90514825

ACOS(Number)

Returns the arccosine of *Number*.

Example: ACOS(0.5) = 1.04719755 (=PI/3)

ASIN(Number)

Returns the arcsine of *Number*.

Example: ASIN(0.5) = 0.52359878 (=PI/6)

ATAN(Number)

Returns the arctangent of Number.

Example: ATAN(1) = 0.78539816 (PI/4)

COS(Number)

Returns the cosine of *Number*.

Example: COS(PI(1)/3) = 0.5

SIN(Number)

Returns the sine of *Number*.

Example: SIN(PI(1)/6) = 0.5

TAN(Number)

Returns the tangent of *Number*.

Example: TAN(PI(1)/4) = 1

BJ0(Number)

Returns the value of the Bessel function of the first kind and zero level for *Number*.

Example: BJ0(0.9) = 0.8075238

BJ1(Number)

Returns the value of the Bessel function of the first kind and second level for *Number*.

Example: BJ1(0.9) = 0.40594955

BJN(Level; Number)

Returns the value of the Bessel function of the first kind for the given *Level* and *Number*.

Example: BJN(1;0.9) = 0.40594955

BY0(Number)

Returns the value of the Bessel function of the second kind and zero level for *Number*.

Example: BY0(0.9) = 0.00562831

BY1(Number)

Returns the value of the Bessel function of the second kind and first level for *Number*.

Example: BY1(0.9) = -0.87312658

BYN(Level; Number)

Returns the value of the Bessel function of the second kind for the given *Level* and *Number*.

Example: BY1(1; 0.9) = -0.87312658

DEGREES(Number)

Converts *Number,* which represents an angle in radians, to degrees.

Example: DEGREES(PI(1)/4) = 45

RADIANS(Number)

Converts *Number,* which represents an angle in degrees, to radians.

Example: RADIANS(45) = 0.78539816 (=PI(1)/4)

EXPM(Number)

Returns the mantissa *m* of *Number* (*m* is greater than or equal to 0.5 and less than 1) so that *Number = m*2^k*, where *k* is the exponent.

Example: EXPM(1.5) = 0.75

EXPE(Number)

Returns the exponent *k* of *Number* so that *Number* = *m*2^k*, where *m* is the mantissa.

Example: EXPE(1.5) = 1

EXP(Number)

Returns the base of the natural logarithm ("e") raised to the power of *Number*.

Example: EXP(2) = 7.3890561

FACT(Number)

Returns the factorial of *Number*: the product of all integers greater than zero and not greater than *Number*.

Example: FACT(5) = 120

GCD(Number1; Number2)

Returns the greatest common divisor of *Number1* and *Number2*.

Example: GCD(24;16) = 8

SCM(Number1; Number2)

Returns the smallest common multiplicity of *Number1* and *Number2*.

Example: SCM(12; 8) = 24

LOG10(Number)

Returns the base 10 logarithm of *Number*.

Example: LOG10(100) = 2

LOG(Number)

Returns the natural logarithm of *Number*.

Example: LOG(EXP(2)) = 2

LOGX(Number; Base)

Returns the *Base* logarithm of *Number*.

Example: LOGX(2^5; 2) = 5

MOD(Number1; Number2)

Returns the reminder of *Number1* / *Number2*.

Example: MOD(9; 4) = 1

POW10(Number)

Returns 10 to the power of *Number*.

Example: POW10(2) = 100

PI(Number)

Returns the product of *Number* and the value PI (3.141592653589).

Example: PI(1) = 6.2831853

CEIL(Number)

Returns a number representing the smallest integer greater than or equal to *Number*.

Example: CEIL(2.6) = 3

ROUND(Number)

Returns *Number* rounded to the nearest integer.

Example: ROUND(2.4) = 2

ROUNDE(Number)

Returns *Number* rounded to the nearest even number.

Example: ROUNDE(3.1) = 4

Note: The "TEST", "ETEXT", "STEXT", "ROUNDE", "ROUNDO" functions are provided for compatibility with the earlier version of GS-Calc. In GS-Calc 4.0 and later use respectively the "IF", "UPPER", "LOWER", "EVEN", "ODD" functions.

EVEN(Number)

Returns *Number* rounded to the nearest even number.

Example: EVEN(3.1) = 4

ROUNDO(Number)

Returns *Number* rounded to the nearest odd number.

Example: ROUNDO(2.1) = 3

Note: The "TEST", "ETEXT", "STEXT", "ROUNDE", "ROUNDO" functions are provided for compatibility with the earlier version of GS-Calc. In GS-Calc 4.0 and later use respectively the "IF", "UPPER", "LOWER", "EVEN", "ODD" functions.

ODD(Number)

Returns *Number* rounded to the nearest odd number.

Example: ODD(2.1) = 3

ROUNDX(Number; Precision)

Returns *Number* rounded to the factor specified by *Precision*.

Example: ROUNDX(2.1; 0.5) = 2.5

TRUNC(Number)

Returns the integer portion of *Number*.

Example: TRUNC(2.5) = 2

SHIFTL(Number; Positions)

Returns the value of *Number* after shifting its bits by *Positions* to the left.

Example: SHIFTL(2; 3) = 16

SHIFTR(Number; Positions)

Returns the value of *Number* after shifting its bits by *Positions* to the right.

Example: SHIFTR(16; 3) = 2

SQR(Number)

Returns the square of *Number*.

Example:SQR(3) = 9

SQRT(Number)

Returns the square root of *Number*.

Example: SQRT(9) = 3

INTERPOL(X_range; Y_range; Point)

Performs interpolation of a function in *Point* based on points from *X_range* and function values from *Y_range*.

Example: INTERPOL(A1:A10; B1:B10; 3) = 2.3

AREA(Range)

Returns the number of cells in *Range*.

Example: AREA(A1:B10) = 20

AVE(Range)

Returns the arithmetic mean for cells in *Range*.

Exampe: AVE(A1:B10) = 1.1

AVE2(Range)

Returns the square mean for cells in *Range*.

Example: AVE2(A1:B10) = 1.1

AVEG(Range)

Returns the geometric mean for cells in *Range*.

Example: AVEG(A1:B10) = 1.1

AVEH(Range)

Returns the harmonic mean for cells in *Range*.

Example: AVEH(A1:B10) = 1.1

CHI2(Degree; Chi2Expression)

Returns the confidence level for *Chi2Expression,* which has the chi-square distribution.

Example: CHI2(4; 1.1) = 0.89427221

STD(Degree; TExpression)

Returns the confidence level for *TExpression,* which has the Student distribution.

Example: STD(4; 1.1) = 0.33308365

COMBNR(n;k)

Returns the number of *k*-element combinations with repeating from the *n*-element set.

Example: COMBNR(5; 3) = 35

COMBNS(n; k)

Returns the number of *k*-element combinations without repeating from the *n*-element set.

Example: COMBNS(5; 3) = 10

CORREL(Range1; Range2)

Returns the correlation between numbers from *Range1* and *Range2*.

Example: CORREL(A1:A10; B1:B10) = 0.9

COUNT(Range)

Returns the number of not empty cells within the given range.

Example: COUNT(A1:B10) = 4

COUNTV(Value; SearchRange)

Returns the number of cells containing *Value* within the *SearchRange* range. The *Value* parameter can represent a number or a text string.

Example: COUNTV(2.2; A1:B10) = 3

ENTROPY(Range)

Returns the entropy of an information source based on probability factors from the given range.

Example: ENTROPY(A1:A2) = 1

ERF(Number)

Returns the integral of the Gauss function in the range {-*Number**sqrt(2), *Number**sqrt(2)}.

Example: ERF(1.5) = 0.96610809

ERFV(Number)

Returns the argument for which the ERF function returns *Number*.

Example: ERFV(0.96610809) = 1.5

EST(Range)

Returns the standard deviation of a sample from the given range.

Example: EST(A1:B10) = 1.1

ESTN(Range)

Returns the standard deviation of a population.

Example: ESTN(A1:B10) = 1.1

GRANDTOTAL(Range)

Returns the sum of values of formulas from the given range containing the SUBTOTAL function.

Example: GRANDTOTAL(A1:B10) = 12

SUBTOTAL(Range)

Returns the sum of numbers from the given range.

Example: SUBTOTAL(A1:B10) = 12

MAX(Range)

Returns the largest number in the given range.

Example: MAX(A1:B10) = 5

MAX_EX(Range)

Returns the largest number in the given range in all worksheets from the current folder and its subfolders except the active worksheet.

Example: MAX_EX(A1:B5) = 10

MIN(Range)

Returns the smallest number in the given range.

Example: MIN(A1:B10) = -1

MIN_EX(Range)

Returns the smallest number in the given range in all worksheets from the current folder and its subfolders except the active worksheet.

Example: MIN_EX(A1:B5) = 10

PRODUCT(Range)

Returns the product of not empty cells from the given range.

Example: PRODUCT(A1:B10) = 24.6

RAND(Number)

Returns a random number greater or equal to 0 and less or equal to *Number*.

Example: RAND(60) = 34.56

SUM(Range)

Returns the sum of numbers from the given range.

Example: SUM(A1:B10) = 14

SUM_EX(Range)

Returns the sum of numbers from the given range for all worksheets from the current folder and its subfolders except the active worksheet.

Example: SUM_EX(A1:B10)

SUM2(Range)

Returns the sum of squares of numbers from the given range.

Example: SUM2(A1:B10) = 34

SUM3(Range)

Returns the sum of cubes of numbers from the given range.

Example: SUM3(A1:B10) = 46.55

PRODUCTIF(Value1; Value2; ConditionsRange; InputRange)

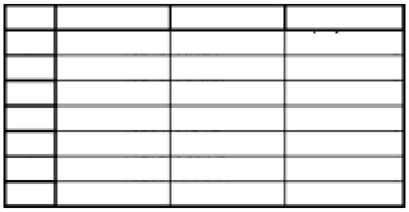
Returns the product of numbers from *InputRange*. Finds only those numbers, for which values at the same position in *ConditionRange* fulfill the requirements:
(a) value > *Value1* and value < *Value2* if *Value1* and *Value2* are not equal
or
(b) value = *Value1* if *Value1* is equal to *Value2*.
Value1 and *Value2* may represent numbers or text strings. *ConditionRage* and *InputRange* must contain the same number of cells.

Example: PRODUCTIF(10248; 10252; A2:A7; C2:C7) = 72

SUMIF(Value1; Value2; ConditionsRange; InputRange)

Returns the sum of numbers from *InputRange*. Finds only those numbers, for which values at the same position in *ConditionRange* fulfill the requirements:
(a) value > *Value1* and value < *Value2* if *Value1* and *Value2* are not equal
or
(b) value = *Value1* if *Value1* is equal to *Value2*.
Value1 and *Value2* may represent numbers or text strings. *ConditionRage* and *InputRange* must contain the same number of cells.

Example: SUMIF(10248; 10252; A2:A7; C2:C7) = 13



SUMIF2(Value1; Value2; ConditionsRange; InputRange)

Returns the sum of squares of numbers from *InputRange*. Finds only those numbers, for which values at the same position in *ConditionRange* fulfill the requirements:
(a) value > *Value1* and value < *Value2* if *Value1* and *Value2* are not equal
or
(b) value = *Value1* if *Value1* is equal to *Value2*.
Value1 and *Value2* may represent numbers or text strings. *ConditionRage* and *InputRange* must contain the same number of cells.

Example: SUM2IF(10248; 10252; A2:A7; C2:C7) = 61

SUMIF3(Value1; Value2; ConditionsRange; InputRange)

Returns the sum of cubes of numbers from *InputRange*. Finds only those numbers, for which values at the same position in *ConditionRange* fulfill the requirements:
(a) value > *Value1* and value < *Value2* if *Value1* and *Value2* are not equal
or
(b) value = *Value1* if *Value1* is equal to *Value2*.
Value1 and *Value2* may represent numbers or text strings. *ConditionRage* and *InputRange* must contain the same number of cells.

Example: SUM2IF(10248; 10252; A2:A7; C2:C7) = 307

BIN(Number)

Returns the text string representing *Number* in the binary notation.

Example: BIN(23) = "10111"

CHAR(Number)

Returns a text string consisting of one character with the character set code equal to *Number*.

Example: CHAR(65) = "A"

CODE(Text)

Returns the character set code of the first character in *Text*.

Example: CODE("Word") = 87

HEX(Number)

Returns the text string representing *Number* in the hex notation.

Example: HEX(1998) = "7ce"

KWOTA(Number)

OCT(Number)

Returns the text string representing *Number* in the octal notation.

Example: OCT(1998) = "3716"

TEXT(Number)

Returns the text string representing *Number* in textual ("verbal") form.

Example: TEXT(23) = "twenty three"

EXACT(Text1; Text2)

Compares two text strings. If *Text1* = *Text2*, 1 (True) is returned. If they are different, 0 (False) is returned. The comparison is case-sensitive.

Example: EXACT("Thing"; "Thing") = 1

FIND(Text1; Text2; Position)

Returns the position (as a 1-base index) of *Text1* in *Text2*. Searching starts in *Text2* from *Position*.

Example: FIND("word"; "two words"; 1) = 5

LEFT(Text; Number)

Returns the first *Number* characters from *Text*.

Example: LEFT("abcdef"; 3) = "abc"

LENGTH(Text)

Returns the length of *Text*.

Example: LENGTH("abcdef") = 6

LOWER(Text)

Converts all letters in *Text* to lowercase.

Example: LOWER("ABCDEF") = "abcdef"

REPLACE(Text1; Number1; Number2; Text2)

Replaces *Number2* characters in *Text1* starting from *Number1* position by *Text2*. For the first character in *Text1* *Number1* = 1.

Example: REPLACE("abcdef"; 3; 2; "x") = "abxef"

STEXT(Text)

Converts all letters in *Text* to lowercase.

Example: STEXT("ABCDEF") = "abcdef"

Note: The "TEST", "ETEXT", "STEXT", "ROUNDE", "ROUNDO" functions are provided for compatibility with the earlier version of GS-Calc. In GS-Calc 4.0 and later use respectively the "IF", "UPPER", "LOWER", "EVEN", "ODD" functions.

MID(Text; StartPosition; EndPosition)

Returns the substring of *Text*. The substring starts at *StartPosition* and includes all characters up to *EndPosition*. The position of the first character in *Text* is 1.

Example: MID("abcdef"; 3; 4) = "cd"

PROPER(Text)

Converts first letters of words in *Text* to uppercase and all other letters to lowercase.

Example: PROPER("two WORDS") = "Two Words"

REPEAT(Text; Number)

Returns string that consists of *Text* repeated *Number* times.

Example: REPEAT("abc "; 3) = "abc abc abc "

RIGHT(Text; Number)

Returns the last *Number* characters from *Text*.

Example: RIGHT("abcdef"; 3) = "def"

TRIM(Text)

Removes all trailing and leading whitespaces from Text.

Example: TRIM("   abcdef   ") = "abcdef"

LTRIM(Text)

Removes all leading whitespaces from Text.

Example: LTRIM("   abcdef") = "abcdef"

RTRIM(Text)

Removes all trailing whitespaces from Text.

Example: RTRIM("abcdef   ") = "abcdef"

UPPER(Text)

Converts all letters in *Text* to uppercase.

Example: UPPER("abcdef") = "ABCDEF"

ETEXT(Text)

Converts all letters in *Text* to uppercase.

Example: ETEXT("abcdef") = "ABCDEF"

Note: The "TEST", "ETEXT", "STEXT", "ROUNDE", "ROUNDO" functions are provided for compatibility with the earlier version of GS-Calc. In GS-Calc 4.0 and later use respectively the "IF", "UPPER", "LOWER", "EVEN", "ODD" functions.

TRAIL(Text; Number)

If *Text* length is less than *Number*, this function adds trailing spaces to *Text* so that *Text* contains exactly of *Number* characters.

Example: TRAIL("abc", 6) = "abc   "

VALUE(Text)

Converts *Text* to number.

Example: VALUE("123") = 123

NUM2STR(Number; Radix)

Returns the text string representing *Number* in the *Radix* notation.

Example: NUM2STR(1998; 10) = "1998"

MERGE(Text1; Text2)

Returns the text string consisting of merged *Text1* and *Text2*.

Example: MERGE("abc"; "def") = "abcdef"

CTERM(Rate; Fv; Pv)

Returns the number of compounding periods after which the initial amount (*Pv*) will obtain the value specified by *Fv* at the given interest rate *Rate*.

Example: CTERM(9%/12; 10000; 5000) = 92.76576606

FVAL(Pv; Rate; Periods)

Returns the future value of an investment or loan *Pv* (with no periodic payments) at the given interest rate and the given number of periods.

Example: FVAL(5000; 9%/12; 93) = 10017.51731932

FV(Payment; Rate; Periods)

Returns the future value of an investment (with periodic payments) after the given number of periods and with the given interest rate.

Example: FV(500; 9%/12; 24) = 13094.23529284

PMT(Pv; Rate; Periods)

Returns the payment amount for a loan during the given number of periods.

Example: PMT(20000; 9%/12; 24) = 913.69484558

PV(Payment; Rate; Periods)

Returns the present value for a loan or investment.

Example: PV(500; 9%/12; 24) = 10944.57306872

RATE(Fv; Pv; Periods)

Returns the periodic interest rate for a loan or investment (*Pv*) at the given number of periods.

Example: RATE(10000; 5000; 48) = 0.01454533

SLN(Cost; Salvage; Life)

Returns the straight-line depreciation for one period.

Example: SLN(3000; 400; 5) = 520

SYD(Cost; Salvage; Life; Period)

Returns the sum-of-the-years'-digits' depreciation for the given period.

Example: SYD(3000; 400; 5; 4) = 346.66666667

TERM(Payment; Rate; Pv)

Returns the number of payment periods for an investment or loan (*Pv*) with the given interest rate and payment.

Example: TERM(500; 9%/12; 10000) = 18.7047196

DATE(Year; Month; Day)

Returns the date/time number for the given *Year*, *Month* and *Day*. This integer portion represents the number of days since December 30, 1899, midnight and the fractional portion represents hours.

Example: DATE(1998; 1; 1) = 35796

TIME(Hour; Minute; Second)

Returns the date/time number for the given *Year*, *Month* and *Day*. This integer portion represents the number of days since December 30, 1899, midnight and the fractional portion represents hours.

Example: TIME(11; 53; 44) = 0.49564815

DATEDIF(DateNumber1; DateNumber2; Type)

Calculates the difference between two dates. The result depends on *Type*:
1 - difference as the number of days
2 - remaining hours of the difference
3 - remaining minutes of the difference
4 - remaining seconds of the difference
5 - difference as the total number of hours
6 - difference as the total number of minutes
7 - difference as the total number of seconds

Example: DATEDIF(DATE(1998;3;1)+TIME(10;45;0); DATE(1998;1;1)+TIME(9;45;0); 6) = 85020   (minutes)

DATEVALUE(Text)

Returns the date/time number for the given date string *Text*. To be parsed correctly, the text string must conform to the currently defined in system short date/time format.

Example: DATEVALUE("13/1/98") = 35808

TIMEVALUE(Text)

Returns the date/time number for the given time string *Text*. To be parsed correctly, the text string must conform to the currently defined in system short date/time format.

Example: TIMEVALUE("9:45:34 PM") = 0.90664352

DAY(DateNumber)

Returns the day of the month (1-31) for the *DateNumber*.

Example: DAY(DATE(1998;1;1)) = 1

HOUR(DateNumber)

Returns the hour(0-23) for the *DateNumber*.

Example: HOUR(TIME(23;12;56)) = 23

MINUTE(DateNumber)

Returns the minute (0-59) for the *DateNumber*.

Example: MINUTE(TIME(23;12;56)) = 12

SECOND(DateNumber)

Returns the second (0-59) for the *DateNumber*.

Example: SECOND(TIME(23;12;56)) = 56

MONTH(DateNumber)

Returns the month (1-12) for the *DateNumber*.

Example: MONTH(DATE(1998;1;1)) = 1

YEAR(DateNumber)

Returns the year for the *DateNumber*.

Example: YEAR(DATE(1998;1;1)) = 1998

DAYOFWEEK(DateNumber)

Returns the day of the week (Sun = 1) for the *DateNumber*.

Example: DAYOFWEEK(DATE(1998;1;1)) = 5

DAYOFYEAR(DateNumber)

Returns the day of the year (Jan 1 = 1) for the *DateNumber*.

Example: DAYOFYEAR(DATE(1998;1;1)) = 1

FORMATDATE(DateNumber; FormattingText)

Returns the text representing date/time formatted according to *FormattingText*. If *FormattingText* is an empty string (""), the default Windows format for the date representation is used. Otherwise, it may contain any text combined with the following special character sequences:

%a Abbreviated weekday name

%A Full weekday name

%b Abbreviated month name

%B Full month name

%c Date and time representation appropriate for locale

%d Day of month as decimal number (01 – 31)

%H Hour in 24-hour format (00 – 23)

%I Hour in 12-hour format (01 – 12)

%j Day of year as decimal number (001 – 366)

%m Month as decimal number (01 – 12)

%M Minute as decimal number (00 – 59)

%p Current locale’s A.M./P.M. indicator for 12-hour clock

%S Second as decimal number (00 – 59)

%U Week of year as decimal number, with Sunday as first day of week (00 – 51)

%w Weekday as decimal number (0 – 6; Sunday is 0)

%W Week of year as decimal number, with Monday as first day of week (00 – 51)

%x Date representation for current locale

%X Time representation for current locale

%y Year without century, as decimal number (00 – 99)

%Y Year with century, as decimal number

%z, **%Z** Time-zone name or abbreviation; no characters if time zone is unknown

%% Percent sign

%#c Long date and time representation

%#x Long date representation, appropriate to current locale

%#d, **%#H**, **%#I**, **%#j**, **%#m**, **%#M**, **%#S**, **%#U**, **%#w**, **%#W**, **%#y**, **%#Y** Remove leading zeros in the values explained above

Example:

FORMATDATE(DATE(1998;1;1); "") = "1/1/98"
FORMATDATE(DATE(1998;1;1); "%A, %B") = "Thursday, January"
FORMATDATE(DATE(1998;1;1)+TIME(10;30;0); "%#c") = "Thursday, January 01, 1998 10:30:00"

FORMATTIME(DateNumber; FormattingText)

Returns the text representing date/time formatted according to *FormattingText*. If *FormattingText* is an empty string (""), the default Windows format for the date representation is used. Otherwise, it may contain any text combined with the following special character sequences:

%a Abbreviated weekday name

%A Full weekday name

%b Abbreviated month name

%B Full month name

%c Date and time representation appropriate for locale

%d Day of month as decimal number (01 – 31)

%H Hour in 24-hour format (00 – 23)

%I Hour in 12-hour format (01 – 12)

%j Day of year as decimal number (001 – 366)

%m Month as decimal number (01 – 12)

%M Minute as decimal number (00 – 59)

%p Current locale’s A.M./P.M. indicator for 12-hour clock

%S Second as decimal number (00 – 59)

%U Week of year as decimal number, with Sunday as first day of week (00 – 51)

%w Weekday as decimal number (0 – 6; Sunday is 0)

%W Week of year as decimal number, with Monday as first day of week (00 – 51)

%x Date representation for current locale

%X Time representation for current locale

%y Year without century, as decimal number (00 – 99)

%Y Year with century, as decimal number

%z, **%Z** Time-zone name or abbreviation; no characters if time zone is unknown

%% Percent sign

%#c Long date and time representation

%#x Long date representation, appropriate to current locale

%#d, **%#H**, **%#I**, **%#j**, **%#m**, **%#M**, **%#S**, **%#U**, **%#w**, **%#W**, **%#y**, **%#Y** Remove leading zeros in the values described above

Example:

FORMATTIME(TIME(10;30;45); "") = "10:30:45 AM"

FORMATTIME(TIME(10;30;45); "minutes: %M, seconds: %S") = "minutes: 30, seconds: 45"

FORMATTIME(DATE(1998;1;1)+TIME(10;30;0); "%#c") = "Thursday, January 01, 1998 10:30:00"

TODAY(DaysNumber)

Returns the current date/time string +/- *DaysNumber*.

Example: TODAY(0) = "12/11/97"

NOW(HoursNumber)

Returns the current date/time string +/- *HoursNumber*.

Example: NOW(0) = "2:49:43 PM"

COLS(Range)

Returns the number of columns in *Range*.

Example: COLS(A1:B10) = 2

ROWS(Range)

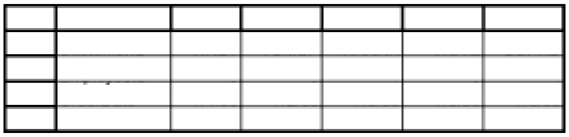
Returns the number of rows in *Range*.

Example: ROWS(A1:B10) = 10

HLOOKUP(SearchItem; SearchRange; RowIndex)

SearchItem is compared against values in the first row of *SearchRange*. If a matching value is found, the contents of the cell from the corresponding column and the *RowIndex* row are returned. If *SearchItem* can not be found in the first row of *SearchRange*, the largest value that is less than *SearchItem* is used. *SearchItem* and the cells from the first row of *SearchRange* may represent numbers or text. The data in the first row of *SearchRange* must be sorted in ascending order. The *RowIndex* numbering starts from 1 (the first row of *SearchRange*). If no matching value is found, the *error* value is returned. The search in not case-sensitive.

Example: HLOOKUP(10251; B1:F4; 3) = 3



HLOOKUP_EX(Worksheet; SearchItem; SearchRange; RowIndex)

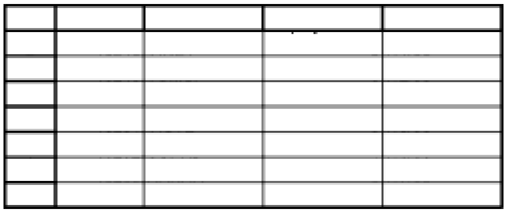
This is a version of the HLOOKUP function which enables you to specify the worksheet to search.

Example: HLOOKUP_EX("\Reports2000\June"; 10251; B1:F4; 3)

VLOOKUP(SearchItem; SearchRange; ColumnIndex)

SearchItem is compared against values in the first column of *SearchRange*. If a matching value is found, the contents of the cell from the corresponding row and the *ColumnIndex* column are returned. If *SearchItem* can not be found in the first column of *SearchRange*, the largest value that is less than *SearchItem* is used. *SearchItem* and the cells from the first column of *SearchRange* may represent numbers or text. The data in the first column of *SearchRange* must be sorted in ascending order. The *RowIndex* numbering starts from 1 (the first column of *SearchRange*). If no matching value is found, the *error* value is returned. The search in not case-sensitive.

Example: VLOOKUP(10252; A2:D7; 4) = "7/6/93"



VLOOKUP_EX(WorkSheet; SearchItem; SearchRange; ColumnIndex)

This is a version of the VLOOKUP function which enables you to specify the worksheet to search.

Example: VLOOKUP_EX("\Reports2000\June"; 10252; A2:D7; 4)

INDEX(Range; ColumnNumber; RowNumber)

Returns the contents of the cell from *Range*. The cell position is relative to *Range*, i.e. *ColumnNumber* equal to 1 and *RowNumber* equal to 1 refers to the upper left corner of *Range*.

Example: INDEX(A1:B10; 1; 1)

INDEX_EX(Worksheet; Range; ColumnNumber; RowNumber)

This is a version of the INDEX function which enables you to specify the worksheet to use.

Example: INDEX("\Reports2000\June"; A1:B10; 1; 1)

RANGE(LeftColumn; TopRow; RightColumn; BottomRow)

Creates and returns the range specified by two columns *LeftColumn*, *RightColumn* (1-512) and two rows *TopRow*, *BottomRow* (1-2,097,152).

Example: RANGE(1; 1; 2; 10)   (=A1:B10)

MATCH(SearchItem; SearchRange; SearchType)

SearchItem is compared against values in the given range. If found, the position of the matching value is returned. If there is no matching value, the *error* value is returned. The following search types may be specified:
0 - search for the first value equal to *SearchItem*
1 - search for the biggest value equal to or less than *SearchItem*
2 - search for the smallest value equal to or greater than *SearchItem*
SerachItem may represent a number or a text. The first position in *SearchRange* is equal to 1. *SearchRange* does not have to be sorted. The search in not case-sensitive.

Example:

MATCH("victe"; A2:D7; 0) = 14
MATCH(10250.5; A2:A7; 1) = 3

MATCH_EX(Worksheet, SearchItem; SearchRange; SearchType)

This is a version of the MATCH function which enables you to specify the worksheet to search.

Example: MATCH_EX("\Reports2000\June"; 10250.5; A2:A7; 1)

ADDRESS(ColumnNumber; RowNumber)

Returns the cell address specified as *ColumnNumber* (1 - 512) and *RowNumber* (1 - 2,097,152).

Example: ADDRESS(3; 4)   (=C4)

CELL(Feature; CellAddress)

Returns the chosen feature of a cell. *Feature* may have one of the following values:
0 - cell column is returned
1 - cell row is returned
2 - cell value is returned

Example: CELL(0; B5) = 2

CELL_EX(Feature; CellAddress; Worksheet)

Returns the chosen feature of a cell from another worksheet in the same file. *Feature* may have one of the following values:
0 - cell column is returned
1 - cell row is returned
2 - cell value is returned

A full path (incl. the initial backslash) must be used to access worksheets in other (non-descendant) folders. For worksheets in the same folder and its subfolders you may use a relative path.

Example:
CELL_EX(2; B5; "\Reports\July\021") = 2
CELL_EX(2; B5; "July\021") = 2
CELL_EX(2; B5; "021") = 2

OFFSET(Range; ColumnsNumber; RowsNumber)

Returns *Range* shifted horizontally and/or vertically by *ColumnsNumber* and *RowsNumber*.

Example: OFFSET(A1:B10; 1; 1)   (=B2:C11)

GETEXVAL(Document; CellAddress; AutoOpen)

Returns the value of the specified cell from another spreadsheet document and the default worksheet (that is, a worksheet that was active when the document was saved). *Document* should contain the full path of an existing *.gsc file. Before obtaining this value, *Document* must be opened. If *AutoOpen* is equal to 1, *Document* is opened automatically. If *AutoOpen* is equal to 0 and *Document* is not loaded, the *error* value is returned.

Example: GETEXVAL("d:\tmp\analyze.gsc"; E9; 1) = 100

OBJECT_COUNT(Type)

Returns the number of objects of the given *Type* (0 - any object, 1 - OLE objects, 2 - charts, 3 - text frames) in the current worksheet.

Example: OBJECT_COUNT(2) = 1

SHEET_COUNT(Type)

Returns the number of worksheets in the current folder and its subfolders if *Type* = 1 or in all folders if *Type* = 0.

Example: SHEET_COUNT(0) = 1

FOLDER_COUNT(Type)

Returns the number of folders in the current folder and its subfolders if *Type* = 1 or in all folders if *Type* = 0.

Example: FOLDER_COUNT(0) = 1

TEST(Number; TrueValue; FalseValue)

If *Number* is greater than or equal to 0, the *TrueValue* is returned. If Number is less than 0, *FalseValue* is returned. *TrueValue* and *FalseValue* may represent any text or numbers.

Example: TEST(10; "abc"; "cde") = "abc"

Note: The "TEST", "ETEXT", "STEXT", "ROUNDE", "ROUNDO" functions are provided for compatibility with the earlier version of GS-Calc. In GS-Calc 4.0 and later use respectively the "IF", "UPPER", "LOWER", "EVEN", "ODD" functions.

HAS_OBJECT(Type)

Returns 1 if the current worksheet contains an object of the given *Type* (0 - any object, 1 - OLE object, 2 - chart, 3 - text frames) or 0 otherwise.

Example: HAS_OBJECT(2) = 1

IF(Condition; TrueValue; FalseValue)

If *Condition*, which may be any expression returning numbers, is different than 0, the *TrueValue* is returned. If *Condition* is equal to 0, *FalseValue* is returned. *TrueValue* and *FalseValue* may represent any text or numbers.

Example: IF((A1>1)*(A1<10); 1; 2) = 1

ISERROR(Value)

Returns 1 (True) if *Value* represents the *error* value or a range containing at least one *error* value. If no error is found, 0 is returned.

Example: ISERROR(1/0) = 1

ISNUMBER(Value)

Returns 1 (True) if *Value* represents a number, cell or a range containing (only) numbers or expressions returning numbers. Otherwise 0 (False) is returned.

Example: ISNUMBER(12) = 1

ISSTRING(Value)

Returns 1 (True) if *Value* represents a text string or a range containing only text. Otherwise 0 (False) is returned.

Example: ISSTRING("abc") = 1

ISFILE(Text)

Returns 1 (True) if *Text* represents a name of an existing file or 0 (False) otherwise.

Example: ISFILE("d:\gscalc52\analyze.gsc") = 1

ISOPEN(Text)

Returns 1 (True) if *Text* represents a name (full path) of the currently open spreadsheet document or 0 (False) otherwise.

Example: ISOPEN("d:\gscalc52\analyze.gsc") = 1

PLAYSOUND(Condition; SoundName; Type)

If *Condition*, which may be any expression returning numbers, is different than 0, this function plays sound specified as *SoundName*. If *Type* is equal to 1, *SoundName* is assumed to be a complete file path, otherwise it's treated as a system sound name alias.

Example: PLAYSOUND(ISERROR(B15); "c:\sound\theme.wav"; 1)

OPENDOC(Condition; Document)

If *Condition*, which may be any expression returning numbers, is different than 0, this function opens a document specified as *Document*. This can be a complete file path of the given document, executable file name, name of a folder or an Internet URL.

Example:

```
OPENDOC(B15 > 10000; "d:\report.txt")
OPENDOC(B15="xyz"; "http://www.xyz.com")
OPENDOC(B15=10; "d:\gsbase\gsbase.exe")
```

OPENSHEET(Condition; SheetName)

If *Condition*, which may be any expression returning numbers, is different than 0, this function opens an existing spreadsheet document specified as *SheetName*. *SheetName* must contain a full path name of a given document.

Example: OPENSHEET(B15<100; "d:\form1.gsc")

MESSAGE(Condition; MessageText)

If *Condition*, which may be any expression returning numbers, is different than 0, this function displays a message specified as MessageText.

Example: MESSAGE(ISERROR(B15); "Error Value in B15!")

ProgID

GS-Calc.Document.6

Remarks

Use this ProgID to create a new automation object (a new empty document). For example, if you're using VC++ and smart pointers, the initializing may look like this:

```
IGSCalcPtr pDocument;

pDocument.CreateInstance(L"GS-Calc.Document.6");
```

or

```
pDocument.CreateInstance(__uuidof(Document));
```

VARIANT_BOOL Load(_bstr_t szFilePath**)**

Return Value

VARIANT_TRUE if the document was successfully opened, otherwise VARIANT_FALSE.

Parameters

szFilePath
The full path of the GS-Calc document to open.

Remarks

Call this method to open an existing document. Any error messages are displayed only if the user is in control of the application (you can change this using the GetUserControl and SetUserControl methods).

[Example]

VARIANT_BOOL LoadDatabase(_bstr_t szFilePath**, _bstr_t** szTable**, short** nType**)**

Return Value

VARIANT_TRUE if the database was successfully opened, otherwise VARIANT_FALSE.

Parameters

szFilePath
The full path of the database to open.

szTable
The name of the table to open. This parameter is used for Access databases only.

nType
The database type. You can use simple numeric values or constants defined in the type library:

```
enum DatabaseFormats
{
 dBaseIII = 1,
 dBaseIV,
 dbFoxPro,
 dbClipper,
 dbAccess2000
}
```

Remarks

Call this method to open an existing database. Any error messages are displayed only if the user is in control of the application (you can change this using the GetUserControl and SetUserControl methods). Note that before calling the Save method to save modified records you must use the SetModifiedRange method to specify the range of records or fields that should be updated.

[Example]

VARIANT_BOOL LoadTextFile(_bstr_t szFilePath**, _bstr_t** szSeparator**, short** nUseOEM**, short** nSaveFormulas**, short** nFormattedNumbers**, short** nEntireFolder, **short** nParsing**)**

Return Value

VARIANT_TRUE if the text file was successfully opened, otherwise VARIANT_FALSE.

Parameters

szFilePath
The full path of the text file to open.
szSeparator
The cell separator (a single character).
nUseOEM
Determines whether text should be convert from the DOS/OEM to Windows character set. If it's 0, no conversion is performed.
nSaveFormulas
1 to save formulas back to the text file and 0 if only their values should be saved.
nFormattedNumbers
If it's 1, all numbers will be saved as formatted strings. If it's 0, no formatting is used.
nEntireFolder
1if all worksheets from the current folder should be saved when you use the **Save** method and 0 if only the current/single worksheet should be saved.
nParsing
Specifies how numbers and formulas should be parsed. This can 0, 1 or 2:
0- all cells will contain text;
1- text strings representing numbers will be converted to numeric values/cells;
2- text strings representing numbers and formulas will be converted to numeric values and formulas.

Remarks

Call this method to open an existing text file. Any error messages are displayed only if the user is in control of the application. (you can change this using the GetUserControl and SetUserControl methods).

[Example]

VARIANT_BOOL LoadExcelFile(_bstr_t szFilePath**, short** nVersion**)**

Return Value

VARIANT_TRUE if the file was successfully opened, otherwise VARIANT_FALSE.

Parameters

szFilePath
The full path of the file to open.
nVersion
Specifies the file version: 1 for Excel 97, 2 for Excel 2000.

Remarks

Call this method to open an existing Excel file. Any error messages are displayed only if the user is in control of the application. (you can change this using the GetUserControl and SetUserControl methods).

HRESULT SetPassword(_bstr_t szPassword**)**

Parameters

szPassword

Specifies the password. If it's NULL or an empty string, the current password is removed.

Remarks

Call this method to set a new password.

VARIANT_BOOL Save( )

Return Value

VARIANT_TRUE if the document was successfully saved; otherwise VARIANT_FALSE.

Remarks

Call this method to save the current document.

[Example]

VARIANT_BOOL SaveExcelFile(_bstr_t szFilePath**, short** nVersion**)**

Return Value

VARIANT_TRUE if the file was successfully saved, otherwise VARIANT_FALSE.

Parameters

szFilePath
The full path of the file to open.
nVersion
Specifies the file version: 1 for Excel 97, 2 for Excel 2000.

Remarks

Call this method to save an Excel file. Any error messages are displayed only if the user is in control of the application. (you can change this using the GetUserControl and SetUserControl methods).

VARIANT_BOOL SaveAs(_bstr_t szFilePath**)**

Return Value

VARIANT_TRUE if the document was successfully saved; otherwise VARIANT_FALSE.

Parameters

szFilePath
The full path of the GS-Calc document to save.

Remarks

Call this message to save the current document under a new name.

[Example]

HRESULT SetModifiedRange(_bstr_t szRange**)**
HRESULT SetModifiedRangeXY(short nLeft**, short** nTop**, short** nRight**, short** nBottom**)**

Parameters

szRange
The modified cell range.
nLeft, nTop, nRight, nBottom
The column and row coordinates of the modified range (<1, 512> for columns, <1, 2,097,152> for rows).

Remarks

Call these methods to specify which records or fields should be updated when saving databases.

short GetUserControl( )
short SetUserControl(short nUserControl**)**

Return Value

Nonzero if the user is in control of the application; otherwise 0.

Parameters

nUserControl

If it's equal to 1, the user will be put in control of the application: after closing the current document, the app will remain open and visible and all error messages generated by GS-Calc will be displayed. If it's equal to 0, the automation client will "control" the application.

Remarks

Call these methods to determine whether the user is in control of the application.

[Example]

HRESULT Show( )
HRESULT Hide( )

Remarks

Call these methods to show or hide the current document. If there are no other documents open, calling the **Hide** method causes hiding the application window as well. In addition, hiding a document sets its state to "not modified". Note that most operations is performed much faster if the document and app windows stay invisible.

[Example]

HRESULT ResetDocument( )

Remarks

Call this method to delete the entire contents of the current document and remove any associated file path information.

HRESULT QuitApplication( )

Remarks

Call this method to quit the application. If you don't pass the app control to the user and don't call the Show method, the application terminates automatically after releasing all document references.

[Example]

HRESULT AddDropDownList(_bstr_t szName**, _bstr_t** szItems**)**
HRESULT RemoveDropDownList(_bstr_t szName**)**

Parameters

szName
Specifies the list name. Up to 50 characters.
szItems
Specifies the list of items separated by the "\r\n" (CR/LF) pair. For example: "1999\r\n2000\r\n2001\r\n2002"

Remarks

Call these methods to add a new drop down list to the current document.

HRESULT Delete(_bstr_t szCell**, short** nType**)**
HRESULT DeleteXY(short nLeft**, short** nTop**, short** nRight**, short** nBottom**, short** nType**)**

Parameters

szCell
The cell or range address.

nLeft, nTop, nRight, nBottom
Specifies the coordinates of the cell or range to delete (<1, 512> for columns and <1, 2,097,152> for rows).

nType
Specifies the type of the operation. You can use simple numeric values or constants defined in the type library:

```
enum DeleteTypes
{
 dFormatting = 1,       // formatting
 dNumber,               // numbers only
 dFormula,              // formulas only
 dText,                 // text cells only
 dContents              // any cell contents
}
```

Remarks

Call these methods to delete the contents or format of the specified cells.

HRESULT DeleteAll( )

Remarks

Call this method to delete the entire contents of the current worksheet.

HRESULT Evaluate(_bstr_t szExpression**, short*** nType**, double*** dNumber**, BSTR*** pText**)**

Parameters

szExpression
Specifies the expression to evaluate.

**nType*
A pointer to the variable that will contain the type of the evaluated expression (the same as in the GetType method).

**dNumber*
A pointer to the variable that will contain the computed numeric value of the expression.

**pText*
A pointer to the variable that will contain the text value of the expression. If the expression returns a numeric value, this parameter is not used.

Remarks

Call this method to find a numeric or textual value of the specified (spreadsheet) expression. If the evaluated expression returns the "ERROR" value, the (*dNumber) variable will be set to HUGE_VAL.

[Example]

short GetType(_bstr_t szCell**)**
short GetTypeXY(short nColumn**, short** nRow**)**

Return Value

The type of the cell contents.

Parameters

szCell
The cell address.
nColumn, nRow
Specify the column and row coordinates of the cell (<1, 512> for columns, <1;2,097,152> for rows).

Remarks

Call these methods to examine the type of the specified cell. The returned value described the cell type. It can be of the constant values defined in the type library:

```
enum CellTypes
{
    cNumber = 1, // number or empty cell
    cText, // text
    cNumericFormula, // formula returning numbers
    cTextFormula // formula returning text
}
```

_bstr_t GetLastColumn( )

short GetLastColumnXY( )

Return Value

The last column in the current worksheet. The "XY" variant returns the column number (from 1 to 512).

Remarks

Call these methods to get the last column in the current worksheet.

short GetLastRow( )

Return Value

The last row in the current worksheet (from 1 to 2,097,152).

Remarks

Call this method to get the last row in the current worksheet.

[Example]

HRESULT InsertColumnAt(_bstr_t szColumn**)**
HRESULT InsertColumnAtXY(short nColumn**)**

Parameters

szColumn
Specifies the column address (for example: "A").
nColumn
Specifies the number of the column (from 1 to 512).

Remarks

Call these methods to insert a new column at the specified position.

HRESULT InsertRowAt(short nRow**)**

Parameters

nRow
Specifies the number of the row (from 1 to 2,097,152).

Remarks

Call this method to insert a new row at the specified position.

VARIANT_BOOL IsCellEmpty(_bstr_t szCell**)**
VARIANT_BOOL IsCellEmptyXY(short nColumn**, short** nRow**)**

Return Value

VARIANT_FALSE if the cell is not empty; otherwise VARIANT_TRUE.

Parameters

szCell
The cell address.
nColumn, nRow
The column and row coordinates of the cell (<1, 512> for columns, <1; 2,097,152> for rows).

Remarks

Call these methods to verify whether the specified cell is empty.

VARIANT_BOOL IsColumnEmpty(_bstr_t szColumn**)**
VARIANT_BOOL IsColumnEmptyXY(short nColumn**)**

Return Value

VARIANT_FALSE if the column is not empty; otherwise VARIANT_TRUE.

Parameters

szColumn
Specifies the column address (for example: "A").
nColumn
Specifies the number of the column (from 1 to 512).

Remarks

Call these methods to verify whether the specified column contains not empty cells.

VARIANT_BOOL IsRowEmpty(short nRow**)**

Return Value

VARIANT_FALSE if the row is not empty; otherwise VARIANT_TRUE.

Parameters

nRow
Specifies the number of the row (from 1 to 2,097,152).

Remarks

Call this method to verify whether the specified row contains not empty cells.

VARIANT_BOOL PutNumber(_bstr_t szCell**, double** dNumber**)**
VARIANT_BOOL PutNumberXY(short nColumn**, short** nRow**, double** dNumber**)**
double GetNumber(_bstr_t szCell**)**
double GetNumberXY(short nColumn**, short** nRow**)**

Return Value

The "Put" methods return VARIANT_TRUE if the specified number was inserted into the cell(s); otherwise VARIANT_FALSE. The "Get" methods return the contents of the cell.

Parameters

szCell
The cell or range address.
nColumn, nRow
The column and row coordinates of the cell (<1, 512> for columns, <1; 2,097,152> for rows).
dNumber
Specifies the inserted number.

Remarks

Call these methods to insert a number into the specified cell or rage of cells or to get the contents of the cell.

[Example]

VARIANT_BOOL PutText(_bstr_t szCell**, _bstr_t** szText**)**
VARIANT_BOOL PutTextXY(short nColumn**, short** nRow**, _bstr_t** szText**)**
_bstr_t GetText(_bstr_t szCell**)**
_bstr_t GetTextXY(short nColumn**, short** nRow**)**

Return Value

The "Put" methods return VARIANT_TRUE if the specified text was inserted into the cell(s); otherwise VARIANT_FALSE. The "Get" methods return the contents of the cell.

Parameters

szCell
The cell or range address.
nColumn, nRow
The column and row coordinates of the cell (<1, 512> for columns, <1; 2,097,152> for rows).
szText
Specifies the inserted text.

Remarks

Call these methods to insert a text string into the specified cell or rage of cells or to get the contents of the cell.

[Example]

VARIANT_BOOL PutFormula(_bstr_t szCell**, _bstr_t** szFormula**)**
VARIANT_BOOL PutFormulaXY(short nColumn**, short** nRow**, _bstr_t** szFormula**)**
_bstr_t GetFormula(_bstr_t szCell**)**
_bstr_t GetFormulaXY(short nColumn**, short** nRow**)**

Return Value

The "Put" methods return VARIANT_TRUE if the specified formula was inserted into the cell(s); otherwise VARIANT_FALSE. The "Get" methods return the contents of the cell.

Parameters

szCell
The cell or range address.
nColumn, nRow
The column and row coordinates of the cell (<1, 512> for columns, <1; 2,097,152> for rows).
szFormula
Specifies the inserted formula.

Remarks

Call these methods to insert a formula into the specified cell or rage of cells or to get the contents of the cell.

[Example]

HRESULT Recalculate( )

Remarks

Call this method to recalculate all formulas within the current worksheet.

HRESULT RecalculateAll ( )

Remarks

Call this method to recalculate all formulas within all worksheets of the current document.

HRESULT RemoveColumns(_bstr_t szColumns**)**
HRESULT RemoveColumnsXY(short nStartColumn**, short** nEndColumn**)**

Parameters

szColumns
The column or range address (for example, "A" , "A1:C1")
nStartColumn, nEndColumn
The numbers of columns (from 1 to 512).

Remarks

Call these methods to remove the contents of the specified column(s).

HRESULT RemoveRows(short nStartRow**, short** nEndRow**)**

Parameters

nStartRow, nEndRow

Specify the numbers of the rows to remove (from 1 to 512).

Remarks

Call this method to remove the contents of the specified row(s).

HRESULT SetColumnWidth(_bstr_t szColumn**, short** nWidth**)**
HRESULT SetColumnWidthXY(short nStartColumn**, short** nEndColumn**, short** nWidth**)**
short GetColumnWidth(_bstr_t szColumn**)**
short GetColumnWidthXY(short nColumn**)**

Return Value

The "Get" methods return the width (in pixels) of the specified column.

Parameters

szColumn
The column or range address (for example, "A" , "A1:C1")
nStartColumn, nEndColumn, nColumn
The column numbers (from 1 to 512).
nWidth
The new width of the specified column(s).

Remarks

Call these methods to change or get the width of the specified column(s). The "Set" method accepts a column or a range as the *szColumn* parameter and the "Get" method requires an address of a single column.

HRESULT SetCurrentColumn(_bstr_t szColumn**)**
HRESULT SetCurrentColumnXY(short nColumn**)**
_bstr_t GetCurrentColumn( )
short GetCurrentColumnXY( )

Return Value

The "Get" methods return the current column in the document window.

Parameters

szColumn
The column address (for example, "A")
nColumn
The number of the column (from 1 to 512).

Remarks

Call these methods to get or set the current column (the column of the current cell in the document window).

HRESULT SetCurrentRow(short nRow**)**
short GetCurrentRow( )

Return Value

The GetCurrentRow method returns the number of the current row.

Parameters

nRow
The number of the row (from 1 to 2,097,152).

Remarks

Call these methods to get or set the current row (the row of the current cell in the document window).

HRESULT SetDocumentState(short nState**)**
short GetDocumentState( )

Return Value

The GetDocumentState returns 1 if the document has been modified since it was last saved (or, in general, since its "modification" state was last set to 0); otherwise 0.

Parameters

nState
Specifies the new state of the document: 1 if the document should be treated as modified and 0 otherwise.

Remarks

Call these methods to change or get the state of the current document. Note that the Hide, Save and SaveAs methods set the document state to "not modified".

[Example]

HRESULT SetFirstColumn(_bstr_t szColumn**)**
HRESULT SetFirstColumnXY(short nColumn**)**
_bstr_t GetFirstColumn( )
short GetFirstColumnXY( )

Return Value

The "Get" methods return the first visible column in the document window.

Parameters

szColumn
The column address (for example, "A")
nColumn
The number of the column (from 1 to 512).

Remarks

Call these methods to get or set the first visible column (the left-most column in the document window).

HRESULT SetFirstRow(short nRow**)**
short GetFirstRow( )

Return Value

The GetFirstRow method returns the first row in the document window.

Parameters

nRow
The number of the row (from 1 to 2,097,152).

Remarks

Call this method to get or set the first visible row (the top row in the document window).

HRESULT SetFormat(_bstr_t szCell**, short** nType**)**
HRESULT SetFormatXY(short nLeft**, short** nTop**, short** nRight**, short** nBottom**, short** nType**)**
short GetFormat(_bstr_t szCell**, short** nType**)**
short GetFormatXY(short nLeft**, short** nTop**, short** nRight**, short** nBottom**, short** nType**)**
HRESULT RemoveFormat(_bstr_t szCell**, short** nType**)**
HRESULT RemoveFormatXY(short nLeft**, short** nTop**, short** nRight**, short** nBottom**, short** nType**)**

Return Value

The "Get" methods return 0 if the specified cells don't have the requested format, 1 if all of them have the requested format and 3 if the chosen range contains both cell categories.

Parameters

szCell
The cell or range address (for example, "A1" , "A1:C1")

nLeft, nTop, nRight, nBottom
The column and row coordinates of the desirable range (<1, 512> for columns, <1, 2,097,152> for rows).

nType
Specifies the format type. You can use simple numeric values or constants defined in the type library (the highlighted types use some additional parameters):

```
enum FormatCommands
{
 fCurrency = 1,        // currency
 fSeparators,  // separators in numbers
 fPercent,             // percentage
 fDateTime,    // date and/or time
 fBinary,              // binary
 fHexadecimal,         // hexadecimal
 fDefinedStyle,        // custom number style
 fNormal,              // default number style
 fLeft,                // align left
 fHCenter,             // centered horizontally
 fRight,               // align right
 fTop,                 // align top
 fVCenter,             // centered vertically
 fBottom,              // align bottom
 fBold,                // bold
 fItalic,              // italic
 fStrikeOut,   // strikeout
 fUnderline,   // underline
 fNormalFont, // default font format
 fFontName,    // font name
 fFontSize,            // font size
 fFontColor,   // font color
 fBackground,// background color
 fProtected,   // cell protection
 fFrame,               // frames
 fHidden,              // hidden cells
 fMultiline,           // text displayed in multiple lines
 fDropDownList,        // drop down list
 fFixedDecimals,       // fixed number of decimal places
 fFontShadow // font shadow
}
```

Remarks

Call these methods to set, get or remove various formats for the specified cells. Some of the format types listed above use several additional variables that you have to initialize before calling the "Set" methods or that you can retrieve after calling the "Get" methods:

fDefinedStyle
BSTR PositiveNumbers // The format of positive numbers. Up to 8 characters.

BSTR NegativeNumbers // The format of negative numbers. Up to 8 characters.
// (see: Number styles).

fFontName

BSTR FontName // The font name.

fFontSize

short FontSize // The font size.

fFontColor

long FontColor // The font color specified as a RGB value.

fBackground

long BackgroundColor // The cell background color specified as a RGB value.

fDropDownList

BSTR InputListName // The name of the drop down list. Up to 50 characters.
boolean UpdateInputList // If set to VARIANT_TRUE, new items will added to the list.

fFixedDecimals

short DecimalPlaces // The number of decimal places (<0, 8>).

fFontShadow

long ShadowColor // The color of the shadow specified as a RGB value.
short HorizontalShadow // The shadow position specified as a (neg. or pos.) percentage value.
short VerticalShadow // The shadow position specified as a (neg. or pos.) percentage value.

fRame

short FrameType // The type of the frame. You can use simple numeric values or the constants defined in the type library:

```
enum FrameTypes
{
          fAllEdges = 1,      // All edges
          fTopEdge,           // top edges only
          fBottomEdge,        // bottom edges only
          fLeftEdge,          // left edges only
          fRightEdge,         // right endges
          fRoundFrame,        // round frame
          fAroundArea         // frame around the specified range
}
```

long FrameColor // The color of the frame specified as a RGB value.
short FrameStyle // The zero-based index of the frame style.

[Example]

HRESULT SetRowHeight(short nStartRow**, short** nEndRow**, short** nHeight**)**
short GetRowHeight(short nRow**)**

Return Value

The GetRowHeight method returns the width (in pixels) of the specified row.

Parameters

nStartRow, nEndRow
Specify the numbers of the rows (from 1 to 512).
nHeight
The new height of the specified row(s).

Remarks

Call these methods to change or get the height of the specified row(s).

HRESULT SetZoomFactor(short nZoom**)**
short GetZoomFactor( )

Return Value

The GetZoomFactor method returns the current magnification factor.

Parameters

nZoom
The zoom factor specified as percentage value from 10 to 400.

Remarks

Call these methods to set or get the current magnification factor of the displayed worksheet view.

HRESULT Sort(_bstr_t szRange**, short** nDescending**, short** nReverseWords**, _bstr_t** szFirstKey**, _bstr_t** szSecondKey**, _bstr_t** szThirdKey**)**
HRESULT SortXY(short nLeft**, short** nTop**, short** nRight**, short** nBottom**, short** nDescending**, short** nReverseWords**, short** nFirstKey**, short** nSecondKey**, short** nThirdKey**)**

Parameters

szRange
The cell range.
nLeft, nTop, nRight, nBottom
The column and row coordinates of the range (<1, 512> for columns, <1, 2,097,152> for rows).
nDescending
Determines the sort order: 1- descending sorting, 2 - ascending sorting.
nReverseWords
If set to 1, when comparing text cells the order of words will be reversed. If it's 0, the standard comparing method is used.
szFirstKey, szSecondKey, szThirdKey
The column addresses (for example, "A") of the subsequent sort keys. If the second and/or third key don't exist, NULL values or empty strings should be used.
nFirstKey, nSecondKey, nThirdKey
The numbers of the columns (from 1 to 512) of the subsequent sort keys. If the second and/or third key don't exist, zero values should be used.

Remarks

Call these method to sort the specified range of cells.

[Example]

HRESULT MaximizeAppWindow( )
HRESULT MinimizeAppWindow( )
HRESULT RestoreAppWindow( )

Remarks

Call these methods to maximize, minimize or restore (the previous size of) the main application window. They do nothing if the window is not visible.

[Example]

HRESULT MaximizeDocWindow( )
HRESULT MinimizeDocWindow( )
HRESULT RestoreDocWindow( )
HRESULT TileDocWindows(short nType**)**

Parameters

nType
Determines how the existing document windows should be tile: 0 - vertically; 1 - horizontally.

Remarks

Call these methods to maximize, minimize or restore (the previous size of) the current document window, or tile all visible document windows.

[Example]

short SetTreePaneWidth(short nWidth**)**
short GetTreePaneWidth( )

Return Value

The SetTreePaneWidth returns the previous width of the tree view and the GetTreePaneWidth: the current width.

Parameters

nWidth
The width of the pane.

Remarks

Call these methods to change or get the width of the worksheet tree pane.

[Example]

HRESULT Print( )

Remarks

Call this method to print the current worksheet.

[Example]

HRESULT PrintAll( )

Remarks

Call this method to print all worksheets from the current folder (that is, from the folder containing the current worksheet).

[Example]

HRESULT PrintCellRange(_bstr_t szRange**)**
HRESULT PrintCellRangeXY(short nLeft**, short** nTop**, short** nRight**, short** nBottom**)**

Parameters

szRange
The cell range.
nLeft, nTop, nRight, nBottom
The column and row coordinates of the range (<1, 512> for columns, <1, 2,097,152> for rows).

Remarks

Call these methods to print a range of cells from the current worksheet.

HRESULT PrintPageRange(long nStartPage**, long** nEndPage**)**

Parameters

nStartPage, nEndPage
The numbers of the first and last page to print.

Remarks

Call this method to print a range of pages of the current worksheet.

[Example]

HRESULT SetPageHeaders(_bstr_t szHeader**, double** dHeaderPosition**, _bstr_t** szFooter**, double** dFooterPosition**)**
HRESULT GetPageHeaders(BSTR* pHeader**, double*** pHeaderPosition**, BSTR*** pFooter**, double*** pFooterPosition**)**

Parameters

szHeader, szFooter
The text printed at the top (header) and at the bottom (footer) of each page.(see: headers and footers)
dHeaderPosition, dFooterPosition
The header and footer positions.
pHeader, pFooter
Pointers to the variables that will contain the header and footer text.
pHeaderPosition, pFooterPosition
Pointers to the variables that will contain the header and footer positions.

Remarks

Call these methods to get or set page headers, footers and their positions. The way the numeric values are interpreted (cm or inches) depends on the current Windows setting.

[Example]

HRESULT SetPageOptions(short nFirstPage**, short** nHeadings**, short** nShrinkSize**, short** nScale**)**
HRESULT GetPageOptions(short* pFirstPage**, short*** pHeadings**, short*** pShrinkSize**, short*** pScale**)**

Parameters

nFirstPage
1 if you want to start page numbering and printing headers or footers from the first page; otherwise 0.
nHeadings
1 if you want to print row and column headers; otherwise 0.
nShrinkSize
1 if the printed document size should be shrunk to the page size; otherwise 0.
nScale
The magnification of the printed document as a percentage value. 100 is the default value.
pFirstPage
A pointer to the variable that will contain the current value of the nFirstPage option.
pHeadings
A pointer to the variable that will contain the current value of the nHeadings option.
pShrinkSize
A pointer to the variable that will contain the current value of the nShrinkSize option.
pScale
A pointer to the variable that will contain the current value of the nScale parameter.

Remarks

Call these methods to set or get various page options.

[Example]

HRESULT SetPageMargins(double dLeft**, double** dRight**, double** dTop**, double** dBottom**, short** nCenterVertically**, short** nCenterHorizontally**)**
HRESULT GetPageMargins(double* pLeft**, double*** pRight**, double*** pTop**, double*** pBottom**, short*** pCenterVertically**, short*** pCenterHorizontally**)**

Parameters

dLeft, dRight, dTop, dBottom
The left, right, top and bottom page margins.
nCenterVertically
1 if the printed page should be centered vertically.
nCenterHorizontally
1 if the printed page should be centered horizontally.
pLeft, pRight, pTop, pBottom
Pointers to the variables that will contain values of the left, right, top and bottom page margins.
pCenterVertically, pCenterHorizontally
Pointers to the variables that will contain values of the *nCenterVertically* and *nCenterHorizontally* parameters.

Remarks

Call these methods to get or set the page margins and centering options for the current worksheet. The way the numeric values are interpreted (cm or inches) depends on the current Windows setting.

HRESULT ShowGridLines( )
HRESULT HideGridLines( )

Remarks

Call these methods to show or hide the worksheet grid lines. The setting affects both displaying and printing.

[Example]

short CopyWorksheet(long nWorksheetID**, long** nFolderID**, short** nFirst**, short** nLeaveOriginal**)**

Return Value

Nonzero value if the given worksheet was successfully copied; otherwise 0.

Parameters

nWorksheetID
    The ID of the worksheet to copy.
nFolderID
    The ID of the target folder. Can be 0 if the worksheet is copied to the root folder.
nFirst
    1 if the worksheet should appear as the first item in the target folder or 0 if it should be added at the end of the item list of the target folder.
nLeaveOriginal
    1 if the original worksheet should be left in its source location and 0 if the original copy should be removed.

Remarks

Call this method to copy an existing worksheet to the desirable folder.
Note: If the original item is removed, copying invalidates the ID of the copied worksheet. You must call the GetItemID method to update it and to re-use it later if necessary.

[Example]

VARIANT_BOOL DeleteItem(long nItemID**)**

Return Value

VARIANT_TRUE if the item was successfully deleted, otherwise VARIANT_FALSE.

Parameters

nItemID
The ID of the worksheet or folder to delete.

Remarks

Call this method to delete a worksheet or folder.

[Example]

HRESULT ExpandFolder(long nFolderID**)**
 HRESULT CollapseFolder(long nFolderID**)**

Parameters

nFolderID
The ID of the folder.

Remarks

Call these methods to expand or collapse folders in the tree pane. The state of the worksheet/folder tree is saved along with the document file and once you save it, it's restored after you open the document.

[Example]

long GetCurrentItem( )

Return Value

The ID of the current worksheet.

Remarks

Call this method to get the ID of the current worksheet. Moving and deleting the given worksheet or folder invalidate its ID.

long GetFirstItem(long nFolderID**)**

Return Value

The item (worksheet or folder) ID. Zero value means that there are no item in the folder.

Parameters

nFolderID
The ID of the folder whose items are to be retrieved.

Remarks

Call this method to get the ID of the first item in the desirable folder. Moving and deleting the given worksheet or folder invalidate its ID.

[Example]

long GetItemID(_bstr_t szPath**)**

Return Value

Nonzero value means the ID of the item. If it's zero, the item doesn't exist.

Parameters

szPath
The full path (with the initial backslash) of the requested worksheet or folder.

Remarks

Call this method to get the ID of the desirable item.

[Example]

_bstr_t GetItemPath(long nItemID**)**

Return Value

The full path of the given item.

Parameters

nItemID
The worksheet or folder ID.

Remarks

Call this method to get the full path of the desirable item.

long GetNextItem(long nFolderID**, long** nItemID**)**

Return Value

The item (worksheet or folder) ID. Zero value means that there are no more items in the folder.

Parameters

nFolderID
The ID of the folder whose items are retrieved.
nItemID
The ID of the previous item in the folder. This value is returned either by the GetFirstItem method or by the previous call to GetNextItem.

Remarks

Call this method to get the ID of the next item in the desirable folder. Moving and deleting the given worksheet or folder invalidate its ID.

[Example]

VARIANT_BOOL IsFolder(long nItemID**)**

Return Value

VARIANT_TRUE if the item is a folder; otherwise VARIANT_FALSE.

Parameters

nItemID

The item ID.

Remarks

Call this method to verify whether the given item is a folder.

[Example]

HRESULT InsertFolder(long nFolderID**, _bstr_t** szName**, short** nFirst**)**

Parameters

nFolderID
The ID of the target folder. Can be 0 if the root folder is the target folder.

szName
The name of the inserted folder. Up to 50 characters.

nFirst
1 if the inserted folder should appear as the first item in the target folder, or 0 if it should be added at the end of the item list of the target folder.

Remarks

Call this method to add a new folder to the document.

[Example]

HRESULT InsertWorksheet(long nFolderID**, _bstr_t** szName**, short** nFirst**)**

Parameters

nFolderID
The ID of the target folder. Can be 0 if the root folder is the target folder.
szName
The name of the inserted worksheet. Up to 50 characters.
nFirst
1 if the inserted worksheet should appear as the first item in the target folder, or 0 if it should be added at the end of the item list of the target folder.

Remarks

Call this method to add a new worksheet to the document.

[Example]

VARIANT_BOOL IsSelected(long nItemID**)**

Return Value

VARIANT_TRUE if the item ID represents the current worksheet; otherwise VARIANT_FALSE.

Parameters

nItemID
The item ID.

Remarks

Call this method to verify whether the given item is selected.

VARIANT_BOOL IsWorksheet(long nItemID**)**

Return Value

VARIANT_TRUE if the item is a worksheet; otherwise VARIANT_FALSE.

Parameters

nItemID
The item ID.

Remarks

Call this method to verify whether the given item is a worksheet.

[Example]

HRESULT SelectItem(long nItemID**)**

Parameters

nItemID
The item ID.

Remarks

Call this method to select the given worksheet.

[Example]

HRESULT SetItemName(long nItemID**, _bstr_t** szName**)**
_bstr_t GetItemName(long nItemID**)**

Return Value

The GetItemName method returns the name of the given item.

Parameters

nItemID
The item ID.
szName
The name of the item. Up to 50 characters.

Remarks

Call these methods to set or get the name of the specified item.

Editing cells (COM automation samples)

```
#include <stdio.h>
#include <math.h>
#include <tchar.h>

#import "gscalc.tlb"

using namespace GSCalcLib;

struct StartOle {
    StartOle() { CoInitialize(NULL); }
    ~StartOle() { CoUninitialize(); }
} _inst_StartOle;

void main()
{
    IGSCalcPtr pDocument;

    pDocument.CreateInstance(L"GS-Calc.Document.6");
    // or: pDocument.CreateInstance(__uuidof(Document));

    pDocument->PutText("B2", "Company A");
    pDocument->PutText("B3", "Company B");
    pDocument->PutText("B4", "Company C");
    pDocument->PutText("B5", "Company D");

    pDocument->PutNumber("C2", 111.4);
    pDocument->PutNumber("C3", 45.9);
    pDocument->PutNumber("C4", 64);
    pDocument->PutNumber("C5", 89.7);

    pDocument->PutNumber("D2", pDocument->GetNumber("C2") + 1);
    pDocument->PutNumber("D3", pDocument->GetNumber("C3") + 3);
    pDocument->PutNumber("D4", pDocument->GetNumber("C4") + 7);
    pDocument->PutNumber("D5", pDocument->GetNumber("C5") + 15);

    pDocument->Sort("B2:D5", 0, 0, "C", "D", "");

    short nType = 0;
    double dNumber = 0;
    BSTR szText = NULL;
    pDocument->Evaluate("MATCH(\"Company C\";B2:B5;0)", &nType, &dNumber, &szText);
    if ( dNumber != HUGE_VAL )
        _tprintf(_T("Company C: %.02f\n"), pDocument->GetNumberXY(3, (short)dNumber + 1));

    pDocument->PutFormula("C6", "SUM(C2:C5)");
    pDocument->PutFormula("D6", "SUM(D2:D5)");

    pDocument->SaveAs("d:\\test1.gsc");
}
```

Formatting (COM automation samples)

```
#include <stdio.h>
#include <tchar.h>

#import "gscalc.tlb"

using namespace GSCalcLib;

struct StartOle {
	StartOle() { CoInitialize(NULL); }
	~StartOle() { CoUninitialize(); }
} _inst_StartOle;

void main()
{
	IGSCalcPtr pDocument;

	pDocument.CreateInstance(L"GS-Calc.Document.6");

	pDocument->FontName = "Courier New";
	pDocument->FontSize = 16;
	pDocument->FontColor = RGB(128,0,0);
	pDocument->SetFormat("B2", fFontColor);
	pDocument->SetFormat("B2", fFontName);
	pDocument->SetFormat("B2", fFontSize);
	pDocument->SetFormat("B2", fBold);
	pDocument->PutText("B2", "Courier New, 16pt, bold");

	pDocument->PutText("B3", "Currency:");
	pDocument->SetFormat("C3:E3", fCurrency);
	pDocument->SetFormat("C3:E3", fSeparators);
	pDocument->PutNumber("C3", 1200);
	pDocument->PutNumber("D3", 2200);
	pDocument->PutNumber("E3", 3200);

	pDocument->PutText("B4", "Date:");
	pDocument->SetFormat("C4:E4", fDateTime);
	pDocument->PutFormula("C4", "DATE(1999; 8; 16)");
	pDocument->PutFormula("D4", "DATE(1999; 8; 23)");
	pDocument->PutFormula("E4", "DATE(1999; 8; 30)");

	pDocument->PutText("B5", "Defined:");
	pDocument->PositiveNumbers = "%06.f";
	pDocument->NegativeNumbers = "%05.f";
	pDocument->SetFormat("C5:E5", fDefinedStyle);
	pDocument->PutNumberXY(3, 5, 12);
	pDocument->PutNumberXY(4, 5, 25);
	pDocument->PutNumberXY(5, 5, 37);

	pDocument->PutText("B6", "Centered:");
	pDocument->SetFormat("C6:E6", fHCenter);
	pDocument->PutText("C6:E6", "text");

	pDocument->FrameType = fAroundArea;
```

```
    pDocument->FrameStyle = 2;
    pDocument->FrameColor = RGB(0,0,160);
    pDocument->SetFormat("B2:E6", fFrame);

    pDocument->BackgroundColor = RGB(255,255,200);
    pDocument->SetFormat("B2:E6", fBackground);

    pDocument->Show();
    Sleep(2000);
    pDocument->SetUserControl(1);
}
```

Editing databases (COM automation samples)

```
#include <stdio.h>
#include <tchar.h>

#import "gscalc.tlb"

using namespace GSCalcLib;

struct StartOle {
    StartOle() { CoInitialize(NULL); }
    ~StartOle() { CoUninitialize(); }
} _inst_StartOle;

void main()
{
    IGSCalcPtr pDocument;

    pDocument.CreateInstance(L"GS-Calc.Document.6");

    _tprintf(_T("\nLoading and editing sampdata.mdb..."));

    pDocument->LoadDatabase("d:\\databases\\sampdata.mdb", "orders", dbAccess2000);

    for ( int i = 2; i <= pDocument->GetLastRow(); ++i )
        if ( pDocument->GetTextXY(14, i) == _bstr_t("FRANCE") )
        {
                pDocument->PutTextXY(14, i, "France");
                pDocument->SetModifiedRangeXY(14, i, 14, i);
                pDocument->Save();
        }

    _tprintf(_T("Done.\n"));

    _tprintf(_T("Loading and editing orders.dbf..."));

    pDocument->LoadDatabase("d:\\databases\\orders.dbf", "", dBaseIV);

    for ( int j = 2; j <= pDocument->GetLastRow(); ++j )
        if ( pDocument->GetTextXY(14, j) == _bstr_t("FRANCE") )
                pDocument->PutTextXY(14, j, "France");
    pDocument->SetModifiedRangeXY(14, 2, 14, pDocument->GetLastRow());
    pDocument->Save();

    _tprintf(_T("Done.\n"));

    _tprintf(_T("Loading and editing orders.txt..."));

    pDocument->LoadTextFile("d:\\databases\\orders.txt", "\t", 0, 0, 0, 0);

    for ( int k = 2; k <= pDocument->GetLastRow(); ++k )
        if ( pDocument->GetTextXY(14, k) == _bstr_t("FRANCE") )
                pDocument->PutTextXY(14, k, "France");
    pDocument->Save();
    _tprintf(_T("Done.\n"));

    pDocument->Show();
    Sleep(1000);
    pDocument->SetUserControl(1);
}
```

Manipulating worksheets (COM automation samples)

```
#include <stdio.h>
#include <tchar.h>

#import "gscalc.tlb"

using namespace GSCalcLib;

struct StartOle {
	StartOle() { CoInitialize(NULL); }
	~StartOle() { CoUninitialize(); }
} _inst_StartOle;

void main()
{
	IGSCalcPtr pDocument;

	pDocument.CreateInstance(L"GS-Calc.Document.6");

	pDocument->Show();
	pDocument->MaximizeAppWindow();
	pDocument->MaximizeDocWindow();
	pDocument->SetTreePaneWidth(120);

	// create two empty folders
	pDocument->InsertFolder(0, "Folder 1", 0);
	pDocument->InsertFolder(0, "Folder 2", 0);

	long nFolderID1 = pDocument->GetItemID("\\Folder 1");
	long nFolderID2 = pDocument->GetItemID("\\Folder 2");

	// create ten worksheets and one folder in 'Folder 1'
	for ( int i = 0; i < 10; ++i )
	{
		TCHAR szName[20];
		_stprintf(szName, _T("sheet %d"), i + 1);
		pDocument->InsertWorksheet(nFolderID1, szName, 0);
	}
	pDocument->InsertFolder(nFolderID1, "Folder 1/1", 0);

	pDocument->ExpandFolder(nFolderID1);

	Sleep(5000);

	// move all worksheets from 'Folder 1' to 'Folder 2'
	long nItemID = pDocument->GetFirstItem(nFolderID1);
	while ( nItemID )
	{
		if ( pDocument->IsWorksheet(nItemID) )
		{
				pDocument->CopyWorksheet(nItemID, nFolderID2, 0, 0);
				nItemID = pDocument->GetFirstItem(nFolderID1);
		}
		else
```

```
                nItemID = pDocument->GetNextItem(nFolderID1, nItemID);
    }

    pDocument->ExpandFolder(nFolderID2);

    Sleep(5000);

    // delete the empty 'Folder 1/1' folder from 'Folder 1'
    pDocument->DeleteItem(pDocument->GetItemID("\\Folder 1\\Folder 1/1"));

    Sleep(5000);

    // copy all worksheets from 'Folder 2' to the root folder
    nItemID = pDocument->GetFirstItem(nFolderID2);
    while ( nItemID )
    {
        if ( pDocument->IsWorksheet(nItemID) )
                pDocument->CopyWorksheet(nItemID, 0, 0, 1);
        nItemID = pDocument->GetNextItem(nFolderID2, nItemID);
    }

    // select the 'sheet 1' worksheet in the root folder
    pDocument->SelectItem(pDocument->GetItemID("\\sheet 1"));

    pDocument->PutFormula("B2", "\"Total number of worksheets: \" & NUM2STR(SHEET_COUNT(0);10)");

    Sleep(5000);

    pDocument->Hide();
    pDocument->QuitApplication();
}
```

Printing (COM automation samples)

```
#include <stdio.h>
#include <tchar.h>

#import "gscalc.tlb"

using namespace GSCalcLib;

struct StartOle {
    StartOle() { CoInitialize(NULL); }
    ~StartOle() { CoUninitialize(); }
} _inst_StartOle;

void main()
{
    IGSCalcPtr pDocument;

    pDocument.CreateInstance(L"GS-Calc.Document.6");

    pDocument->Load("d:\\samples\\charts.gsc");

    double dHeaderPosition = 0, dFooterPosition = 0;
    BSTR szHeader = NULL, szFooter = NULL;

    pDocument->GetPageHeaders(&szHeader, &dHeaderPosition, &szFooter, &dFooterPosition);
    pDocument->SetPageHeaders("&c Document: &f, printed at: &t", dHeaderPosition, "&c Page &p",
dFooterPosition);
    pDocument->SetPageOptions(1, 1, 1, 100);
    pDocument->HideGridLines();

    pDocument->PrintAll();

    pDocument->SetPageOptions(1, 1, 0, 100);

    pDocument->PrintPageRange(1, 2);

    pDocument->SetDocumentState(0);
    pDocument->QuitApplication();
}
```

Paste Link command (Edit menu)

Use this command to paste link. This command is inactive when the Clipboard doesn't contain any link.

Links command (Edit menu)

Use this command to display the Links dialog box which lets you edit links between your document and other documents.

This command is unavailable if you have no links in your document.

Links dialog box

See: Links command

Object verb (Edit menu)

After selecting an OLE object this command is replaced by a new menu. Its contents depends on what type of object you selected.

New Object (Insert menu)

Inserts and embeds an object in a document. The application in which the object was created becomes active on the screen.

Select the object you want to insert into your document using the Insert New Object dialog box.

Insert New Object dialog box

See: New Object.

Paste Special command (Edit menu)

Use this command to paste the Clipboard contents using one of the available formats.

Paste Special dialog box

See: Paste Special command

Update command

Use this command to update the currently edited GS-Calc objects which are not updated automaticaly.

Save Copy As... command

Use this command to create and save a new copy of the currently edited GS-Calc object.

OLE object frame

If you want to change the position or size of the currently edited OLE object, move or change the size of the frame displayed around it.

Print command (File menu)

Use this command to print a document. This command presents a **Print dialog box**, where you may specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.

Print All command (File menu)

Use this command to print all worksheets from the current folder (that is, from the folder containing the current worksheet) and its subfolders.

Print dialog box

See: Print command.

Print Progress Dialog

The Printing dialog box is shown during the time that GS-Calc is sending output to the printer. The page number indicates the progress of the printing.

To abort printing, choose **Cancel**.

Print Preview command (File menu)

Use this command to display the active document as it would appear when printed. When you choose this command, the main window will be replaced with a print preview window in which one or two pages will be displayed in their printed format. The print preview toolbar offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job.

Print Preview toolbar

The print preview toolbar offers you the following options:

Print
: Bring up the print dialog box, to start a print job.

Next Page
: Preview the next printed page.

Prev Page
: Preview the previous printed page.

One Page / Two Page
: Preview one or two printed pages at a time.

Zoom In
: Take a closer look at the printed page.

Zoom Out
: Take a larger look at the printed page.

Close
: Return from print preview to the editing window.

Print Setup command (File menu)

Use this command to select a printer and a printer connection. This command presents the **Print Setup dialog box** where you specify the printer and its connection.

Print Setup dialog box

See: Print Setup command.