GS-Base 5.5 - Help Contents

Using GS-Base

Creating a new database
Modifying the structure of an existing database
Adding, deleting and editing records
Entering numbers, text, formulas, date and time
Using binary fields to store images, OLE objects and text
Copying data from one database field to another
Searching for records
Using predefined search keys
Printing tables, forms and labels
Sending e-mail messages
Verifying Internet addresses
Converting databases to HTML documents
Using drop-down lists
Changing the order of fields in tables and forms
Using definable number styles
Using Drag-Drop functions
Using linked documents
Importing and exporting databases
Changing the decimal and thousand separators
Changing the currency, date and time format
Editing the registry keys

Using COM Automation

Automation methods and properties
Sample code

Automation methods and properties

As a COM automation server, GS-Base enables you to create, open and edit its documents directly from your own applications. Creating the default automation object means creating or loading a single GS-Base database. You can use the following methods and properties (*):

Construction
ProgID

Creating and Modifying Database Structures
AddField / RemoveField
CreateDatabase
GetFieldIndex
SetFieldName / GetFieldName
SetFieldType / GetFieldType
SetFieldLength / GetFieldLength
SetDefaultValue / GetDefaultValue
GetFieldCount

Initialization, Opening, Saving and Termination
ExcludeField / IncludeField
Load
LoadDatabase
LoadTextFile
SetDocumentState / GetDocumentState
Save
SaveCopyAs
SaveCopyAsDatabase
SaveCopyAsText
SaveCopyAsHtml
SetUserControl / GetUserControl
Show / Hide
QuitApplication

Editing and Formatting
AddDropDownList / RemoveDropDownList
AddNewRecords
CompactDatabase
Delete
GetBinaryText
PutNumber / GetNumber
PutText / GetText
RemoveRecords
SetColumnWidth / GetColumnWidth
SetFormat / GetFormat / RemoveFormat
SetRowHeight / GetRowHeight
SetSelectionFlag / GetSelectionFlag

Searching, Sorting and Browsing Records
Bottom
GetRecordCount
LookUp
LookUpComplement
LookUpDuplicates
LookUpSelected

Next
Previous
RemoveAllFilters
SetCurrentField / GetCurrentField
SetFilter / GetFilter
SetCurrentRecord / GetCurrentRecord
Sort
Top

Window Size and Position
MaximizeAppWindow
MaximizeDocWindow
MinimizeAppWindow
MinimizeDocWindow
RestoreAppWindow
RestoreDocWindow
TileDocWindows

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

Sample code

Creating databases, editing and formatting fields
Searching and sorting
Saving HTML files

Creating a new database

To create a new database, use the **New** command from the **File** menu. The displayed **Database** dialog box enables you to specify the list of fields your database will contain. To add a new field, click the **Add** button and select the field type and length. You can define up to 512 text, numeric, date and binary fields.

Field type	*Length in bytes*	*Contents*
Text	1-256	Any text.
Numeric (Double-precision floating-point number)	8	Values from 1.79769313486232E308 to 4.94065645841247E–324 for negative values and 4.94065645841247E–324 to 1.79769313486232E308 for positive values. Formulas entered in numeric fields must return numbers. For example: =DATEDIF(DATEVALUE(Exp. date); DATEVALUE(TODAY(0)); 1) =CTERM(9%/12; 10000; 5000)
Date	8	Date and time in a short format, as specified by Regional Settings in Control Panel. These values are stored in a 8-byte field and may be treated as a 8-byte floating point number. The (positive or negative) integer portion represents the number of days from 30 December 1899, midnight and the fractional portion represents hours. Formulas entered in date fields must return text. For example: =FORMATDATE(DATEVALUE(TODAY(0))+TIMEVALUE(NOW(0)); "") =FORMATDATE(DATEVALUE(TODAY(1)); "")
Binary	*	Text in the RTF format or any number of JPEG images, DIB bitmap images or OLE objects. GS-Base may compress some 24- and 32-color bitmaps before saving them to a database. If a bitmap is compressed, (after it's saved) the **Image Properties** dialog box displays information about its size before and after compression.

Related topics

Entering numbers, formulas, text, date and time
Using binary fields to stores images, OLE objects and text
Inserting default values and converting field contents
Modifying the structure of an existing database
Changing the order of fields in tables and forms

Adding, deleting and editing records

Press ENTER or any letter/digit to edit the contents of the current field. To accept new data press ENTER once again or use one of the cursor keys. Pressing ESC will return the original value. GS-Base performs text length and date validation as text and date fields are modified. To copy data from another field to the edited field, press SHIFT and click that field.
If you press ENTER to edit an empty field, GS-Base automatically inserts the current default value (a number, text or formula) defined in the **Field Properties** dialog box. To insert the default value for all record at once, use the **Convert** command from the Edit menu.
All changes are saved automatically after scrolling to another record or after performing any action that changes the current selection. Using the **Save** command from the **File** menu has the same effect. If the database has been modified, it's signalized by the "**MODIFIED**" text displayed on the status bar.

To add a new record(s), scroll beyond the last record. To insert new records at the current position, use the **Insert New Records** command from the **Tools** menu.
The **Remove Records** command from the **Tools** menu enables you to delete selected records or records from a given range.

The database window can use two, three or four panes to display table, form and binary views. To activate a given pane, click it. To resize it, drag any of the line separating the panes.
To browse records, use the standard cursor keys: Arrow Left (+CTRL), Arrow Right (+CTRL), Arrow Up, Arrow Down, Page Up, Page Down, Home, End.
To scroll to the last record, press CTRL + Page Down or CTRL + END. To scroll to the first record, press CTRL + Page Up or CTRL + HOME.
To scroll to the given record, enter its number in the edit field displayed below the table or form view and press ENTER.

You can a (mouse) wheel to perform scrolling or zooming. In Windows 2000 pressing the x-buttons in any of the view panes scrolls records forward and backward.

Notes

* Inserting records (as opposed to adding records at the end of the current record set) and removing records can't be undone.
* You can change the size of each displayed pane. However, such changes are not treated as operations that modify the database contents. To keep them, use the **Save** command before closing the database window.

Related Topics

Entering numbers, text, formulas, date and time
Using binary fields to store images, OLE objects and text
Copying data from one database field to another
Searching for records

Entering numbers, text, formulas, date and time

By default, numbers with more than 15 digits are displayed in an exponential format, unless you specify a user defined style for the given field. If a number can't be fully displayed in a table column, the "#" characters are displayed. You may use both the point "." and the currently defined decimal separator to input decimal places in numbers.
You can enter any formula consisting of numbers, operators, functions and other field names (a field name used within formulas can't not start with digits). GS-Base evaluates such an expression and saves the computed value.
Note: To enter an expression, you must use = or + as the first character and text strings must be enclosed in quotation marks (" ").

GS-Base uses Windows Regional Settings to validate text entered in date/time fields. For example, if you define the short date and time format as "M/d/yy" and "h:mm:ss tt", the following expressions will be valid: "1/01/2099", "1/1/1999", "11:00 AM 1/1/1680", "1/01/01 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 – 31 December 9999.

Examples:

numbers	**text/date**	**formulas (*)**
5.56	abc	=first_name & " " & last_name
=5.56	="abc"	=CTERM(9%/12; 10000; 5000)
12.45e+1	2/1/98	=IF(DATEVALUE(joined_on) >= DATEVALUE("1/1/00"); "new";
2	11:00 AM	"existing")
-1995	1/01/97 11:00	=FORMATDATE(DATEVALUE(joined_on); "%A")
23%	AM	=DATEDIF(DATEVALUE(Exp. date); DATEVALUE(TODAY(0)); 1)
		=DATEDIF(DATEVALUE(Exp. date); DATE(2001; 1; 1); 1)
		="Type: " & "N/A"
		=12 * SIN(PI(1)/8) + 1

(*) 'first_name' and 'last_name' as sample text fields; 'joined_on' and 'Exp. date' are sample date/time fields.

Related Topics

Creating a new database
Using binary fields to store images, OLE objects and text
Using definable number styles

Using binary fields to store images, OLE objects and text

You can use binary fields to store formatted (RTF) text, JPEG images, DIB bitmaps and OLE objects. A single binary field can contain up to 16 MB of text or any number (and of any size) of images or OLE objects. GS-Base may compress some 24- and 32-color bitmaps before saving them to a database. If you modify the contents of a field, it'll be saved automatically when you scroll to another record or perform any action that changes the current selection. Using the **Save** command from the **File** menu has the same effect. If the database has been modified, it's signalized by the "**MODIFIED**" text displayed on the status bar.
Binary fields are displayed in two of four panes available within the database window. They can be used to browse and edit two different binary fields at once. The tab control enables you to switch among all defined binary fields. The remaining table and form panes contain only short contents type descriptions: "**Text**" or "**Images/OLE objects**". They are updated automatically as you modify binary field contents. If you delete such a field value, both the binary field contents and field entry in a database file are removed (as opposed to editing fields within the binary view, which never removes existing field entries from database files).

Binary fields are saved to the *database_name*.BIN file. GS-Base uses a fixed size block by which the field length can be increased. If you frequently edit/delete binary field contents, you can use the **Compact Binary Fields** command in the **Database / Information** dialog box to remove the unused space from the BIN file.

To edit OLE objects or view, modify and convert images, use commands from corresponding context menus (right-clicking or double-clicking a given object).

The **Search** button enables you to perform a full-text search. You can search the whole database or limit the searching to the current record set. Scanned data includes text from binary fields and names of images defined in the **Image Properties** dialog box.

To perform a text search within the current field of the current record, use the **Find/Replace** commands from the **Edit** menu.

Notes

* When editing binary fields GS-Base activates a "local" undo buffer which works independently of the main undo/redo buffer available after activating the table or form view. The local buffer is reset each time you scroll to another record.
* The size and position of the printed binary fields can be specified in the **Page Setup** dialog box.
* If you're modifying the same binary field in two different panes at once, GS-Base first saves the contents of the upper (or left) pane and next the contents of the bottom (or right) pane, overwriting the first contents.

Related Topics

Creating a new database
Entering numbers, text, formulas, date and time

Copying data from one database field to another

To copy data from one field to another in all records in the current record set:
1. Select the target field and in the "**Field Properties / General**" dialog box specify its default as a formula containing a reference to the desirable source field. For example:

```
=Name
```

or, if the target field depends on more source fields:

```
=First_Name & " " &   Last_Name
=Unit Price * Quantity
=DATEDIF(Date1; Date2; 6)
```

2. Use the "**Edit / Convert / Insert the default value**" command for the target field.

Note: Field names used in formulas can not start with digits or decimal separators.

Searching for records

Press CTRL+ENTER, double-click the field above the column heading or click the **Search** button in the form view to change the search expression for the current field. To accept new data press ENTER. Pressing ESC returns the original value.

In the simplest form, the search expression can be a number, date or any text and entering it results in searching for records and fields containing respectively the same number, date or text. Building more complex queries is explained further in this article.
Active search expressions are signalized by the "FILTER" indicator displayed on the status bar.

Additionally, if you browse records in a table you can use the **Find**/**Replace** commands from the **Edit** menu to find and/or replace any text within the current record set. These commands sequentially scan displayed records scrolling to the found table cells.
GS-Base also enables you to perform full-text searches using binary fields. For more information about this feature, see Using binary fields to store images, OLE objects and text

Search expressions

Note: Except the expressions using the * or $ operators, the "text" parameter can refer to a text string, a number or a date/time value.

Syntax	**Searching for records which meet the following conditions:**
text	Field contents = *text* (*text* may be a prefix)
*text	Field contents must end with *text*
$text	Field contents must contain *text*
{text1;text2;...;textN}	Field contents = *text1* OR Field contents = *text2* OR ... OR Field contents = *textN*
[text1;text2;...;textN]	Field contains all of the words: text1, text2,...,textN.
[$text1;$text2;...;$textN]	Field contains all of the substrings: text1, text2,...,textN.
<text1;text2>	Field contents >= *text1* AND Field contents <= *text2*
(text1;text2)	Field contents > *text1* AND Field contents < *text2*
!text	Field contents and *text* are different (*text* may be a prefix)
!*text	Field contents don't end with *text*
!$text	Field contents don't contain *text*
!{text1;text2;...;textN}	Field contents aren't equal to *text1* AND *text2* AND ... AND *textN*
![text1;text2;...;textN]	Field doesn't contain at least one of the words: text1, text2,...,textN.
![$text1;$text2;...;$textN]	Field doesn't contain at least oneof the substrings: text1, text2,...,textN.
!<text1;text2>	Field contents < *text1* OR Field contents > *text2*
!(text1;text2)	Field contents <= *text1* OR Field contents >= *text2*

You may combine the described expressions. For example, the following expressions are valid:
{*text1;$text2;...;textN}, <text1;text2), (text1;text2>, !(text1;), !<text1;>, (;text2), <;text1).

Notes

* Do not insert spaces between arguments of the search expression if it should refer to a text field.
* You can specify whether the search should be case-sensitive clicking the "Case-sensitive search" button on the **Search Toolbar** or changing the appropriate settings in the **Options** dialog box.
* The result of the comparison is determined by the currently selected language (not by the ASCII character set order).
* If you can press and hold down the SHIFT key and click the given field, its contents will be copied to the edited search expression.
* If you want to search for records containing an empty text field, use the asterisk "*" as a search expression for this field and then use the "Complement of the current record set" predefined search key.

Example

In the following example GS-Base will search for persons
- whose numbers are greater than 10 and smaller or equal to 100 and
- whose first names start with A, B or C and
- whose last names end with "son" and
- who work in New York in companies which names begin with M.

Related Topics

Using binary fields to store images, OLE objects and text
Using predefined search keys

Using predefined search keys

GS-Base enables you to use some predefined search keys. These "special" keys can't be expressed as a list of simple search conditions. Using them you may search for duplicate or selected records, duplicate field values or for records which constitute the complement of the current record set. You may choose the predefined keys from the **Search Toolbar**.

Related Topic

Searching for records

Printing tables, forms and labels

To print records in tables, activate the table view and use the **Print** command from the **File** menu. You can print all records or a page range. The **Grid** command from the **View** menu enables you to turn on or off printing (and displaying) grid lines. One printed page can contain up to 256 rows of the printed table.

To print records vertically, activate the form view or the binary view before using the **Print** command. You can print all records, specified pages or only the current record. Note that printing one record may require more than one page. For each record you can print only one binary field at a time. Which binary field is printed is determined by the field displayed in the second (bottom or right) binary view pane.

Using the **Page Setup** command you can change page margins, adjusting, centering and specify additional information that can be printed in headers and footers: the database name, data and time, page numbers or any text. GS-Base also enables you to specify which fields should be printed, the size and position of the current binary field and many other options.

Besides printing tables and forms, GS-Base also supports printing forms, labels or reports. However, you have to use GS-Calc to design such printouts. For more information about this feature, see Printing forms, labels and reports.

Related Topics

Headers and footers
Editing the registry keys
Printing forms, labels and reports

Printing forms, labels and reports

To print user-defined labels and reports:

1. In the **Database/Forms** dialog box enter the full path of the form(s) that you've design using GS-Calc. You can enter and save up to three different paths.
2. Select the desirable form and the **Use the selected form when printing** checkbox.

Designing forms and lables

Forms and labels are spreadsheet documents created by GS-Calc. They may have any size and may contain any data, including formatting and formulas. Worksheet cells which contain the database field names are treated as references to these fields. Each field name must be placed in a separate cell. Before printing each page all valid references are replaced by the corresponding field values and all formulas are recalculated. The program automatically recognizes and groups references to the fields of the same record. Each occurrence of the same field is assumed to be a reference to the next record.

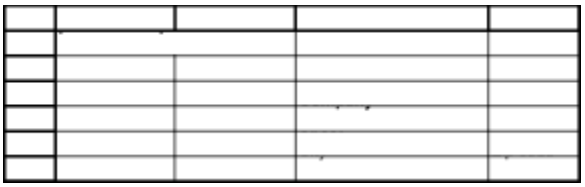
Notes

* If the numbers of printed records is greater than the number of records that the form refers to, the form will be printed as many times as it's necessary to print all data.
* If the number of printed records is smaller than the number of records that the form refers to, the remaining references will be replaced by empty cells.
* When printing forms GS-Base uses the GS-Calc document's page settings (including margins, adjusting and grid lines) rather than the default database page settings.

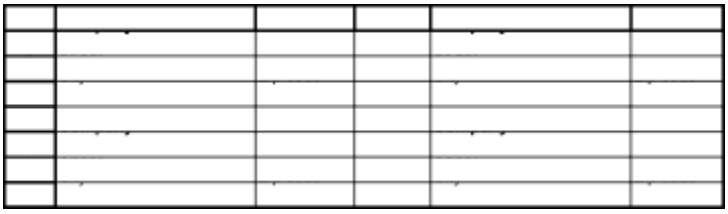
Example

The following examples show how the various forms may look when edited in GS-Calc. The "Company", "Street", "City", "Zip code", "Sales in January", "Sales in February", "Sales in March" names are sample field names.

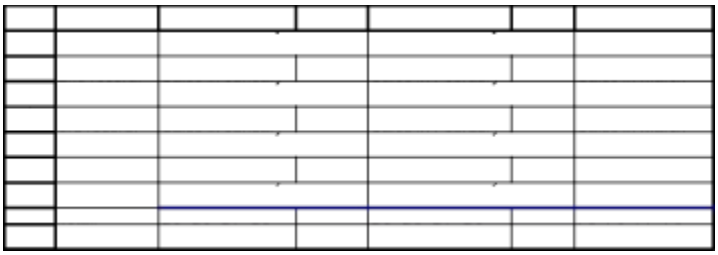
1.



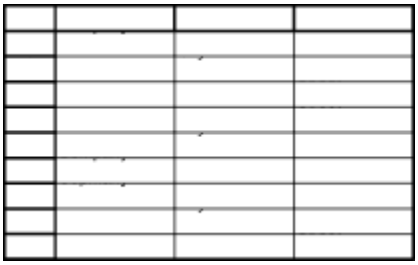
2.



3.



4.



Related Topics

Headers and footers
Editing the registry keys

Sending e-mail messages

Using GS-Base you can send bulk e-mail messages with customized forms and file attachments. It supports MIME encoding (base64, quoted-printable) and text/html message forms. After choosing the **Send E-mail Messages** command from the **Tools** menu the program displays a dialog box which enables you to specify all parameters before and even after the sending process is started. The sending procedure is performed as a separate program thread and can be run independently for each open database document.

Step 1.
Select the database field that contains the recipients' e-mail addresses. Enter your e-mail address, server name and server channel. By default, the "From" and "Reply-To" fields in the message header contain the same e-mail address. If you want to use a different one for the "Reply-To" field, add it to the **Email Address** edit field, separating the two addresses with a semicolon (;). Once you enter the data, you may click the **Save As Default** button to save it.
Step 2.
Select the message form. This can be any text or HTML document. The message form may contain references to the database fields. The embedded fields are represented by their names enclosed in double curly braces {{,}}. For example, if there are two fields: Customer and Number, the form may look like this:

Dear {{Customer}},
Here is your registration number: {{Number}}.

You can specify a character set and text encoding for your messages. In general, you should use the "**quoted-printable**" encoding for **html forms**. File attachments can be specified as files or database fields containing full file paths. The second method enables you to customize the list of attachments for each message (recipient). Optionally, you can change the file attachment description which will appear in the message headers.

There are no limits for the form size, the number of references in a single message or for the number of attachments.

Step 3.
Click the **Send** button to start sending messages. You can also earlier click the **Save As File** button to save all or part of them to a text file. As soon as you start e-mailing, all e-mail addresses and values of the embedded fields are saved in memory, so you may continue editing the current file. However, if you modify the current record set, suspend and then resume the process (pressing the **Stop** button and then again the **Send** button), the updated data will be used. Similarly, you can always change the message form, file attachments and other settings after pressing the **Stop** button.

Verifying Internet addresses

GS-Base enables you to verify Internet addresses stored in database fields. This procedure is performed as a separate program tread and can be run independently for each open database. To verify URLs from the current record set:

1. Scroll to the database field that contains the Internet addresses. If you're connecting to the Internet through a proxy server, click the **Options** button and specify the proxy/cache server name and the port number.

2. Click the **Start** button to start verifying URLs. At this moment all addresses are saved in memory, so you may continue editing the current file. However, if you modify the current record set, suspend and then resume the process (pressing the **Stop** button and then again the **Start** button), the updated data will be used.
During this process, GS-Base shows its current state, estimated time left and information about checked URLs: their names, recent modification (if that date is available) and their status.
You can sort the output list by name, date or status. To scroll to the record containing the given URL, double-click it or use the **Go To** button.

Converting databases to HTML documents

To save the current database as a HTML file(s), use the **Save Copy As** command from the **File** menu choosing the "HTML files" file type.
Before saving the HTML file GS-Base displays the **Save As HTML File** dialog box which enables you to specify several options concerning the appearance and contents of the table or forms. Records can be rendered vertically in adjusted forms or horizontally in tables.

Using CSS styles GS-Base saves all formats (including fonts, colors and alignments) and column widths. You can specify the document title, table caption, colors of the document background, table background and table border, border size, which fields should be saved and whether the first column should contain record numbers.

If you specify the maximum number of records per page, the program - if necessary - creates multiple linked HTML documents containing (no more than) the given number of records. The second and the subsequent file names are created by adding (1), (2),...,(n) to the original file name.
Exported JPEG images (if any) can be saved in the same folder or in a new subfolder.

When specifying the document title and table caption you can use the following special characters:

&p	Current page number
&m	Total number of pages
&s	Total number of records
&d	Current system date
&e	Current system date - long format
&t	Current system time

If you want to minimize the HTML file size or to emphasize the character of the data, you can also save subsequent duplicate values of the same field as the given text (by default, it's the asterisk '*'). For example, for the following records:

ACME Company	product 1
ACME Company	product 2
ACME Company	product 3
New ACME Company	product 1

the table contents may look like this:

ACME Company	product 1
*	product 2
*	product 3
New ACME Company	product 1

E-mail addresses and Internet URLs placed in table captions, text fields and binary fields are automatically converted to hyperlinks. For example, the following addresses:
sales@jps-development.com
www.jps-development.com
http://www.jps-development.com
will be converted to:
<A href="mailto:sales@jps-development.com">sales@jps-development.com</A>
<A href="http://www.jps-development.com">www.jps-development.com</A>
<A href="http://www.jps-development.com">http://www.jps-development.com</A>

By default, all special characters and multiple spaces are encoded. If you want to place some unencoded text within

table captions or text / binary fields, you can enclose it in double curly braces. For example:
{{<IMG border="0" src="images/arrow.gif" alt="*" width="13" height="13">}}
You can also use this method if you don't want some Internet address to be treated as a hyperlink.

Using the **Preview** button you can quickly examine the result without closing the dialog box.
The most recent information specified in the **Save As HTML File** dialog box is saved in the database file: once you choose the conversion options you can reuse it later easily.

Notes

* In order to display all formats, your Web browser must support style sheets and column grouping.
* The saved tables use the collapsing border model.
* If you define a background color for the given field using the **Field Properties** dialog box, this color takes precedence over the specified background color of the HTML table for the respective column.
* GS-Base uses the current magnification factor to increase/decrease the size of the saved table or form.
* If you specify 0 as the border size, the cell padding will be set to 1. Otherwise, this parameter is equal to 0.
* GS-Base saves all suggested column widths but doesn't specify the fixed table width. Web browsers may try to decrease the column widths (spanning rows, if necessary) if their sum exceeds the current width of the browser's window. If you want the column widths to be always the same, you may consider adding the "table-layout: fixed" or the fixed "width" properties to the TABLE style.

Using linked documents

If you enter a document name in a text field, you can later easily open or print that document. The entered name may specify a file on a local disk, a folder or an Internet URL. Using the **Open Document** and **Print Document** commands from the **Edit** menu (or from the context menu) you can for example connect to a web site or launch your e-mail client application:
http://www.jps-development.com
mailto:sales@jps-development.com

Using drop-down lists

Using the **Field Properties / Input** dialog box you can add drop-down lists to your database. They are displayed when you start editing a given field. A drop-down list can contain a constant number of predefined elements or can be automatically expanded by appending each new unique value entered to the corresponding field.
GS-Base enables you to use the same drop-down list for more than one field or to create a separate list for each field in your database (actually, you can define up to 65,535 lists for one database). Drop-down list are stored in the *database_name*.DAT files.

Notes

* The "DAT" files are created only when you define at least one list for the current database.
* The "DAT" files are not duplicated when you use the **Save Copy As** command from the **File** menu.
* Deleting the "DAT" file doesn't affect the database itself or the way it's opened, but the contents of all drop-down lists for that database will be reset.
* If - using the **Field Properties/Input/Edit** command - you delete all drop-down lists for defined for the current database, the "DAT" file will be deleted as well.

Modifying the structure of an existing database

Use the **Properties** command from the **File** menu to change the current database definition. You may add or delete fields, change their names, types and lengths. As long as you don't delete the given field, the modified database will contain the contents of that field. If you change the field type or length, GS-Base will convert the field contents accordingly.

Changing the order of fields in tables and forms

Use one of the following methods to change the default field order:

(1) Drag the header of the chosen field to the desirable position.
or
(2) Use the **Field Properties** command from the **Tools** menu. Then choose the **General** page and select the desirable field name from the field list. After changing the list selection GS-Base switches the positions of the current and selected field.

Note

* The original/default field order is restored after you use the **Properties** command from the **File** menu to change the definition of your database. If you want to change the order permanently, you can export the database to any of the supported formats and import it back to GS-Base.

Using definable number styles

Apart from some common styles like currency or percent, you can define your own number format. To define a new style, select the **New** item from the **Style** list in the **Field Properties/Format** dialog box. In the displayed **New Style** dialog box enter your formatting expressions 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	123456789101121314	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

Any text that doesn't contain the "%" characters. It 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.

Using the Drag-Drop functions

Use the Drag-Drop functions to copy/paste field contents. If you copy textual data within table and form views, you must additionally press and hold down the CTRL key while dragging the mouse pointer to a new position.

Importing and exporting databases

GS-Base 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 **Save Copy As** dialog box. To import data, choose the exact file type in the **Open** dialog box and next specify the name and path of the created GS-Base file.
Depending on the chosen file format, you can use various conversion options:

Text (*.txt, *.csv, *.tab, ...)
You can define any character as the field separator in the saved text file. Field values containing this separator, double quotation marks (") or CR/LF characters are enclosed in double quotation marks (the inner quotation marks are doubled). You may specify whether the textual contents of binary fields should be saved as plain text or in the Rich Text (RTF) format. The data representing OLE objects is saved using the Base64 encoding and is preceded by additional readable headers. You can also specify the character set, which fields should be saved and whether the first row should contain field names.
When importing text files, GS-Base automatically creates text and binary fields.

HTML document (*.htm)
See: Converting databases to HTML documents

dBase III+, dBase IV, FoxPro 2.x, Clipper (*.dbf, *dbt, *.fpt) (*)
Before exporting records you can define the dBase record structure changing field types (Character, Numeric, Float, Logical, Date, Memo), lengths and decimal places. Text from binary fields can be saved with or without RTF formatting. Images and OLE objects are not saved; instead, the "*Images / OLE objects* " text will be saved in the corresponding memo fields. Records marked as "selected" become "deleted" records in a DBF file. Imported records marked as "deleted" are displayed as "selected" in the created GSB database.

Excel 97, Excel 2000 (*.xls) (*)
Before importing Excel workbook you can specify which worksheets should be imported and whether the first row of the first worksheet contains field names. Similarly, the first row of the exported worksheet can optionally contain field names.
Saving and reading Excel files GS-Base converts all necessary formatting information including fonts, cell formats, background colors, numeric styles, column widths and row heights. By default, if the imported Excel worksheet contains formulas, GS-Base imports their current values.

Access 95-2000 (*.mdb) (*)
You can define the record structure changing field types and lengths. Text from binary fields can be saved with or without RTF formatting. Images and OLE objects are not saved; instead, the "*Images / OLE object(s)* " text will be saved in the corresponding memo fields. GS-Base opens Access databases (*.mdb) in the read/write mode. To open an Access database placed on a CD or any other read-only media, you must copy it on your hard disk and remove the read-only file attribute.

Changing the decimal and thousand separators

GS-Base uses the decimal separator and the separator between groups of digits defined in Windows Regional Settings. You can change them using Control Panel.

Changing the currency, date and time format

GS-Base uses the format defined in Windows Regional Settings. You can change them using Control Panel.

Editing the registry keys

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

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

There are three parameters that can't be changed in the program, but can be modified in the registry database:

DefaultFont
Modify this key to change the global font used by GS-Base. The default name is "Arial".
DuplicateString
Modify this key to change the string used to replace duplicate field values when printing tables.
NullString
Modify this key to change the string used to replace empty record fields of the imported Access database.

Related Topic

Printing tables, forms and labels
Importing and exporting databases

Commands

File
Edit
View
Insert
Tools
Window
Help

File menu

New
Open
Save
Save Copy As
Merge
Print
Print Preview
Print Setup
Page Setup
Password
Properties
Send
Exit

Edit menu

Undo
Redo
Repeat
Cut
Copy
Copy E-mail Addresses
Paste
Paste Special
Delete
Remove Search Key
Open Document
Print Document
Size To Content
Find
Replace
Convert

View menu

Status Bar
Standard Toolbar
Search Toolbar
Format Toolbar
Zoom
Grid
Full Screen

Insert menu

Formula
Date
Document Name
Image
OLE Object

Tools menu

Sort
Field Properties
Add To Search Keys
Search Keys
Send E-mail Messages
Verify URLs
Select Record
Select Group
Insert New Records
Remove Records
Options

Window menu

Split
Cascade
Tile Horizontally
Tile Vertically
Arrange Icons

Help menu

Contents and Index
About GS-Base

New command (File menu)

Use this command to create a new database.

Open command (File menu)

Use this command to open an existing database. You can open multiple databases at once. Use the Window menu to switch among the multiple open windows.

Close command (File menu)

Use this command to close the active document window.

Save command (File menu)

Use this command to save the active database to its current name and directory.

Save Copy As command (File command)

Use this command to create a copy of the current database. Only the currently found records are copied to the new database.

Page Setup command (File menu)

Use this command to change page layout.

Password command (File menu)

Use this command to change the current password.

Properties command (File menu)

Use this command to display and change the database properties. You may add or delete fields, change their names, types and lengths. As long as you don't delete a given field, the modified database will contain the contents of that field. If you change the field type or length, GS-Base performs all necessary conversions.

Related Topic

Creating a new database
Changing the database definition

Related Topic

Creating a new database
Changing the database definition

Related Topic

Creating a new database
Changing the database definition

Related Topic

Creating a new database
Changing the database definition

Merge command (File menu)

Use this command to import records from an existing GS-Base database. Both databases must have the same definition.

1, 2, 3, 4 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 (or fax). 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-Base session.

Undo command (Edit menu)

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

Redo command (Edit menu)

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

Cut command (Edit menu)

Use this command to remove the currently selected field and put it on the clipboard.

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

Copy command (Edit menu)

Use this command to copy the currently selected field or column onto the clipboard.

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

Copy E-mail Addresses command (Edit menu)

Use this command to save the values from the current column and copy them to the clipboard. The copied text strings are assumed to be e-mail addresses. They are stored in the format required by mailing programs. For example, if the column contains the "name1@company1.com", "name2@company2.com", ...,"name100@company100.com" text strings, the clipboard contents will be: "name1@company1.com; name2@company2.com; ...;name100@company.com". Then, the data can be easily inserted into the "To:" field of your mailing program.

Paste command (Edit menu)

Use this command to insert a copy of the clipboard contents to the currently selected field. This command is unavailable if the clipboard is empty or doesn't contain text data.

Delete command (Edit menu)

Use this command to remove the currently selected field.

Clear Search Key command (Edit menu)

Use this command to remove all search expressions for all fields and display all records.

Formula command (Insert menu)

Use this command to insert a formula into the current field. After choosing this command GS-Base displays the **Insert Formula** dialog box which enables you to retrieve all available formulas.

Document Name command (Insert menu)

Use this command to insert a file name into the current field. The displayed **Insert Document Name** dialog box enables you to browse existing files and folders. If a given field contains a document name, Internet URL or an e-mail address (along with the "mailto:" prefix) you can use the **Open Document** and **Print Document** commands from the **Edit** menu to open an associated application.

Open document command (Edit menu)

Use this command to open a document whose paths (or name) is placed in the current field. This can be a file name, folder name, Internet site name or Internet URL. In general, you may use any document as long as you have a respective app which can manage such documents.

Print Document command (Edit menu)

Use this command to print a document whose paths (or name) is placed in the current field. This can be a file name, folder name, Internet site name or Internet URL. In general, you may use any document as long as you have a respective app which can manage such documents.

Size To Content (Edit menu)

Adjust the width of the current table column.

Find command (Edit menu)

Use this command to search the active database for text. While comparing text GS-Base uses unformatted cells contents.

If you select any column before using this command, the search will be limited to this column.

Replace command (Edit menu)

Use this command to find and replace text in the active database. While comparing text GS-Base uses unformatted cells contents.

If you select any column before using this command, the search will be limited to this column.

Convert command (Edit menu)

Use this command to convert the current field in all records.

Replace dialog box

Related Topic

Replace command

Find dialog box

Related Topic

Find command

Paste Special (Edit menu)

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

Image (Insert menu)

Use this command to insert a new JPEG or DIB bitmap image to a binary field.

OLE Object (Insert menu)

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

Object Properties

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.

Object verb

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

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-Base, 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-Base.

To hide or display the Toolbar, choose the Standard Toolbar command from the **View** menu

Search Toolbar command (View menu)

Use this command to display and hide the **Search Toolbar**, which includes the list of the predefined search keys.

Search Toolbar

The Search Toolbar is displayed across the top of the application window, below the menu bar. The **Search Toolbar** provides quick mouse access to all predefined search key that may be used by GS-Base. These are search keys that can't be expressed as a list of search conditions.

To hide or display the **Search Toolbar**, choose the Search Toolbar command from the **View** menu

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-Base 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 contain five indicators:

Indicator	**Description**
Records number	The number of found records.
FILTER	You entered some search expressions.
MODIFIED	The database has been modified. You can save the changes using the save command or let the program save them automatically (for example, when you scroll to another record).
CAP	Caps Lock is on.
NUM	Num Lock is on.

Zoom command (View menu)

Use this command to change the current displaying scale.

Grid command (View menu)

Use this command to show or hide the table grid.

Full Screen (View menu)

Use this command to toggle the full screen mode on and off.

Date command (Insert menu)

Use this command to insert a date and/or time value into the current field.

Sort (Tools menu)

Use this command to sort the current record set in ascending or descending order. The sorting function treats hyphens and apostrophes in a special way in order to ensure that words such as "coop" and "co-op" stay together within a sorted record set. The comparison is case sensitive and is determined by the language selected by the user at setup time or via the control panel.

Sort Ascending command

Use this command to sort the current record set in ascending order. The sorting function treats hyphens and apostrophes in a special way in order to ensure that words such as "coop" and "co-op" stay together within a sorted record set. The comparison is case sensitive and is determined by the language selected by the user at setup time or via the control panel.

Sort Descending command

Use this command to sort the current record set in descending order. The sorting function treats hyphens and apostrophes in a special way in order to ensure that words such as "coop" and "co-op" stay together within a sorted record set. The comparison is case sensitive and is determined by the language selected by the user at setup time or via the control panel.

Field Properties command (Tools menu)

Use this command to change the properties of the current field. This command presents the **Field Properties** dialog box which enables you to change the field position, displayed name (which may differ from the field name), the default value and the field format.

Add To Search Keys (Tools command)

Use this command to add a new search key to the user-defined keys.

Search Keys (Tools command)

Use this command to organize user-defined search keys.

Send E-mail Messages command (Tools menu)

Use this command to send (bulk) e-mail messages. This command displays the "Send E-mail Messages" dialog box for the current database. You may open one such a dialog box for each open document.

Related Topic

Sending bulk e-mail messages

Verify URLs (Tools menu)

Use this command to verify Internet addresses. This command displays the "Verify URLs" dialog box for the current database. You may open one such a dialog box for each open document.

Related Topic

Verifying Internet addresses

Go To command (Tools menu)

Use this command to scroll the current row to a given record.

Select Record command (Tools menu)

Use this command to set or remove selection for the current record.

Select Group command (Tools menu)

Use this command to set or remove selection for a group of records.

Insert New Records (Tools menu)

Use this command to insert new records at the current position.

Note: This action can't be undone.

Remove Records command (Tools menu)

Use this command to remove records from the current database.

Note: This action can't be undone.

Options command (Tools menu)

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

Case-sensitive Search command

Use this command to turns on or off case-sensitive search.

Split command (Window menu)

Use this command to split the active document window into two panes which present records in a table and in a singe-record form.

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.

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, ...

GS-Base 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.

Contents and Index 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-Base and various types of reference information.

About GS-Base command (Help menu)

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

Context Help command

Use the Context Help command to obtain help on some portion of GS-Base. 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-Base window, such as another Toolbar button. The Help topic will be shown for the item you clicked.

Display As Thumbnail command

Use this command to toggle between displaying the selected image as a thumbnail image or using its actual dimensions.

Save Image As command

Use this command to save the selected image to a disk file as a JPEG or DIB bitmap image.

Image Properties command

Use this command to view the properties of the selected image. The text entered in the "Name" edit field is used during binary searches and when you're exporting HTML documents along with linked JPEG images.

Format Toolbar

This toolbar enables you to use various text formatting tools when editing text in binary fields. Click the edit control in the pane displaying the binary field to activate its buttons. Note that these buttons remain inactive if the selected binary window contains OLE objects.

Format Toolbar (View menu)

Use this command to hide or show the Format toolbar.

Titlebar

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.

Note: You can also move dialog boxes by dragging their title bars.

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.

Editing records

Editing records

Editing records

Editing records

Editing records

Editing records

Editing records

Editing records

Editing records

Installing GS-Base

System requirements: Windows 95 / 98 / ME / NT / 2000

To install GS-Base, run the SETUP.EXE program. The setup (full version) installs the following files in the GS-Base folder:
gsbase.exe
gsbase.hlp
gsbase.cnt
gsbase.tlb
ijl15.dll
sample1.gsb
sample2.gsc
sample3.gsc
readme.txt

Note:
To import and export databases in MS Access 2000 (*) format, GS-Base uses DAO 3.6.
You don't have to install the DAO components listed below if you're not planning to use this file format.
If you want to use this file format and if DAO is not currently installed in your system (if you're not sure, please try to export some Access file), you'll have to download and install the **mdac_typ.exe** (Microsoft Data Access 2.1)(*) file from www.microsoft.com.
To download the mdac_typ.exe file:
(1) go to: http://www.microsoft.com/downloads
and specify "Data Access Components SDK" as the product to search for, then choose MDAC version 2.1
or
(2) use this direct link: http://www.microsoft.com/data/download.htm

(*) 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 suggestion, 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 help available

No help is available for this area of the window.

No help available

No help is available for this message box.

Specifies the color used to display records marked as "selected". Click this button to change the current color.

Specifies the name and fields of the imported table. A check mark that appears in front of the field name indicates that the field will be included.

Specifies fields that should be imported.

Specifies the filter expression for imported records. This can be a string conforming to the SQL WHERE clause syntax. You may leave this field blank.
Examples:
(1) City>="Ne" AND Number=12
(2) City<= "W" OR Number>10
(3) (April_Sales+May_Sales+June_Sales)/3>50000
(4) Title+Name="Mr.Smith" OR Title+Name="Mr.Jones"
(5) Name LIKE "*son"

Note: You must use square brackets if the field name contains spaces or some special characters (e.g. "[Product number]").

Specifies whether the imported records should be sorted.

Specifies the field that will be used as a sort key. By default, GS-Base performs ascending sorting. If you want to perform descending sorting, add the DESC keyword to the selected field name (e.g. "Name DESC" instead of "Name").

Note: You must use square brackets if the field name contains spaces or some special characters (e.g. "[Product number]").

Specifies the field name.

Specifies the field type. If you change the initial values, GS-Base will perform any suitable 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 text from binary fields should be exported in the RTF (Rich Text Format) format. If you clear this option, only plain text without formatting will be saved.

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

Specifies how numbers should be displayed and printed.

Specifies the alignment for the table view.

Specifies whether numbers should contain separators.

Specifies the number of decimal places you want to use when displaying numbers.

Specifies whether the text fields should be displayed in multiline cells in the table view.

Specifies whether to hide the contents of the current field.

Specifies the font for the current field. Press the **Font** button to change it.

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 database. Otherwise it will be checked each time you try to open the document.

Specifies the current password.

Specifies the text printed at the top (header) and at the bottom (footer) of each printed page. It may contain up to 80 characters. You may place the following special characters within this text:

&f Filename
&s File date
&z File time
&d Current system date
&e Current system date - long format
&t Current system time
&p Page number
&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.

Specifies whether page numbering and printing headers/footers should start from the first page.

Specifies whether row and column headers should be printed.

Specifies whether the printed document should be shrinked so that it can be printed on a single page.

Specifies whether printing subsequent duplicate field values (when printing tables) should be suppressed. By default, GS-Base uses the asterisk "*" to replace the duplicated text. For example, for the following records:

ACME Company	product 1
ACME Company	product 2
ACME Company	product 3
New ACME Company	product 1

the printed table contents may look like this:

ACME Company	product 1
*	product 2
*	product 3
New ACME Company	product 1

Specifies the print scaling.

Specifies margins on the printed page.

Specifies whether printed records should be centered horizontally or vertically.

Specifies which fields should be printed.

Specifies whether the contents of the currently selected (visible) binary field should be printed when printing forms.

Specifies whether the printed contents of the binary field should be surrounded by a thin black frame.

Specifies the size and position of the printed binary field window.

Specifies author's name.

Specifies document notes.

Displays the title of the active document.

Specifies the subject of the current document.

Displays the document (file) size.

Displays the date of last saving.

Specifies the magnification factor.

Specifies the character set you want to use when opening or saving a text file. If you choose the OEM character set, GS-Base will perform conversion from the OEM to the default Windows set.

Specifies the field separator.

Specifies whether the first line of the text file should contain fields names.

Specifies which fields should be saved.

Specifies whether the text from binary fields should be exported in the RTF (Rich Text Format) format. If you clear this option, only plain text without formatting will be exported.

Specifies the record number.

Specifies the currently selected field. If you change the selection, the respective column positions will be switched.

Specifies the displayed name of the given field. By default this name and the original field name are the same.

Specifies the default value for the current field. This may be any expression containing numbers, formulas or text. If you press ENTER to edit an empty cell, the program will automatically insert this value. You may also use the **Convert** command from the **Edit** menu to insert the default values into the fields of all records at once.
Examples:
(1) =first_name & " " & last_name
(2) =CTERM(9%/12; 10000; 5000)
(3) =IF(DATEVALUE(joined_on) >= DATEVALUE("1/1/98"); "new"; "existing")
(4) =FORMATDATE(DATEVALUE(joined_on); "%A")
(4) ="Type: " & "N/A"
(5) N/A
(6) 123

Note: When entering expressions containing functions, operators or field names you must use = or + as the first character and text strings must be enclosed in quotation marks (" ").

Click this button to browse the list of all available formulas which can be used as the default value.

Displays all available methods of conversion. If you click the **OK** button, the program will change the contents of the current field according to the chosen description. This action will be performed for the current record set only.
Note: Some of the conversion types (like converting to uppercase) can be applied only to text fields.

Specifies how the program should scroll the current selection after accepting a new field contents with ENTER. By default, the current position is not changed.

Update record set when search expressions are modified

Specifies whether GS-Base should perform a new search after each change of the search expressions.

Match case when searching

Specifies whether the program should perform case sensitive comparison when searching.

Format fields using only True-Type fonts

Specifies whether only True-Type fonts should be used. If you don't choose this option, you will be able to select any installed Windows font when formatting the field contents.

Use AutoComplete in drop-down lists

Specifies whether the AutoComplete feature should be used when editing drop-down lists.

Wrap text displayed in binary fields

Specifies whether GS-Base should wrap text displayed in binary fields.

Keep the last used record set and sort order

Specifies whether GS-Base should save the record set state (which records are displayed and in which order) before closing a database and restore it after opening that database. If you clear this check box, the displayed record set will be determined only by the last used search expressions that are stored by default when the database is closed.

Specifies the undo level. You can use values between 0 and 16347.

Specifies the color used to highlight selected records. Click this button to change the current color.

Specifies the default folder for the "File / Open" and "File / Save Copy As" dialog boxes.

Displays the file size.

Displays the date when the database was created.

Displays comments. You may enter 256 characters.

Specifies the name of the selected field.

Specifies the type of the selected field.

Field type	*Length in bytes*	*Contents*
Text	1-256	Any text including letters, digits and special characters. Formulas entered to this field must return text. For example: =first_name & " " & last_name =FORMATDATE(DATEVALUE(joined_on); "%A")
Numeric (Double-precision floating-point number)	8	Values from 1.79769313486232E308 to 4.94065645841247E–324 for negative values and 4.94065645841247E–324 to 1.79769313486232E308 for positive values. Formulas entered to this field must return numbers. For example: =DATEDIF(DATEVALUE(Exp. date); DATEVALUE(TODAY(0)); 1) =CTERM(9%/12; 10000; 5000)
Date	8	Text representing time and date in the currently defined date/time format. These values are stored in a 8-byte field and may be treated as a 8-byte floating point number. The (positive or negative) integer portion represents the number of days from 30 December 1899, midnight and the fractional portion represents hours. Formulas entered to this field must return text. For example: =FORMATDATE(DATEVALUE(TODAY(0))+TIMEVALUE(NOW(0)); "") =FORMATDATE(DATEVALUE(TODAY(1)); "")
Binary	*	Text in the RTF format (up to 16 MB) or any number of OLE objects.

Specifies the length of the current field. Only lengths of text fields can be changed.

Displays all fields. If you want to change the name, type or length of a field, select that field and modify its description in the **Name**, **Type** and **Length** edit fields.

Click this button to remove the selected field from the current database.

Click this button to add a new field to the current database.

Use this command to save the database structure information to a HTML file. This summary will contain information on database files, database fields, defined search keys and drop-down lists.

Specifies the field of the current database.

Specifies the following:
- the greatest number for number fields
- the greatest length of text strings for text fields
- the latest date/time for date/time fields

Specifies the following:
- the smallest number for number fields
- the smallest length of text strings for text fields
- the earliest date/time for date/time fields

Specifies the following:
- the mean for number fields
- the average text length for text fields
This value is not used for date/time fields.

Specifies the following:
- the sum of numbers for number fields
- the sum of text lengths for text fields
This value is not used for date/time fields.

Specifies whether the displayed information should refer to the current record set or to all records.

Click this button to update statistics for the current field.

Click this button to compact (and decrease the size of) the BIN file containing binary fields.

Specifies the GS-Calc document name (full path) to be used as a printing form for GS-Base. You may store three form names. To make the given form active, select its number (1, 2 or 3) and check the **Use the selected form when printing** option.

Specifies whether the current form should be used to print records.

Click this button to browse the available GS-Calc documents which can be used as printing forms.

Click this button to edit the selected form.
Note: You must install GS-Calc to be able to edit the GS-Calc documents.

Specifies whether the selected records should be removed from the current database.

Specifies the range of records which should be removed from the current database.

Specifies the group of records which will be selected. You may choose all records or a range of records from the current record set.

Specifies whether the selection should be removed for the chosen group of records.

Specifies the separator to be used when merging fields.

Specifies the database field containing the recipients' e-mail addresses.

Specifies the name of the sender.

Specifies the e-mail address of the sender. If you enter two e-mail addresses separated with a semicolon (;), the first one will be inserted into the "From" field in a message header, and the second one into the "Reply-To" field. Otherwise, both addresses will be the same.

Specifies the organization of the sender.

Specifies the SMTP server name for outgoing messages.

Specifies the SMTP server channel. The default value is 25.

Saves the current settings.

Specifies the subject of the message.

Specifies the file to be used as a message form. This can be any text or html file. fields. The merged database fields are represented by their names enclosed in double curly braces {{,}}.
For example, if there are two fields: Customer and Number, the form may look like this:

Dear {{Customer}},
Here is your registration number: {{Number}}.

Specify the type of the message form. If you select the text/html form, you should also select the quoted-printable text encoding.

Browse available printing forms.

Specifies the character set and the text encoding for the message form.

Displays the list of file attachments.

Click this button to add a file attachment.

Removes the selected file from the list.

Specifies advanced options for the selected file attachment.

Starts sending messages.

Suspends sending messages.

Saves the messages to a text file.

Displays the status of the current process: sent messages and the estimated time needed to complete it.

Specifies the database field containing the file name (full path).

Displays the currently verified file. If the file doesn't exist or is inaccessible, the program displays a respective message.

Click this button to check whether all files pointed by the selected database field exist and are accessible.

Specifies the character set for the selected message form.

Specifies the text encoding for the selected message form.

Specifies whether the selected message forms contains references to the database fields.

Specifies the file attachment type. You may leave the default "application/octet-stream" type.

Specifies the file attachment description. This edit field is empty if it refers to a database field - the suitable description is added during the sending procedure and may be different for each message.

Specifies the "Content-Disposition" field in the attachment header. You may leave the default "Attachment" parameter.

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

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

Click this button to manage lists available for the current database.

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

Specifies the list name.

Specify the list items. When editing, you can use copying and pasting to modify it quickly.

Click the **Add**, **Rename** and **Remove** buttons to add, rename or remove lists. You may create up to 65,535 lists for one database.

Specifies the number of new records to insert at the current position. The current record and all records below it are shifted down.

Specifies the title of the HTML document. You may place the following special characters within this text:

&p Current page number
&m Total number of pages
&s Total number of records
&d Current system date
&e Current system date - long format
&t Current system time

Specifies the character set for the HTML document.

Specifies the background color of the HTML document.

Specifies the background color of the table.

Specifies the color and the size of the table border.
If you choose 0 as the border width, the 'cellpadding' attribute of the table will be set to 1, otherwise it's equal to 0.

Specifies the table caption. You may place the following special characters within this text:

&p	Current page number
&m	Total number of pages
&s	Total number of records
&d	Current system date
&e	Current system date - long format
&t	Current system time

Specifies the columns that should be saved in the HTML document. You can change their order changing the order of fields in the table or record view (see: Changing positions of columns in tables and forms).

Specifies whether subsequent duplicate values of the same field should be replaced by the defined text. This can help you to minimize the HTML document size or to emphasize the character of the data. For example, for the following original records:

ACME Company	product 1
ACME Company	product 2
ACME Company	product 3
New ACME Company	product 1

the table may look like this:

ACME Company	product 1
*	product 2
*	product 3
New ACME Company	product 1

Specifies whether the first column of the saved table should contain record numbers.

Specifies how records should be saved. When saving forms, records are rendered vertically (similarly to the record view in the document window) and GS-Base exports the (textual) contents of binary fields.

Specifies whether the program - if necessary - should create multiple HTML documents (files) containing the given maximum number of records. The appropriate links are added automatically to each page. The second and the subsequent file names are created by adding 1, 2, etc. to the original file name.

Specifies the name of the subfolder where exported images should be saved. Please enter only the actual folder name, not its full path. If you don't specify it, all image and html files will be save in the same location.

Shows the state of the currently performed operation.

Specifies the database field containing Internet addresses.

Specifies whether the only information on not existing / not available URLs should be displayed.

Click this button to start checking the URLs or to suspend the process.

Displays information on checked URLs: their names, when they were modified and their status.

The total number of not existing / not available Internet addresses.

Click this button to scroll to the record containing the selected URL.

If you're connecting to the Internet through a proxy server, click this button to specify the proxy server name and port.

Specifies the sort order.

Specifies whether the reverse order of words should be used when comparing. For example, if some fields contain two names:

Tom A. Jones
Mark Smith

GS-Base 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.

Specifies the background color for the current field.

Specifies the font color for the current field.

Specifies the text to search for.

Specifies whether the program should match whole words only.

Specifies whether the program should match case when comparing words.

Specifies whether the whole database or just the current record set should be searched.

Operators

Operator	Operation	Comments	Precedence
=	Equal	Compares numbers or text strings (the comparison is not case-sensitive). Example: field1 = 4 , field2 = "abc"	6
<	Less than	Compares numbers or text strings (the comparison is not case-sensitive). Example: field1 < 4 , field2 < "abc"	6
>	Greater than	Compares numbers or text strings (the comparison is not case-sensitive). Example: field1 > 4 , field2 > "abc"	6
<=	Less than or equal	Compares numbers or text strings (the comparison is not case-sensitive). Example: field1<= 4 , field2 <= "abc"	6
>=	Greater than or equal	Compares numbers or text strings (the comparison is not case-sensitive). Example: field1 >= 4 , field2 >= "abc"	6
<>	Not equal	Compares numbers or text strings (the comparison is not case-sensitive). Example: field1 <> 4 , field2 <> "abc"	6
+	Addition	Adds numbers.	5
-	Subtraction	Subtracts numbers.	5
&	String concatenation	Merges text strings. Example: field1 & "abc" , "a" & "b"	5
*	Multiplication	Multiplies numbers.	4
/	Division	Divides numbers.	4
^	To the power of	Calculates the power of.	3
-	Negative	Changes the sign of a number. Example: -10	2
%	Percent	Specifies a number entered as a percentage. Example: 12%	1

Specifies the format of the date that will be inserted into your document. If you check this option, the long date format currently defined in the Windows settings will be used. Otherwise the returned string will represent a date in the short format.

Specifies whether the returned string should include the current time. The time format is determined by the current Windows settings.

Specifies the number of records in a database and the date when they were edited for the last time.

Specifies the list of imported fields.

Specifies the range of records that should be imported.

Specifies whether GS-Base should perform text conversion from the OEM to the default Windows character set.

Specifies which worksheets should be imported.

Specifies whether the first row of the first imported worksheet contains field names. If you clear this check box, GS-Base will assume the first row contains data and set field names to FIELD_1, FIELD_2, FIELD_3, etc.

Specifies which fields should be exported.

Specifies whether field names should be stored in the first row of the exported worksheet.

Specifies whether text from binary fields should be exported in the RTF (Rich Text Format) format. If you clear this check box, only plain text without formatting will be saved.

Displays the list of all user-defined search keys. To modify the name or search expressions of an existing key, double-click the desirable item on that list. The total length of all items is limited to 2000 characters.

Use this button to add a new search key. By default, the current search key values are used to initialize its search expressions.

Use this button to remove the selected search key.

Specifies the edited toolbar.

Click this button to restore the default button set for the specified toolbar.

Displays the list of buttons in the specified toolbar. Click the check box below each button (or press Spacebar) to add or remove it from the toolbar.

Describes the command associated with the current button.

Specifies the current view configuration. To take effect, any changes require re-opening opened database windows.

Functions

mathematical functions
statistical functions
financial functions
date/time functions
text functions
logical functions

Mathematical functions

ABS
AND
BJ0
BJ1
BJN
BY0
BY1
BYN
CEIL
COMBNR
COMBNS
DEGREES
EVEN
EXP
EXPE
EXPM
FACT
GCD
LOG
LOG10
LOGX
MOD
NOT
ODD
OR
PI
POW10
RADIANS
RAND
ROUND
ROUNDE
ROUNDO
ROUNDX
SCM
SGN
SHIFTL
SHIFTR
SQR
SQRT
TRUNC
XOR

Hyperbolic functions

ARCOSH
ARSINH
ARTGH
COSH
COSECH
SECH
SINH
TANH

Trygonometrical functions

ACOS
ASIN
ATAN
COS
SIN
TAN

Statistical functions

CHI2
STD
ERF
ERFV

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

Logical functions

IF
ISERROR
ISFILE
ISNUMBER
ISOPEN
ISSTRING
TEST

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-Base. In GS-Base 3.5 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-Base. In GS-Base 3.5 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

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

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

RAND(Number)

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

Example: RAND(60) = 34.56

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-based 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-Base. In GS-Base 3.5 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-Base. In GS-Base 3.5 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*. The integer portion represents the number of days from 30 December 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 *Hour, Minute and Second*. The integer portion represents the number of days from 30 December 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 formatted as *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 formatted as *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 number +/- *DaysNumber*.

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

NOW(HoursNumber)

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

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

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-Base. In GS-Base 3.5 and later use respectively the "IF", "UPPER", "LOWER", "EVEN", "ODD" functions.

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((number>1)*(number<10); 1; 2) = 1

ISERROR(Value)

Returns 1 (True) if *Value* represents the *error* value.

Example: ISERROR(1/0) = 1

ISNUMBER(Value)

Returns 1 (True) if *Value* represents a number. Otherwise 0 (False) is returned.

Example: ISNUMBER(12) = 1

ISSTRING(Value)

Returns 1 (True) if *Value* represents a text string. 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:\gsbase\sample1.gsb") = 1

ISOPEN(Text)

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

Example: ISOPEN("d:\gsbase47\sample1.gsb") = 1

ProgID

GS-Base.Document.5

Remarks

Use this ProgID to create a new automation object. For example, if you're using VC++ and smart pointers, you can use the following code:

```
IGSBasePtr pDocument;

pDocument.CreateInstance(L"GS-Base.Document.5");
```

or

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

After creating a new object you must:
(1) create a new database structure using the AddField / RemoveField methods and save it with the CreateDatabase method
or
(2) load an existing database using the Load, LoadDatabase or LoadTextFile methods.

Note: The main application window must be invisible (or GS-Base should not be running) before constructing a new object. Otherwise, the standard "Database" dialog box is displayed and you have to define the initial database structure manually.

short AddField(_bstr_t szName**, short** nType**, short** nLength**)**
short RemoveField(short nFieldIndex**)**

Return Value

The number of fields in the record if the field was added, otherwise 0.

Parameters

szName
The name of the added field.
nType
The field type. This can be a numeric value or one of the predefined constants:

```
enum FieldTypes
{
 ftText = 1,
 ftNumeric,
 ftDateTime,
 ftBinary
}
```

nLength
The field length of a text field (from 1 to 256).

Remarks

Call these methods to add or remove fields from the database structure. You can define up to 512 fields. 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).

short CreateDatabase(_bstr_t szFilePath**)**

Return Value

1 if the database was successfully created and saved, otherwise 0.

Parameters

szFilePath
The full path of the saved file. If the file already exists, it's overwritten.

Remarks

Call this method to create a new database after defining its structure. 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).

short GetFieldIndex(_bstr_t szFieldName**)**

Return Value

The 1-based index of the field if it exists, otherwise 0.

Parameters

szFieldName
The name of the field.

Remarks

Call this method to obtain the field position within a record. This is a number from the range <1, number of fields>.

HRESULT SetFieldName(short nType**, short** nFieldIndex**, _bstr_t** szFieldName**)**
_bstr_t GetFieldName(short nType**, short** nFieldIndex**)**

Return Value

The "Get" method returns the actual name of the field if nType is 1 or the name displayed in the form or table view.

Parameters

nType
If it's 0, the actual field name (used within the record) is set / returned. If it's 1, this is text displayed in the form or table view for that field.
nFieldIndex
The field index.
szFieldName
The name of the field.

Remarks

Call these methods to get or set a new field name.

short SetFieldType(short nFieldIndex**, short** nFieldType**)**
short GetFieldType(short nFieldIndex**)**

Return Value

The "Get" method returns the field type for the specified field index. The available field types are:

```
enum FieldTypes
{
    ftText = 1,
    ftNumeric,
    ftDateTime,
    ftBinary
}
```

Parameters

nFieldIndex
The field index.
nType
The new field type.

Remarks

Call these methods to get or set field types. When changing field types in an existing database that contains some records, GS-Base performs all necessary conversion of the field contents.

short SetFieldLength(short nFieldIndex**, short** nFieldLength**)**
short GetFieldLength(short nFieldIndex**)**

Return Value

The "Get" method returns the field length for the specified field index. The "Set" method returns 1 if the length was changed, otherwise 0.

Parameters

nFieldIndex
The field index.
nLength
The new field length (<1, 256>).

Remarks

Call these methods to get or set field lengths. Note that this value can be changed only for fields of the "Text" type.

HRESULT SetDefaultValue(short nFieldIndex**, _bstr_t** szValue**)**
_bstr_t GetDefaultValue(short nFieldIndex**)**

Return Value

The "Get" method returns the default value for the specified field index.

Parameters

nFieldIndex
The field index.
szValue
The new default value (up to 128 characters).

Remarks

Call these methods to get or set the default value for the specified field.

short GetFieldCount( )

Return Value

The number of fields in the record.

Remarks

Call this methods to obtain the number of fields in the record.

HRESULT ExcludeField(short nFieldIndex**)**
HRESULT IncludeField(short nFieldIndex**)**

Parameters

nFieldIndex
The field index.

Remarks

Call these methods to specify which fields should be exported when saving Html and text files. By default, all fields are exported.

VARIANT_BOOL Load(_bstr_t szFilePath**)**

Return Value

VARIANT_TRUE if the database was successfully opened, otherwise VARIANT_FALSE.

Parameters

szFilePath
The full path of the GS-Base database to open.

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).

VARIANT_BOOL LoadDatabase(_bstr_t szTargetFile**, _bstr_t** szSourceFile**, _bstr_t** szTable**, short** nType**, _bstr_t** szFilter**)**

Return Value

VARIANT_TRUE if the database was successfully opened and imported, otherwise VARIANT_FALSE.

Parameters

szTargetFile
The full path of the database that will contain the imported data. If the file already exists, it's overwritten.
szSourceFile
The full path of the database to import.
szTable
The name of the table to open. This paramenter is used only for Access databases.
nType
The database type. You can use simple numeric values or constants defined in the type library:

```
enum DatabaseFormats
{
 dbExcel97 = 1,
 dbExcel2000,
 dBaseIII,
 dBaseIV,
 dbFoxPro,
 dbClipper,
 dbAccess2000
}
```

To import a GS-Calc file, use 0 as the *nType*.
szFilter
Specifies the filter expression. This may be an expression conforming to the SQL WHERE clause syntax. For example:

```
City>="Ne" AND Number=12
City<= "W" OR Number>10
(April_Sales+May_Sales+June_Sales)/3>50000
Title+Name="Mr.Smith" OR Title+Name="Mr.Jones"
Name LIKE "*son"
```

Square brackets must be used if the used field name contains spaces or some special characters (e.g. "[Product number]"). This paramenter is used only for Access databases.

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).

VARIANT_BOOL LoadTextFile(_bstr_t szTargetFile**, _bstr_t** szSourceFile**, _bstr_t** szSeparator**, short** nCharSet**, short** nNamesInFirstRow**)**

Return Value

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

Parameters

szTargetFile
The full path of the database that will contain the imported data. If the file already exists, it's overwritten.
szSourceFile
The full path of the text file to import.
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.
nNamesInFirstRow
1if the first row of the imported text file contains values that should be used as fields names.

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).

VARIANT_BOOL Save( )

Return Value

VARIANT_TRUE if the database was successfully saved; otherwise VARIANT_FALSE.

Remarks

Call this method to save the current document. Note that all changes are save automatically when you scroll the current record or perform any action (like sorting, searching, printing, exporting etc.) which changes the current record position. Thus, the "unsaved" state means that there is either exactly one modified record or some modified setting (filters, page settings, column widths, row heights, formats etc.).

VARIANT_BOOL SaveCopyAs(_bstr_t szFileName**)**

Return Value

VARIANT_TRUE if the database was successfully saved; otherwise VARIANT_FALSE.

Parameters

szFileName
The full path of the saved file. If the database already exists, it's overwritten.

Remarks

Call this method to save the current record set to a new database.

VARIANT_BOOL SaveCopyAsDatabase(_bstr_t szFileName**, short** nType**, short** nUseRTF**)**

Return Value

VARIANT_TRUE if the database was successfully saved; otherwise VARIANT_FALSE.

Parameters

szFileName
The full path of the saved file. If the database already exists, it's overwritten.
nType
The database type. You can use simple numeric values or constants defined in the type library:

```
enum DatabaseFormats
{
 dbExcel97 = 1,
 dbExcel2000,
 dBaseIII,
 dBaseIV,
 dbFoxPro,
 dbClipper,
 dbAccess2000
}
```

To save a GS-Calc file, use 0 as the *nType*.
nUseRTF
1 if binary field contents should be saved along with formatting. If it's 0, plain text will be saved.

Remarks

Call this method to export the current record set to a new database.

VARIANT_BOOL SaveCopyAsText(_bstr_t szFileName**, _bstr_t** szSeparator**, short** nOemCharSet**, short** nNamesInFirstRow**)**

Return Value

VARIANT_TRUE if the database was successfully saved; otherwise VARIANT_FALSE.

Parameters

szFileName
The full path of the saved file. If the database already exists, it's overwritten.
szSeparator
Specifies the field separator to be used in the created text file.
nOemCharSet
1 if the program should perform conversion to OEM character set. 0 if no conversion is necessary.
nNamesInFirstRow
1 if the first row of the exported text file should contain field names. 0 if this information should not be saved.

Remarks

Call this method to export the current record set to a new text file.

VARIANT_BOOL SaveCopyAsHtml(_bstr_t szFileName**)**

Return Value

VARIANT_TRUE if the database was successfully saved; otherwise VARIANT_FALSE.

Parameters

szFileName
The full path of the saved file. If the database already exists, it's overwritten.

Remarks

Call this method to export the current record set to a new html file or files. You can change the default conversion settings modifying any of the following properties:

short HtmlSaveForms
1 if records should be saved in forms, otherwise 0.

_bstr_t HtmlDocTitle
The title of the Html document. Up to 99 characters.

short HtmlCharSet
The character set used it the document. This is a 0-based index having of the following values: 0 - Baltic (ISO), 1- Baltic (Windows), 2 - Central European (ISO), 3 - Central European (Windows), 4 - Cyrillic (ISO), 5 - Cyrillic (KOI8-R), 6 - Cyrillic (Windows), 7 - Greek (ISO), 8 - Greek (Windows), 9 - Latin 3 (ISO), 10 - Turkish, 12 - Western European.

long HtmlDocBackground
The color of the document background specified as an RGB value.

_bstr_t HtmlTableCaption
The table caption. Up to 99 characters.

long HtmlTableBackground
The color of the table background specified as an RGB value.

long HtmlBorderColor
The color of the table frame specified as an RGB value.

short HtmlBorderSize
The thickness of the table frame in pixels.

short HtmlSupressDuplicates
1 if duplicated field values should not be saved (see: Converting databases to Html). Otherwise 0.

_bstr_t HtmlDuplicateText
Specifies the text used in place of duplicated field values if the HtmlSupressDuplicates is set to 1. Up to 99 characters.

short HtmlSaveRecordNumbers
1 if record numbers should be saved in the first column of the table. Otherwise 0.

short HtmlMultiplePages
1 if records should be saved to more than one page with *HtmlRecordsPerPage* records per page, otherwise 0.

short HtmlRecordsPerPage
Specifies the maximum number of record per page. If the total number of saved record is greater, multiple linked documents are saved.

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-Base 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.

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" (see: Save. Note that most operations is performed much faster if the document and app windows stay invisible.

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.

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 database or to remove an existing list.

long AddNewRecords(short nBeforeCurrent**, long** nNumber**)**

Return Value

The number of inserted records if the operation was successful, otherwise 0.

Parameters

nBeforeCurrent

0 if the new records should be appended at the end of the current record set. 1 if they should be inserted at the current position - the current and subsequent records will be moved down by *nNumber* positions.

nNumber

The number of records to inserts.

Remarks

Call this method to add new records to an existing database. The first of the inserted records becomes the current record. 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 CompactDatabase( )

Remarks

Call this method to remove unused space from file containing binary fields.

HRESULT Delete(short nFieldIndex**)**

Parameters

nFieldIndex
Specifies the index of the field whose contents should be deleted.

Remarks

Call this method to delete the contents of the specified field in the current record.

_bstr_t GetBinaryText(short nFieldIndex, **short** nUseRTF**)**

Return Value

The text representing the contents of the specified binary field. NULL if the field contains no data or ole objects.

Parameters

nFieldIndex
Specifies the field index of the current record (<1, number of fields>).
uUseRTF
1 if the returned text should contain RTF formatting. 0 if plain text should be returned.

Remarks

Call this method to get the contents of a binary field that contains text data.

VARIANT_BOOL PutNumber(short *nFieldIndex***, double** dNumber**)**
double GetNumber(short *nFieldIndex***)**

Return Value

The "Put" method returns VARIANT_TRUE if the specified number was inserted into the given field; otherwise VARIANT_FALSE. The "Get" methods return the contents of the specified field.

Parameters

nFieldIndex
Specifies the field index of the current record (<1, number of fields>).
dNumber
Specifies the inserted number.

Remarks

Call these methods to insert a number into the specified field or to get the contents of the field.
To get the text contents of a binary field, use the GetBinaryText method.

VARIANT_BOOL PutText(short *nFieldIndex***, _bstr_t** szText**)**
_bstr_t GetText(short *nFieldIndex***)**

Return Value

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

Parameters

nFieldIndex
Specifies the field index of the current record (<1, number of fields>).
szText
Specifies the inserted text.

Remarks

Call these methods to insert text into the specified field or to get the contents of that field. When used for a binary field, the *szText* parameter of the **PutText** method may contain text along with RTF formatting.
To get the text contents of a binary field, use the GetBinaryText method.

long RemoveRecords(short nSelected**, long** nStartRecord**, long** nEndRecord**)**

Return Value

1 if the specified records were successfully removed; otherwise 0.

Parameters

nSelected
: 1 if only the selected records from the current record set should be removed. 0 if the records from specified range of the current set should be removed.

nStartRecord, nEndRecord
: Specify the record numbers (from 1 to the number of records in the current record set).

Remarks

Call this method to remove records from the database. Note that the record numbers specify their position within the current record set, not their absolute indexes within the entire database.

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

Return Value

The GetDocumentState returns 1 if the database 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 database. The Hide andSave methods set the document state to "not modified". Note that all changes are save automatically when you scroll the current record or perform any action (like sorting, searching, printing, exporting etc.) which changes the current record position. The "modified" state means that there is either exactly one modified record or some modified setting (filters, page settings, column widths, row heights, formats etc.).

HRESULT SetColumnWidth(short nFieldIndex**, short** nWidth**)**
short GetColumnWidth(short nFieldIndex**)**

Return Value

The GetColumnWidth method returns the width of the specified table column (in pixels).

Parameters

nFieldIndex
The field index.
nWidth
The new width (in pixels) of the column.
Remarks

Call these methods to set or get a column width for the given field.

HRESULT SetRowHeight(short nHeight**)**
short GetRowHeight()

Return Value

The GetRowHeight method returns the height of the table row (in pixels).

Parameters

nFieldIndex
The field index.
nHeight
The new height (in pixels) of the rows.
Remarks

Call these methods to set or get the table row height.

HRESULT SetFormat(short nFieldIndex**, short** nType**)**
short GetFormat(short nFieldIndex**, short** nType**)**
HRESULT RemoveFormat(short nFieldIndex**, short** nType**)**

Return Value

The "Get" method returns 1 if the specified field has the requested format; otherwise 0.

Parameters

nFieldIndex
Specifies the field index (<1, number of fields>).
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
{
 ftCurrency = 1,          // currency
 ftSeparators, // separators in numbers
 ftPercent,               // percent
 ftNormal,                // default number style
 fBin,// binary
 fHex,           // hexadecimal
 ftDefinedStyle,          // custom number style
 ftLeft,                  // align left
 ftHCenter,     // centered horizontally
 ftRight,                 // align right
 ftBold,                  // bold
 ftItalic,                // italic
 ftStrikeOut,   // strikeout
 ftUnderline,   // underline
 ftNormalFont,            // default font format
 ftFontType,    // font name
 ftBackground,            // background color
 ftHidden,                // hidden cells
 ftMultiline,             // text displayed in multiple lines
 ftDropDownList,          // drop down list
 ftFixedDecimals,         // fixed number of decimal places
}
```

Remarks

Call these methods to set, get or remove various formats for the specified field. 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:

ftDefinedStyle

```
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).
```

ftFontType

```
BSTR FontName        // The font name.
short FontSize       // The font size.
long FontColor       // The font color specified as a RGB value.
```

ftBackground

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

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

short SetSelectionFlag(short nSelection**)**
short GetSelectionFlag( )

Return Value

1 if the current record is marked as "selected", otherwise 0.

Parameters

nSelection
Specifies whether the current record should be selected (1) or de-selected (0).

Remarks

Call these methods to select or remove selection for the current record.

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.

long Bottom( )

Return Value

The current record number.

Remarks

Call this method to scroll to the last record in the current record set.

long GetRecordCount(short nCurrentSet**)**

Return Value

The number of records in the current record set if nCurrentSet is 1 or the total number of record in file if nCurrentSet is 0.

Remarks

Call this method to get the total number of records in the current record set or in the entire database.

long LookUp( )

Return Value

The number of found records.

Remarks

Call this method to search the database using the current filter. The first record in the new record set becomes the current record.

long LookUpComplement( )

Return Value

The number of found records.

Remarks

Call this method to find records that are a complement of the current record set. The first record in the new record set becomes the current record.

long LookUpDuplicates(short nFieldIndex**)**

Return Value

The number of found records.

Parameters

nFieldIndex

If it specifies a valid field Index, GS-Base will search for duplicated values of that field. If it's 0, the program will search for duplicates records.

Remarks

Call this method to find duplicated records or records with duplicated values of the specified field. The first record in the new record set becomes the current record.

long LookUpSelected( )

Return Value

The number of found records.

Remarks

Call this method to find selected duplicated records. The first record in the new record set becomes the current record.

long Next( )

Return Value

The current record number if the operation was successful, otherwise 0.

Remarks

Call this method to scroll to the next record in the current record set.

long Previous( )

Return Value

The current record number if the operation was successful, otherwise 0.

Remarks

Call this method to scroll to the previous record in the current record set.

HRESULT RemoveAllFilters( )

Remarks

Call this method to remove all filters. Note that you have to call the LookUp method to update the record set.

short SetCurrentField(short nFieldIndex**)**
short GetCurrentField( )

Return Value

If the "Set" method was successful, it returns the index of the current field, otherwise 0. The "Get" method returns the index of the current field.

Parameters

nFieldIndex
The field index.

Remarks

Call these methods to find duplicated records or records with duplicated values of the specified field.

HRESULT SetFilter(short nFieldIndex**, _bstr_t** szFilter**)**
_bstr_t GetFilter(short nFieldIndex**)**

Return Value

The "Get" method returns the filter of the specified filter.

Parameters

nFieldIndex
The field index.
szFilter
The expression that will be used as a filter (up to 64 characters).

Remarks

Call these methods to obtain or set the filter expression for the specified field.

long SetCurrentRecord(long nRecordNumber**)**
long GetCurrentRecord( )

Return Value

The number of the current record.

Parameters

nFieldIndex
The field index.
szFilter
The expression that will be used as a filter (up to 64 characters).

Remarks

Call these methods to get or set the filter expression for the specified field.

short Sort(short nDescending**, short** nReverseWords**, short** nFieldIndex1**, short** nFieldIndex2**, short** nFieldIndex3**)**

Parameters

nDescending
Determines the sort order: 1- descending sorting, 0 - 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 field indexes of the subsequent sort keys. If the second and/or third key don't exist, 0 should be used.

Remarks

Call these method to sort the current record set.

long Top( )

Return Value

The current record number if the operation was successful, otherwise 0.

Remarks

Call this method to scroll to the first record in the current record set.

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.

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.

Creating databases, editing and formatting fields (COM Automation samples)

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

#import "gsbase.tlb"

using namespace GSBaseLib;

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

void main()
{
	IGSBasePtr pDocument;

	pDocument.CreateInstance(L"GS-Base.Document.5");
	// or pDocument.CreateInstance(__uuidof(Document));

	pDocument->AddField("Product", ftText, 40);
	pDocument->AddField("Category", ftText, 40);
	pDocument->AddField("UnitPrice", ftNumeric, 8);
	pDocument->AddField("AvailableOn", ftDateTime, 8);
	pDocument->AddField("Description", ftBinary, 8);

	pDocument->CreateDatabase("d:\\samples\\products.gsb");

	pDocument->AddNewRecords(0, 3);

	pDocument->PutText(1, "Northwoods Cranberry Sauce");
	pDocument->PutText(2, "Condiments");
	pDocument->PutNumber(3, 40);
	pDocument->PutText(4, "10/10/1999");

	pDocument->Next();

	pDocument->PutText(1, "Teatime Chocolate Biscuits");
	pDocument->PutText(2, "Confections");
	pDocument->PutNumber(3, 9.2);
	pDocument->PutText(4, "10/10/1999");

	pDocument->FontName = "Arial";
	pDocument->FontSize = 13;
	pDocument->FontColor = RGB(0, 128, 0);
	pDocument->SetFormat(1, ftFontType);
	pDocument->SetFormat(1, ftBold);

	pDocument->AddDropDownList("Category", "Confections\nCondiments\nBeverages\nDairy
Products\nGrains/Cereals");
	pDocument->InputListName = "Category";
	pDocument->UpdateInputList = VARIANT_TRUE;
	pDocument->SetFormat(2, ftDropDownList);
```

```
	pDocument->SetFormat(3, ftCurrency);

	pDocument->Top();
	do
	{
		_bstr_t str = pDocument->GetText(1);
		_tprintf(_T("\nField 1: %s, Field 3: %f"), (TCHAR *)str, pDocument->GetNumber(3));
	} while ( pDocument->Next() );
}
```

Searching and sorting (COM Automation samples)

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

#import "gsbase.tlb"

using namespace GSBaseLib;

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

void main()
{
	IGSBasePtr pDocument;

	pDocument.CreateInstance(L"GS-Base.Document.5");

	pDocument->Load("d:\\samples\\testdata.gsb");

	pDocument->RemoveAllFilters();
	pDocument->LookUp();

	short nName = pDocument->GetFieldIndex("Name");
	short nCompany = pDocument->GetFieldIndex("Company");

	pDocument->Sort(0, 0, nName, nCompany, 0);

	pDocument->SaveCopyAs("d:\\samples\\sorted.gsb");

	pDocument->SetFilter(1, "<A;M>");
	pDocument->LookUp();

	_tprintf(_T("\n%d record(s) found.\n"), pDocument->GetRecordCount(1));

	pDocument->SaveCopyAsDatabase("d:\\samples\\subset.dbf", dBaseIV, 0);
}
```

Saving HTML files (COM Automation samples)

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

#import "gsbase.tlb"

using namespace GSBaseLib;

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

void main()
{
	IGSBasePtr pDocument;

	pDocument.CreateInstance(L"GS-Base.Document.5");

	pDocument->LoadDatabase("d:\\samples\\testdata.gsb", "d:\\samples\\info.mdb", "Products", dbAccess2000,
"");

	pDocument->HtmlMultiplePages = VARIANT_TRUE;
	pDocument->HtmlRecordsPerPage = 100;
	pDocument->HtmlTableBackground = RGB(255,255,128);
	pDocument->HtmlDocBackground = RGB(255,255,0);
	pDocument->HtmlDocTitle = "Sample document";
	pDocument->HtmlTableCaption = "Products";
	pDocument->ExcludeField(pDocument->GetFieldIndex("UnitPrice"));
	pDocument->Save();

	pDocument->SaveCopyAsHtml("d:\\samples\\products.htm");
}
```

Print command (File menu)

Use this command to print a document. This command presents the **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.

Note: The **Print Selection** option is active only when printing the predefined single-record form and means printing the current record. This command always refers to the currently found records.

Print dialog box

See: Print command.

Print Progress Dialog

The Printing dialog box is shown during the time that GS-Base 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.