Getting To The IDE

Menu Tools > Macros > Organize macros > LibreOffice Basic. Select desired module to edit (Edit button).

Overview

The IDE window is made of 10 tools, most of them are described in this refcard.

- Title bar
- Menu Bar: Standard LibreOffice menus, dedicated to macros management.
- Standard toolbar: The LibreOffice Standard toolbar, showing only tools dedicated to code edition and debugging.
- Macros toolbar: Tools dedicated to code edition and debugging.
- Object catalog: Containers tree-view showing libraries, modules and subprograms.
- Code editor: Main area for code edition, with syntax highlighting, breakpoints management and execution control.
- Watch pane: Variable contents checking (watches).
- Call stack panel: Subprogram calls and their parameters.
- Tab bar: Lists and manages the current library modules.
- Status bar: Run-time status.

The main panels (Catalog, Editor, Watches and Call stack) can be detached, swaps from a tool to the next one.

Toolsbars

If not visible: View > Toolbars.

Standard Toolbar

Two toolbuttons are of interest: Select macro and Macros. Selects the watch to remove, (in the Watches panel).

Libraries

- Library selector
- « Compile »
- Run immediately
- Stop
- Stop Over
- Step into
- Step out
- Breakpoint On/off
- Manage Breakpoints
- Enable Watch
- Find Parenthesis
- Import Basic
- Export Basic
- Import Dialog
- Check the gutter: A margin in which the breakpoints are displayed.

Macros Toolbar

- Selection of the library to use.
- Syntax checking.
- Run the subprogram under the cursor.
- Stop the running program.
- Stop step by step execution, bypassing the called.
- Step into the called subprogram.
- Step out to the caller.
- (De)activate the breakpoint on current line.
- Opens the breakpoints dialog.
- Adds the selected var to the watches.

Dialog Toolbar

- Import macro: Calls the Macros selection dialog.
- Calls the Macros Organizer dialog.
- Import dialog source code.

Useful Items in the Edition Panel

- Gutter: Line numbers.
- Editor: Syntax highlighting.
- (Un)Indenting lines

Watches Panel

- Allows checking variable contents at runtime (more in Debugging below).
- Variable name.
- Value.
- Type.

Adding A Watch

1. In the code, select the variable to observe (you may enter its name in the Watch zone).
2. Click the Enable Watch button (F5, F7).

Removing A Watch

1. Select the watch to remove.
2. Click Remove Watch (in the Watches panel).

Call Stack Panel

- In step-by-step execution mode, allows to know the successive subprograms calls.
- Read it bottom-up. All subprograms are listed along with their parameter values.
- Subprogram in is the one in which the execution is currently. The others are numbered subsequently.

LibreOffice RefCard
LibreOffice Basic
IDE
v. 1.05 – 01/15/2019

Beginner

Written using LibreOffice v. 5.3.3 – Platform: All

IDE: Integrated development environment.
A set of tools for writing and debugging your macros code.

LibreOffice Basic
LibreOffice Basic
IDE
v. 1.05 – 01/15/2019

Beginner

Written using LibreOffice v. 5.3.3 – Platform: All

IDE: Integrated development environment.
A set of tools for writing and debugging your macros code.

Containers

- My Macros & Dialogs

Specific to the user’s account, for all documents. Can only be used by the user.

LibreOffice Macros & Dialogs

(aka global macros)

These macros are stored within the LibreOffice global container. As such, they can be viewed and used everywhere.

The ones on the capture are part of a standard LibreOffice install.

Untitled 1 (in the example)

In the current document.

Libraries

- Loaded (colored) or not (grayed-out).
- Standard Library

Apart from global libraries, each container comes with a Standard library.

- Standard is always loaded at opening time (of the application or the document):

  - It can’t be deleted.
  - It can’t be overwritten through some code import.

- Modules

Subprograms (macros)

- Code Editor

Used for code writing (with syntax highlighting) and debugging (read on).

Object Catalog

Displays the 3 containers types and their contents: libraries, modules and macros.

LibreOffice Basic
LibreOffice Basic
IDE
v. 1.05 – 01/15/2019

Beginner

Written using LibreOffice v. 5.3.3 – Platform: All

IDE: Integrated development environment.
A set of tools for writing and debugging your macros code.

Getting To The IDE

Menu Tools > Macros > Organize macros > LibreOffice Basic. Select desired module to edit (Edit button).

Overview

The IDE window is made of 10 tools, most of them are described in this refcard.

- Title bar
- Menu Bar: Standard LibreOffice menus, dedicated to macros management.
- Standard toolbar
- Macros toolbar: Tools dedicated to code edition and debugging.
- Object catalog: Containers tree-view showing libraries, modules and subprograms.
- Code editor: Main area for code edition, with syntax highlighting, breakpoints management and execution control.
- Watch pane:Variable contents checking (watches).
- Call stack panel: Subprogram calls and their parameters.
- Tab bar: Lists and manages the current library modules.
- Status bar: Run-time status.

The main panels (Catalog, Editor, Watches and Call stack) can be detached, swaps from a tool to the next one.

Toolsbars

If not visible: View > Toolbars.

Standard Toolbar

Two toolbuttons are of interest: Select macro and Macros. Selects the watch to remove, (in the Watches panel).

Libraries

- Library selector
- « Compile »
- Run immediately
- Stop
- Stop Over
- Step into
- Step out
- Breakpoint On/off
- Manage Breakpoints
- Enable Watch
- Find Parenthesis
- Import Basic
- Export Basic
- Import Dialog
- Check the gutter: A margin in which the breakpoints are displayed.

Macros Toolbar

- Selection of the library to use.
- Syntax checking.
- Run the subprogram under the cursor.
- Stop the running program.
- Stop step by step execution, bypassing the called.
- Step into the called subprogram.
- Step out to the caller.
- (De)activate the breakpoint on current line.
- Opens the breakpoints dialog.
- Adds the selected var to the watches.

Dialog Toolbar

- Import macro: Calls the Macros selection dialog.
- Calls the Macros Organizer dialog.
- Import dialog source code.

Useful Items in the Edition Panel

- Gutter: Line numbers.
- Editor: Syntax highlighting.
- (Un)Indenting lines

Watches Panel

- Allows checking variable contents at runtime (more in Debugging below).
- Variable name.
- Value.
- Type.

Adding A Watch

1. In the code, select the variable to observe (you may enter its name in the Watch zone).
2. Click the Enable Watch button (F5, F7).

Removing A Watch

1. Select the watch to remove.
2. Click Remove Watch (in the Watches panel).

Call Stack Panel

- In step-by-step execution mode, allows to know the successive subprograms calls.
- Read it bottom-up. All subprograms are listed along with their parameter values.
- Subprogram in is the one in which the execution is currently. The others are numbered subsequently.
Tab Bar
Displays the modules names for the current library. Modules names are shown in alphabetical order, left to right.

Right-click on the bar to manage the modules:
- Insert > Basic Module: Create a new code module (defaults to Module(N)).
- Insert > Basic Dialog: Create a new dialog module (defaults to Dialog(N)).
- Delete: Delete the selected module.
- Rename: Rename the selected module.
- Hide: Temporarily hide the selected module.
- Modules: Calls the Macro Organizer.

Status Bar
From left to right:
- The subprogram full name in which the cursor currently sits.
- The line (Le) and column (Col) (character) number where the cursor currently is.

Debugging
Debugging: checking a program execution.
☞ This operation is realized executing the program part to be checked in step-by-step mode while watching the execution context (variables values).

Debugging uses three panels: code editor ☞, call stack ☞ and watch pane ☞. It runs under the control of the three step-by-step buttons (Macros toolbar ➊) and breakpoints.

Execution Modes
Five buttons define the way the code is run. The ☞ mark displays the execution point.
☞ Note that when you start execution (any mode) while the cursor is in a subprogram that requires non-optional parameters, you get an error message: Argument is not optional.

- Run: F5 / Ctrl+F5 / Maj+F5
  - Runs the subprogram where the cursor sits. The execution stops at breakpoints only.
  - Stops the execution.
  - Step-by-step execution without entering the subprograms.
  - Step in subprograms.
  - Step out to callers.

Managing Breakpoints
Breakpoint: a mark where the execution is paused. This allows to examine the execution context at that moment (watch values).

Execution is paused before executing the marked line.
☞ The actions below may be accomplished during a debugging session.

Adding A Breakpoint
Double-click in the gutter on the line to stop at when no breakpoint exists (F9).

Deleting A Breakpoint
Double-click in the gutter on an existing breakpoint (F9).

Disabling/Re-enabling A Breakpoint
Right-click in the gutter on any existing breakpoint, then disable/ active. The icon turns from (active) to (inactive).

Breakpoints Manager
Call it by:
- right-clicking the execution mark or any existing breakpoint, option Properties
- clicking the Manage Breakpoints button ☞. Top to bottom:
  - (list and edition area)
    - Line numbers where breakpoints are set.
    - Select or create a new one.
  - (Un)Check to (de)activate the selected breakpoint.
  - Active
  - Pass Count
  - New
  - Delete
  - Deletes the selected breakpoint.

Options
LibreOffice offers a set of IDE options in Tools > Options > LibreOffice > Basic IDE
☞ These options are available after having checked the Enable experimental features (needed to get code completion) option.

Enable code completion
- The IDE automatically completes the Basic objects methods.
- Doesn't apply to custom objects.
- The UseExtended types option must be enabled.

Auto correction
- Fixes keywords and variable names syntax.

Auto close quotes
- When typing " (double quote), the IDE adds the closing quote.

Auto close parenthesis
- When typing ( (parenthesis), the IDE adds the closing parenthesis.

Auto close procedures
- When typing Sub Xxx or Function Xxx (Sub of Function declaration), the IDE adds an End Sub of End Function.

UseExtended types
- Allows using UNO object types as valid for Basic.

Hints
Copying A Library Between Containers
1. Open the source document/container.
2. Open the Macro Organizer (oo), Libraries tab.
3. Select the library Location.
4. Export > Export as Basic library.
5. Open the target document/container.
6. Import.

Copying Modules From A Library To Another
(in the same document or between documents/containers)
1. Open both source and target documents/containers.
2. Open the Macro Organizer (oo).
3. Drag/drop from source to target.
☞ By default, modules are moved. To copy: Ctrl + drag/drop.

Hiding Modules
Allows to (temporarily) simplify the tab row: right-click a tab then > Hide.

Encrypting Libraries
Encrypting a library makes a more secure code.
☞ Any library may be encrypted, except Standard.
1. Open the Macro Organizer (oo), Libraries tab.
2. Select the Location.
3. Select the library.
4. Click Password.
5. Enter the New password and confirm.
6. Validate.
☞ Warning: The encryption is very strong (AES): password loss is irretrievable!
☞ Always keep a non-encrypted version for yourself.

Cleaning MSOffice imports
Delete the unnecessary libraries and modules that came with the import.
☞ Always work on a copy of your file!
1. Save to the "flat" format (ex: .fods).
2. Open the "flat" file using a XML-capable text editor.
4. Delete the whole section between (including) the <ooo:library-embedded oo:format="Standard"> and </ooo:library-embedded> marks.
5. Save.
6. Open with LiLo and re-create the correct libraries/modules tree (copy/paste the wanted code from the original document to the current one).

IDE Keyboard Shortcuts

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F11</td>
<td>Step Out</td>
</tr>
<tr>
<td>F8</td>
<td>Step Into</td>
</tr>
<tr>
<td>F7</td>
<td>Step Over</td>
</tr>
<tr>
<td>F8</td>
<td>Run</td>
</tr>
<tr>
<td>Ctrl+F8</td>
<td>Stop</td>
</tr>
<tr>
<td>Maj+F8</td>
<td>Step Out</td>
</tr>
<tr>
<td>Ctrl+F5</td>
<td>Stop</td>
</tr>
<tr>
<td>Maj+F5</td>
<td>Step In</td>
</tr>
<tr>
<td>F5</td>
<td>Step Over</td>
</tr>
</tbody>
</table>

Author: Jean-François Nifenecker – jean-francois.nifenecker@laposte.net

License
This refcard is placed under the CreativeCommons BY-SA v4 (intl) license. More information: https://creativecommons.org/licenses/by-sa/4.0/