

B

Enabling and disabling ReportSmith features

This appendix is intended primarily for those users who:

- Are incorporating ReportSmith for use with other applications.
- Need to customize ReportSmith to implement security features, such as disabled menu commands and toolbar buttons.

Customizing the toolbar

The ReportSmith toolbar is designed for maximum effectiveness and usability. However, from time to time, you probably find that certain toolbar buttons are not needed within a certain report. For example, you might want to prevent report users from editing the SQL statements used to create the report.

ReportSmith enables you to disable the appropriate toolbar button (and its menu-command equivalent) in such a case by using a single dialog box and checking those toolbar or ribbon buttons you want disabled. Once these choices have been put into effect, you can distribute the report without the macro that enables and disables buttons to prevent users from changing your settings.

You can also get an “insider’s look” at the underlying ReportBasic code used to create and implement this dialog box. Then, modify that code as necessary to suit your particular reporting and development needs.

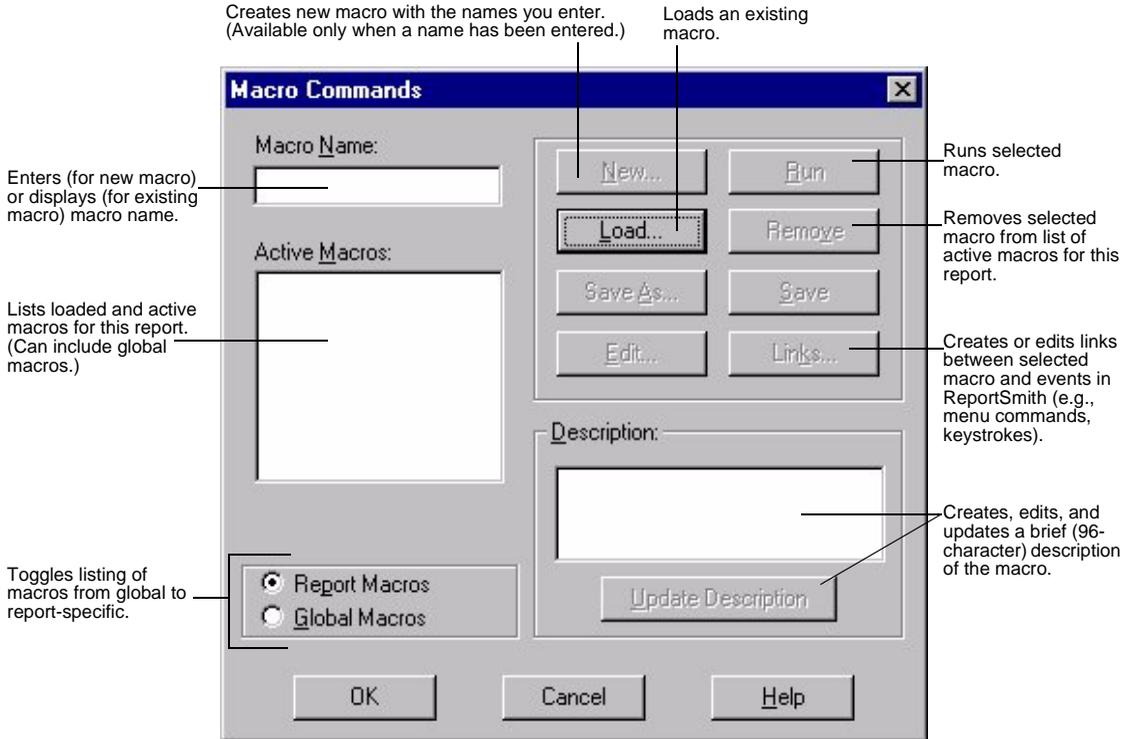
Note: Remember that while every toolbar and ribbon button is represented in the dialog box discussed below, not every menu command is likewise represented. This is because many menu commands do not have toolbar or ribbon equivalents. Nonetheless, you’ll see how quickly and easily you can also disable, or even remove, ReportSmith menu commands, too.

The sections that follow are not intended to present a complete discussion of all features and functionality of the Macro Commands dialog box. For a detailed discussion of macros, their use and their construction, see the ReportSmith “Macro guide” on page 195.

- To access the enable/disable dialog box,
 - 1 Choose Tools|Macro.

The Macro Commands dialog box appears, as in Figure B.1.

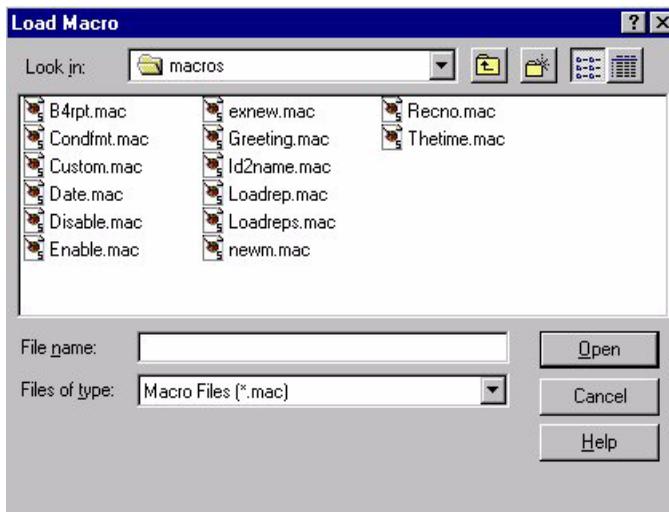
Figure B.1: The Macro Command dialog box



2 Select Report Macros and then choose Load.

The Load Macro dialog box appears, as in Figure B.2.

Figure B.2: The Load Macro dialog box

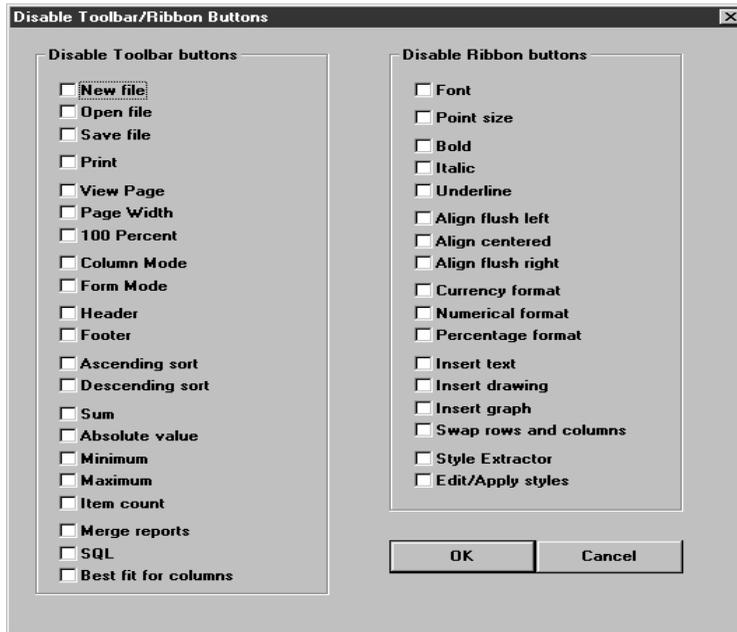


3 Select CUSTOM.MAC from the list of macro files, and then choose OK to return to the Macro Commands dialog box.

4 Choose Run.

CUSTOM.MAC, the macro that actually creates a selection dialog box and implements your enable/disable choices, now displays the Disable Toolbar Buttons dialog box, as shown in Figure B.3.

Figure B.3: The Disable Toolbar Buttons dialog box



- To disable toolbar/ribbon buttons and their corresponding menu commands (if any),
 - 1 Check the check box beside the name(s) of the button(s) you want to disable.
 - 2 Choose OK.

Your enable/disable selections are put into effect, and the appropriate toolbar/ribbon buttons and menu commands appear dimmed and unavailable.

Note: Although you won't see it right away in the Disable Toolbar/Ribbon Buttons dialog box, some check boxes become checked whenever any other check box in that group is checked. For example, checking the "Sum" check box will also disable the Absolute Value, Minimum, Maximum, and Item Count toolbar buttons (and check their corresponding check boxes), because the menu command used to access these features is disabled. Similarly, disabling the Font box in the Ribbon also disables the Point Size box.

- 3 Save the report, but do not distribute CUSTOM.MAC with it.

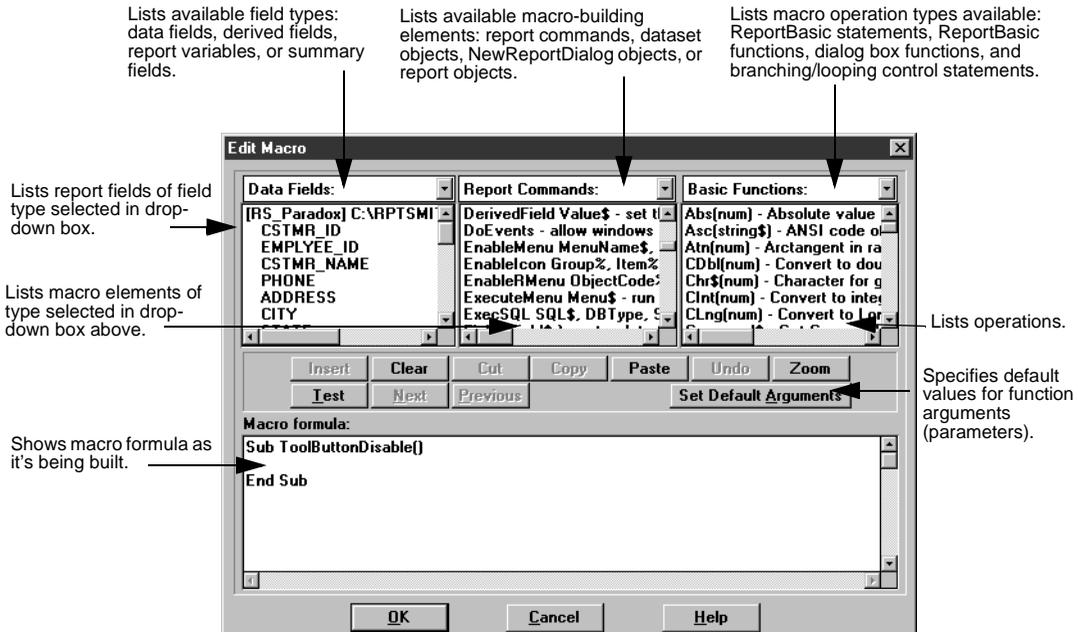
Customizing ReportSmith menus

As mentioned earlier, the Disable Toolbar/Ribbon Buttons dialog box will disable some, but not all, of the ReportSmith menu commands. Because it is geared toward the toolbar and the ribbon, the dialog box disables only those menu commands that correspond to various toolbar or ribbon buttons. To disable particular menu commands, irrespective of their toolbar or ribbon counterparts (or even whether they *have* such counterparts), you use the *EnableMenu* macro command.

- To enable or disable menu commands,
 - 1 Choose Tools|Macro.
 - 2 The Macro Commands dialog box appears, as in Figure B.1 on page 426.
 - 3 Enter a name for the new macro, and then choose New.

The Edit Macro dialog box appears, as in Figure B.4.

Figure B.4: The Edit Macro dialog box



Notice that ReportSmith has already entered two lines of your macro formula for you:

- “ToolButtonDisable ()” creates a *subroutine*, or ReportBasic macro. This is followed by the macro name, “ToolButtonDisable”, and parentheses to enclose any needed arguments or parameters. (In this case, none are needed.)
- “End Sub” marks the end of the macro. All other macro instructions will go between these lines.

You are creating a macro that will apply to the report as a whole, so you need not choose a field type (the far left drop-down box). Likewise, you don’t need to specify functions or statements (the far right drop-down box), nor will you need to set default arguments.

- 4 From the center list (Report Commands), select EnableMenu, and then drag this command into the Macro Formula window between the existing lines.
- 5 On the same line as the EnableMenu command, add the exact name of the menu and menu command, enclosed in double-quotes and separated by a vertical bar.
- 6 Finish this command line by adding a comma and the enabled/disabled flag: 0 to disable the command, or 1 to enable it.

For example, to disable the File|New command, your macro would read:

```
Sub ToolButtonDisable()  
EnableMenu "File|New", 0  
End Sub
```

As when using CUSTOM.MAC to disable toolbar or ribbon buttons, save and run this macro to put your choices into effect, but do not distribute the macro with reports to which it applies if you do not want users to change your settings.

Creating new menu commands

Unlike the toolbar and ribbon, ReportSmith’s menu structure does not limit you to enabling or disabling menu commands. You can create and add new menu commands as needed to suit your particular reporting needs.

The technique for creating and adding menu commands is very similar to that used to enable or disable them, but instead makes use of the *AddMenu* macro command.

- To create and add a menu command,
 - 1 Choose Tools|Macro.
 - 2 The Macro Commands dialog box appears, as in Figure B.1 on page 426.
 - 3 Enter a name for this new macro, and then choose New.

The Edit Macro dialog box appears, as in Figure B.4 on page 429.

- 4 Enter the following macro formula in the Macro Formula text box (or drag and drop the appropriate building blocks and then fill in particulars):

AddMenu "MenuText\$", "Macro\$", "AfterMenu\$", "HelpText\$"

- 1 For MenuText\$, substitute the actual menu and command names, separated by a vertical bar—for example, "Tools|Disable Toolbar Buttons". You must enclose this entire string in double-quotes.
- 2 For Macro\$, substitute the name of the macro file to be called and run by the new menu command—for example, "CUSTOM.MAC". This string must once again be enclosed in double-quotes.

Tip:

While it is possible to have your new menu item call active report or global macros that have not been saved to .MAC files, you will find it easier and safer to save such macros to .MAC files and call them by specifying full path and file names in the AddMenu command line.

- 3 For AfterMenu\$, substitute the name of the existing menu command that your new command should follow. For example, if you want your new menu command of Tools|Disable Toolbar Buttons to follow the existing Tools|Options command, enter "Tools|Options" for this parameter (including the enclosing double-quotes).
- 4 For HelpText\$, substitute whatever text you want to appear in the status bar when the mouse pointer is over this menu command. For example, you might want your Tools|Disable Toolbar Buttons command to have a help text of, "Disables and enables toolbar and ribbon buttons."
- 5 Choose OK to return to the Macro Commands dialog box.
- 6 If you want your macro linked to a specific event (such as opening a report or ReportSmith itself), choose Links and set the linking options.
- 7 Choose Run to run the macro now, or choose OK to return to ReportSmith.

Removing menu commands

You can also remove ReportSmith menu commands, either default commands or custom commands you have created yourself. To do this, you use the KillMenu macro command.

- To remove a menu command,

- 1 Choose Tools|Macro.
- 2 The Macro Commands dialog box appears, as in Figure B.1 on page 426.
- 3 Enter a name for this new macro, and then choose New.

The Edit Macro dialog box appears, as in Figure B.4 on page 429.

- 4 Enter the following macro formula in the Macro Formula text box (or drag and drop the appropriate building blocks and then fill in particulars):

KillMenu "Menu\$"

- 1 Substitute the actual menu and command names (such as "File|New") for "Menu\$" in the code fragment above.
- 2 To have this menu command removed for all reports in this ReportSmith session (rather than a single report), precede the menu name with an exclamation mark, like the following: KillMenu "!File|New".
- 3 This macro command returns a value of 0 if a menu item is removed successfully, or -1 if the menu item is not found. Names must exactly match existing menu commands, excluding vertical-bar characters and ellipses (...). If you omit the vertical-bar character and the command name, ReportSmith assumes you are working on a top-level menu and removes that menu and all of its commands.
- 5 Choose OK to return to the Macro Commands dialog box.
- 6 If you want, link the macro to an event by choosing Links.
- 7 Run the macro, or choose OK to return to ReportSmith.