System
Administrator Kit

Automating RUMBA 8.0 Software Guide

MICRO FOCUS

Contents

1 Automating Complex Tasks

By automating complex and common tasks, both you and your users can make RUMBA software do more for you. You can use either macros or scripts to simplify your work. Macros record a series of actions, such as keystrokes and commands to automate tasks.

If you choose to install the separate Script Editor you can use scripts to automatically manipulate sessions. For example, you could write a script to open a new session and connect to a host.

The Automating RUMBA Software Guide provides you with background information on using macros. Detailed information on the RUMBA Script Editor can be found in its online help.

Note

One of the easiest and most effective ways to familiarize yourself with RUMBA Script Editor/Player capabilities is to run the demo scripts provided with this application. Run one of the script files (such as, demo.csf) to see how a script works. You can also copy a sample script file and edit it to suit your needs.

Use macros in conjunction with Hotspots, custom toolbar buttons, and keyboard shortcuts to complete tedious log-on procedures and other common tasks with a single mouse click or key stroke.

For example, distributing macro files and Hotspot libraries to your users provides them with valuable production tools that they do not even have to take the time to create.

This chapter discusses the following topics:

- This macro looks at the current PROFS screen and returns a message box containing a description of what was found. You can use this macro as a model to test different macro functions.

Creating macros

Macros record a series of actions, such as keystrokes and commands. Once you record a macro, you can play it back to automate routine tasks. For example, with the click of a button, a macro can connect your PC to a host, test the return strings, and then start an application.

There are two ways to create a macro:

- Choose Record Macro from the Tools menu or click  , and then perform your keystrokes and commands. Be sure to save the sequence as a macro file after recording.
- Choose Edit Macro from the Tools menu, and then build a sequence of commands with the Macro Editor.

To combine the speed of recording macros with the power of written command lines, record the basics, and then use the Macro Editor to edit and add to the macro sequence. For details about macro commands, see Appendix A at the end of this guide.

Connect and disconnect macros

Connect macros run automatically each time you connect to a host. For example, the connect macro could contain commands to enter a User ID and password, choose menu items, and open a host application. You can distribute to your users macros that automatically log them on and take them to the correct host application.

In order for a connect macro to work, the communications interface must be properly configured, and the macro must be saved from within the profile with which it is to be used.

Disconnect macros run automatically each time you disconnect from the host. For example, the disconnect macro could run another program on the connected host before logging out. It is important to note that, if you have a disconnect macro defined, you do not disconnect from the host until after the macro has been performed. If you are disconnected from

the host because the physical connection has been broken, then the disconnect macro does not run.

Specifying connect and disconnect macros

Any given profile can have only one connect and one disconnect macro.

To specify that a macro automatically runs upon connecting or disconnecting:

1. From the Tools menu, choose Edit Macro. This opens the Macro Editor dialog box.
2. Select a macro from the Macro Name box.
3. Under the Play Macro On area, check either the Connect or Disconnect box.
4. Select Save from the File menu to save the macro, and close the Macro Editor.

For more information

Help topic: Simplifying your work > Automating Tasks > Macros > Creating and editing macros

Macro Editor

Recording keystrokes and commands may not always produce the macros you need to accomplish complex tasks. When it does not, use the Macro Editor to work with all the macro commands and create complex macros using advanced macro features. Appendix A shows the various commands available for each host.

You cannot record many of the macro commands, such as Pause Macro, Goto Label, Compare Values, Message, and Prompt, because they are not commands or keystrokes. For example, you may need a macro that pauses between two macro commands to give the host enough time to respond.

Copying data between applications

You can use Copy From and Paste To commands in the Macro Editor to designate an area of a display screen to be copied from one application and pasted into another application. In order to do this you need to specify the application and the location (in terms of rows and columns) from which the data should be copied, and the application and location to which it should be pasted.

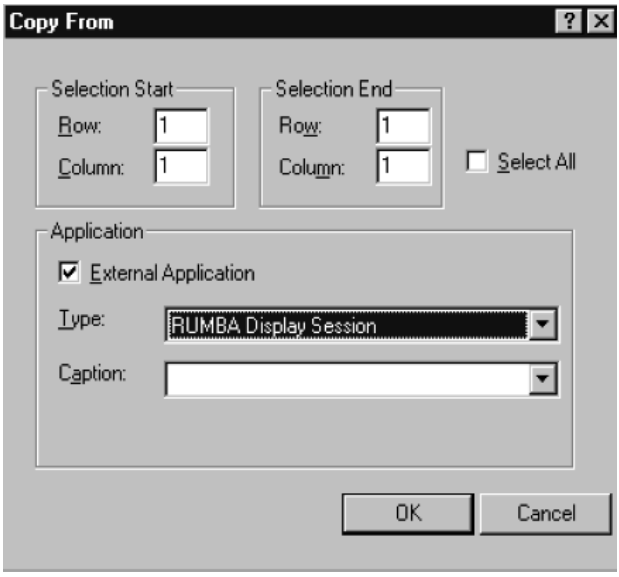

Figure 1-1 Copy From dialog box

Note

If you select Microsoft Excel 97 as the application type when using either the Copy From or Paste To macro, the dialog box changes and you must provide additional information in the File and Worksheet fields.

Using variables

Variables are one of the most powerful macro features. Using variables is like writing your own macro-driven application. With variables, you can create interactive macros that respond to user input and perform specific sets of commands.

To open the Variable Setup dialog box:

1. From the Tools Menu choose Edit Macro to open the Macro Editor dialog box.
2. Right-click in the macro display area.
3. Choose Variable Setup from the popup menu.

In the Variable Setup dialog box, you can select a variable, such as a password or user ID, and define its value based on host screen information, static text, and user input. The variable values are available for the Type, Message, Compare Values, Goto Label, and Prompt commands.

For example, you could create a prompt dialog box to request a password from a user, and then save it as a variable. The variable containing the user's password is then available for any Type, Compare Value, and Message commands you have included in the macro sequence. For example, you could use a Compare Values command to check the password, and then end the macro or even disconnect the PC from the host if an incorrect password is entered.

Example

The following example shows you how to define a screen variable and use it to execute specific actions based on a variable returned by PROFS.

This is a long example, which contains examples of creating several different kinds of macro commands. If you are unfamiliar with macros, we recommend that you create the macro contained in the example to familiarize yourself with the macro editor and macro command syntax.

1. Choose Edit Macro from the Tools menu.
2. Name the Macro “Flow Example.”
3. Right-click in the macro display area, and choose Variable Setup from the popup menu to open the Variable Setup dialog box. Click the Variables list box, and choose Variable for screen text.
4. From the Variable Types box, choose Screen Variable, if it is not already selected.
5. Click the Screen Variable tab.

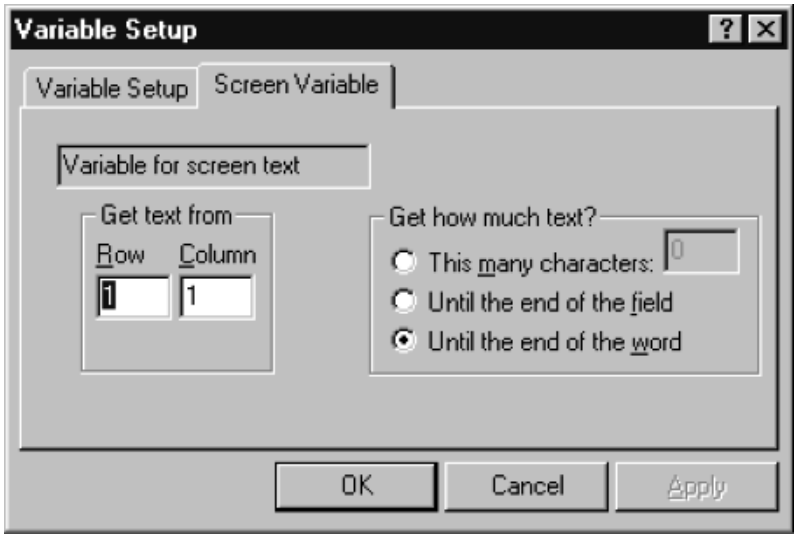

Figure 1-2 Screen Variable tab on the Variable Setup dialog box

6. Change the Get text from Column value to 78.
7. In the Get how much text? area, click This many characters, and type 3 in the box.
8. Click OK to save the settings.

   This gets 3 characters from the screen at row 1, column 78 and saves them as a variable that your macro can use.

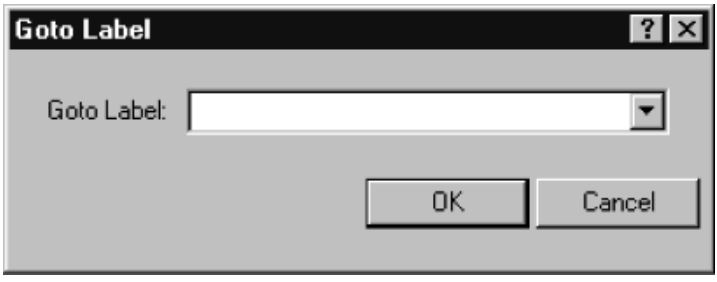

Figure 1-3 Goto Label dialog box

The following steps reference buttons on the toolbar above the macro display area.

9. Click  to open the Goto Label dialog box. Select Variable for screen text from the label list, and click OK.

10. Click  to open the Message dialog box. Type Screen Unknown and click OK.
11. Click  to open the Goto Label dialog box. Type Macro End and click OK.
12. Click  to open the Label dialog box. Type A00 and click OK.
13. Click  to open the Message dialog box. Type PROFS Main Screen and click OK.
14. Click  to open the Receive dialog box.
15. Type Mail Waiting in the Find text box.
16. Under Screen Position, click Only at row, column, and type 24 and 69 in the Row and Column boxes.
17. Under Timeout, click Seconds, and type 30 in the box.
18. In the Otherwise box, type NoMail, and click OK.
19. Click  to open the Message dialog box. Type You have Mail and click OK.
20. Click  to open the Label dialog box. Type NoMail and click OK.
21. Click  to open the Goto Label dialog box. Type Macro End and click OK.
22. Click  to open the Label dialog box. Type C00 and click OK.
23. Click  to open the Message dialog box. Type You are on the Open Mail Screen and click OK.
24. Click  to open the Goto Label dialog box. Type Macro End and click OK.
25. Click  to open the Label dialog box. Type “E01” and click OK.

26. Click  to open the Message dialog box. Type You are reading a message and click OK.

27. Click  to open the Label dialog box. Type Macro End and click OK.

28. Click  to add Exit Macro to end of macro.

Your completed macro should look like this:

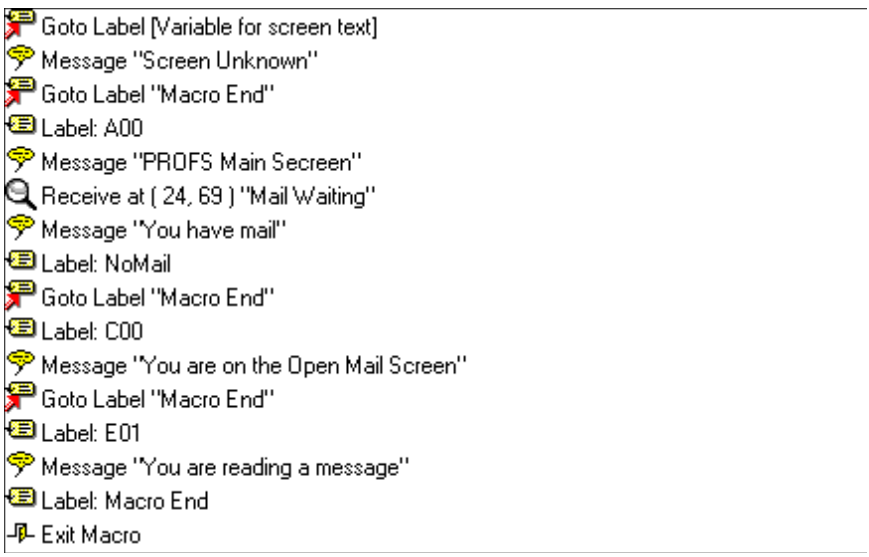

Figure 1-4 Example Macro

29. Choose Save from the File menu, and close the Macro Editor.

This macro looks at the current PROFS screen and returns a message box containing a description of what was found. You can use this macro as a model to test different macro functions.

2 Running Macros

This chapter assumes that you have created your macros. It tells you and your users how to run the standard RUMBA macros and your custom macros in any of the several ways.

This chapter discusses the following topics:

- Running standard or custom macros
- Macros as Hotspots

Running standard or custom macros

- From the Tools Menu, choose Run Macro or click  , and then select the macro you want to run.
- Associate a macro with a Hotspot, and then click the Hotspot text on the host screen to run the macro.
- Assign a macro to a toolbar button, and then click the button to run the macro.
- Link a macro to a keystroke, and then type the keys to run the macro.
- Macros can also be run from other macros. When editing the Run application command, select Macro Files from the Files of Type list. Choose the macro you want to run. When the macro comes to this command, it pauses, runs the new macro, and then runs the remainder of the originating macro.

Each host application type has a default macro directory.

Macros as Hotspots

Hotspots are words or sentences on the host screen that run macros when clicked. Hotspots add point-and-click capability to host screens, making them more like the other Windows applications. Instead of requiring users to master complicated command line statements, you can distribute Hotspot libraries to enable users to click through host screens with fewer errors and more efficiency.

You can find procedures for creating and editing Hotspots in the online help.

For more information

Help topic: Simplifying your work > Automating tasks > Using hotspots on host screens > Creating and editing hotspots

What are Hotspots

You can make any non-protected text on a host screen into a Hotspot and associate that text with any macro. The host screen shows Hotspots as 3D buttons (Figure 2-1). When you click a Hotspot button, the macro runs.

For example, you could make the text in a list of options into Hotspots that automatically enter the appropriate option when clicked.

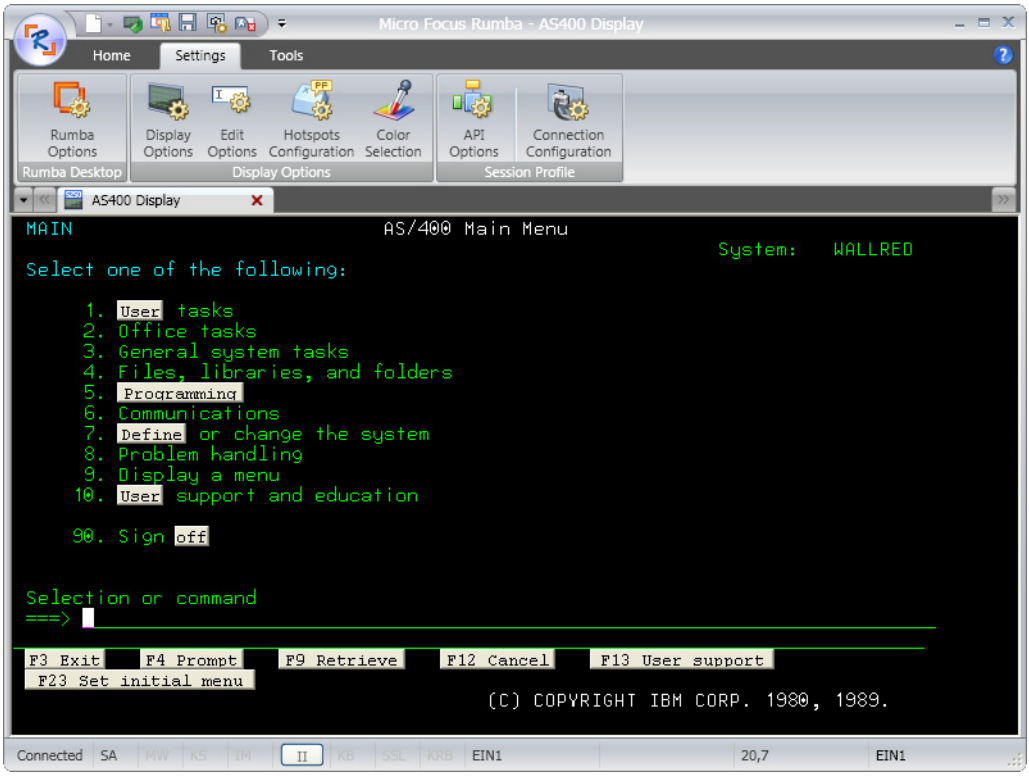

Figure 2-1 AS/400 Display window shows Hotspots as 3D buttons

Hotspots are saved in libraries with an **.hsp** extension. You can create multiple libraries with distinct configurations. To use different collections of Hotspots, or to run different macros with the same Hotspot text, users just select the corresponding library.

Creating and distributing Hotspot libraries

You can create whole libraries of Hotspots and distribute them to users. When the user enables Hotspots, the macro runs when the user clicks the Hotspot. The macros used in your Hotspots are saved with the Hotspot library, so you do not need to distribute the **.hsd** files to your users.

Macros on your toolbars

Macros can be linked to toolbar buttons, and incorporated into your standard RUMBA Toolbar or placed on custom toolbars. With custom toolbar macros, your users can accomplish complex key sequences and commands without errors and without even touching their keyboards.

For more information

Help topic: Simplifying your work > Automating tasks > Working with Macros > To assign a macro to a toolbar button

Keyboard macros

If your users do not like switching between keyboard and mouse as they work, they can link macros to keystrokes. Without even taking their hands off the keyboard, your users can run the macros you distribute which create shortcut keys for complex key sequences.

Distributing Keyboard files

The keyboard files are located in the private directory\hostname\keyboard directory as **.map** files. To distribute keyboard files, you must distribute the **.map** file and any corresponding macro files (**.rmc** & **.mac**) that are linked to mapped keys. The macro files are located in the private directory\hostname\macros directory.

Macro Commands

The macro commands available for each host type differ slightly. The following tables list the macro commands available for each host. Each table lists the macro command, followed by a brief description of its function.

The tables are presented in the following order:

- Mainframe macro commands
- AS/400 macro commands
- HP and UNIX macro commands

Mainframe macro commands

	Use to:
Connect	Connect the PC to the mainframe host
*Compare Values	Compare two character strings or variables, and perform the specified commands based on the results
Cursor Position	Position the cursor on the host screen
Disconnect	Disconnect the PC from the mainframe host
Exit Macro	End the macro sequence
File Transfer Receive	Receive the specified file transfer
File Transfer Send	Send the specified file transfer
*Goto Label	Jump to the specified Label command in the macro sequence
Keystroke	Send a function key to the host application
*Label	Create a marker in the macro sequence to which a Goto Label command can jump
*Message	Show a dialog box with the specified message
*Pause Macro	Pause the macro sequence for the specified number of seconds
*Prompt	Create a dialog box that prompts the user for input
Receive	Search the host screen for the specified characters
*Run Application	Run an application on the local PC
Copy From	Copy selected areas of a display screen to the clipboard for pasting into an application or display screen
Paste To	Paste clipboard information to an open application or session.
Type	Send the specified characters or variable to the host screen

An asterisk (*) indicates macro commands that you cannot record, but can create and edit in the Macro Editor.

AS/400 macro commands

	Use to:
Connect	Connect the PC to the AS/400 host
*Compare Values	Compare two character strings or variables, and perform the specified commands based on the results
Cursor Position	Position the cursor on the host screen
Disconnect	Disconnect the PC from the AS/400 host
Exit Macro	End the macro sequence
*Goto Label	Jump to the specified Label command in the macro sequence
Keystroke	Send a function key to the host application
*Label	Create a marker in the macro sequence to which a Goto Label command can jump
*Message	Show a dialog box with the specified message
*Pause Macro	Pause the macro sequence for the specified number of seconds
*Prompt	Create a dialog box that prompts the user for input
Receive	Search the host screen for the specified characters
Copy From	Copy selected areas of a display screen to the clipboard for pasting into an application or display screen
Paste To	Paste clipboard information to an open application or session.
*Run Application	Run an application on the local PC
Type	Send the specified characters or variable to the host screen

An asterisk (*) indicates macro commands that you cannot record, but can create and edit in the Macro Editor.

HP and UNIX macro commands

	Use to:
Connect	Connect the PC to the UNIX or HP host
*Compare Values	Compare two character strings or variables, and perform the specified commands based on the results
Disconnect	Disconnect the PC from the UNIX or HP host
Exit Macro	End the macro sequence
File Transfer Receive	Receive the specified file transfer
File Transfer Send	Send the specified file transfer
*Goto Label	Jump to the specified Label command in the macro sequence
Keystroke	Send a function key to the host application
*Label	Create a marker in the macro sequence to which a Goto Label command can jump
*Message	Show a dialog box with the specified message
*Pause Macro	Pause the macro sequence for the specified number of seconds
*Prompt	Create a dialog box that prompts the user for input
Receive	Search the host screen for the specified characters
*Run Application	Run an application on the local PC
Type	Send the specified characters or variable to the host screen

An asterisk (*) indicates macro commands that you cannot record, but can create and edit in the Macro Editor.

Index