

# Locate

Mathew Wilson

**COLLABORATORS**

	<i>TITLE :</i> Locate		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Mathew Wilson	August 24, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Locate</b>	<b>1</b>
1.1	Locate	1
1.2	About Locate	2
1.3	Using Locate.	2
1.4	Creating an Address Entry.	2
1.5	What an Address Consists Of.	3
1.6	Using the Address List.	4
1.7	Drag and Drop Script	4
1.8	Using External Information Files.	4
1.9	Searching the List.	5
1.10	Using the Dialer	5
1.11	The Configuration Window.	6
1.12	Merge Loading an Address File.	7
1.13	Importing ASCII files	8
1.14	Printing the Address List.	8
1.15	Altering the Field Names.	9
1.16	Why not register today	9
1.17	Thanks to...	10
1.18	In the Future...	10
1.19	Disclaimer.	10
1.20	arexxscripts	11
1.21	ARexx Commands	11
1.22	rx_about	12
1.23	rx_active	13
1.24	rx_clearall	13
1.25	rx_clearlist	13
1.26	rx_dial	13
1.27	rx_getactive	13
1.28	rx_getentries	14
1.29	rx_getfield	14

---

1.30	rx_getfilename	14
1.31	rx_gettextnotedir	15
1.32	rx_gettextnotename	15
1.33	rx_listformat	15
1.34	rx_load	15
1.35	rx_notedelete	16
1.36	rx_noteedit	16
1.37	rx_noteexamine	16
1.38	rx_phoneactive	16
1.39	rx_putfield	16
1.40	rx_remove	17
1.41	rx_rx	17
1.42	rx_save	17
1.43	rx_saveas	18
1.44	rx_scan	18
1.45	rx_search	18
1.46	rx_select	18
1.47	rx_view	19

---

# Chapter 1

## Locate

### 1.1 Locate

Welcome to the Locate v1.0 User Guide.

~~~~~  
About Locate.

Using the Program.  
-----

How to Create and Address Entry.

What an Address Consist Of.

Using External Information Files.  
-----

Using the Address List.

Searching the List.

Using the Dialer.  
-----

Merge Loading an Address File.

Importing ASCII files.

Printing the Address List.  
-----

Configuration.

ARexx Command Set.  
-----

Thanks to...

In the Future...

Disclaimer.

---

## 1.2 About Locate

About Locate.

Locate© is one amongst a million Address Book programs littered throughout the computer world. There are enough of them around in varied forms so that you can usually find one that suites your own personal needs. I however could not find one that had all the things I needed and suited my philosophy in program design, so I wrote my own. Here is my input to the wonderful world of Address Book programs 8).

Locate functions mainly as an Address lister but it can be altered slightly to lend itself into other areas. Read the entire document to find out what makes this one different from the rest.

Locate is a MUI(Magic User Interface) application. This means you need the MUI files/library to use the Locate. Version 3.1 of MUI is required. I chose to develop using MUI because of its flexibility and easy of use, both for the user and programmer. It allows a fairly flexible environment for both the programmer and the user. MUI is hopefully the direction of the future. MUI is available as a reduced product on the Public Domain, but if you wish to have full control, and support a great product that supports the continuing life of the Amiga, then [register](#) for the full version.

Locate© is classified FreeWare. See [Disclaimer](#) .

Mathew Wilson

Contact me at:

Box 6022, Halifax Street Post Office

Adelaide SA, 5000

AUSTRALIA

At present I can also be e-mailed via David Cottrell

dcottrel@adam.com.au

## 1.3 Using Locate.

Using Locate.

Locate has been designed specifically for ease of use and to provide a variety of useful functions. The program has a standard MUI list and has all the fields for an Address listed below it. This is the fundamental philosophy for all my programs - "Where possible, all data for a list entry should be displayed in the same window as the main list." This way you can activate an entry in the list and see all its associated data instantly. Other programs only allow you to see this information by entering a separate edit window. This slows the process down quite a bit, don't you think? The down side to this is for people using their Workbench in a plain 'High Res' - no Laced mode. In this case everything will fit on the screen but the actual lister gets quite small indeed.

Although Locate may not necessarily be used as an address lister, we shall refer to an individual entry as a collection of fields making an "Address."

The program uses standard tooltypes to store its configuration settings. These are parameters set in the programs/caller's icon that the program looks for at run-time. The settings are generally global and all Address information is saved to a separate file. It is important to realise the difference between saving file-settings and saving Address lists!

## 1.4 Creating an Address Entry.

Creating and Address Entry.

As previously mentioned in the introduction, some programs open "edit" windows when you create an entry and it adds the entry if you accept the new data. Locate works slightly differently to help you speed up data entry and create a system thats functional and easy to use.

This poses some questions for the new user, for example;

---

"Without an accept button, how does the program actually know when to add an Address to the list?"

The answer to this is very simple. Locate requires that a new Address contain something in either or both of the first two fields; Last Name and First Name entry. Locate will add the current Address into the list whenever this requirement is met.

To create a new entry all you must do is deselect anything in the main list which in turn clears the Address fields below the list. This can be done by either clicking on the "New" button or pressing the "DELETE" key from the keyboard (no gadget selected). In the first case, selected or active entries in the list are deselected, the data fields below are cleared and the cursor appears in the first field string gadget. The latter will not activate the "Last Name:" gadget. You will now be in "ADD" mode.

In other words, when nothing is selected in the list, the Address data fields should logically be showing no data from the list as they indicate the currently active entry. So usually whenever you see that the Address fields are ALL blank then you know you're in ADD mode.

So just start entering data and when Locate recognises that the NEW Address criteria are met then it will insert the current data into the list and THEN the entry will appear activated in the list. This now indicates that you are NOT in ADD mode but are in EDIT mode. To edit the entry all you must do is activate it in the list (if it isn't already), change the data and press return to accept it.

NOTE: Locate requires that you **\*\* MUST \*\*** press RETURN after entering data into anyone on the Address data fields. Failure to do this WILL result in the loss of this data! Pressing return means that you are accepting the data. This is a side-effect of the method of data entry used.

There are short cut keys for each of the fields. Press the number 1 to 7 that corresponds with the ordering of the fields.

## 1.5 What an Address Consists Of.

What an Address Consists Of.

An Address must consist of either or both the first two fields. For the sake of clarity we shall refer to the fields with their default names as they can be **renamed** to suit the language or situation.

### 1. Last Name

This field may contain up to 128 alpha/numeric characters.

Editing this field in ADD mode will cause the current Address data be inserted into the list as a new entry.

In EDIT mode the entry will be resorted in the list.

### 2. First Name

This field may contain up to 128 alpha/numeric characters.

Editing this field in ADD mode will cause the current Address data be inserted into the list as a new entry.

In EDIT mode the entry will be resorted in the list.

### 3. Street

This field may contain up to 128 alpha/numeric characters.

### 4. Suburb

This field may contain up to 128 alpha/numeric characters.

### 5. Post Code

This field may contain up to 20 numeric characters.

### 6. Phone

This is a MUI PopString gadget. It is basically a mini-list of phone-number strings. Currently you can have up to 4 entries in this gadget. Just select the "pop" gadget to the right of the string and a standard list of phone numbers will popup. If you are unfamiliar with MUI-PopString gadgets then please read the MUI documentation.

The pop-list may be left open all the time if you wish. This enables you to see the entries as you scroll through entries in the main Address list. If you are ADD mode then the list will default to four standard strings - "Home:", "Fax:", "Work:" and "Email:".

---

You may use these as standard guides to your data entry or you may use your own but it shows that it's a good idea to pre-fix phone numbers with an ID of some sort.

The phone strings may contain up to 60 alpha/numeric characters.

Refer to [Dialing numbers](#) for a full description of the "Dial" button.

7. Notes:

This field may contain up to 128 alpha/numeric characters.

The button marked View is used for viewing and editing [external data files](#) for each individual entry.

## 1.6 Using the Address List.

Using the Address List.

The main list can be used as like any standard MUI List.

You can remove entry(s) with the "Remove:" gadget below the list.

To deselect any active or highlighted entries from the list just press the DELETE key on the keyboard.

You may even change the display/order of items in the list by altering the "List Format:", described in [Configuration](#) .

At the top of the main window where the window title is, you will see some extra information displayed. It will either display "No Entries" or some numbers in the form "#:# of # (#) xxxx". This displays the number of the active entry, the number of selected entries, total Address entries (for a description of the difference between Active and Selected entries, see your MUI documentation), the unique ID number for the Active entry and the name of the currently loaded file.

In addition to the normal loading methods, you may drag-and-drop icons onto the Main List or its app-icon (when iconified). This is performed via a User-Configurable [ARexx Script](#) .

## 1.7 Drag and Drop Script

Drag and Drop Script.

When an Icon is dropped onto the Locate Main List an ARexx script named 'Locate\_DropObj.rexx' is called with its first argument being the name of the File that was dropped. Locate will expect the script to be found in the 'REXX:' drawer.

This method enables the user to specify what exactly is to be done with files that are dropped onto Locate. The normal outcome of such an event would be to attempt to load the file. This is the standard function of the default script that is installed with Locate, but you may of course alter this to suit your own needs.

Like all scripts that are run from Locate, the invoked script has its host address set to the Locate port automatically. This way the script will know exactly who to send its messages to (see example).

## 1.8 Using External Information Files.

Using External Information Files.

Sometimes you need to store an unknown amount of data about a particular person. In a business situation this data could be contact dates and enquires etc. This sort of data does not need to be loaded into memory all the time but you need to be able to access it quickly.

Locate's external data files are a solution to this problem. They are information files that are stored in a directory of their own, ready to be called up when needed.

In the main window next to the "Notes:" data field is a button labelled "View". When you click on this button a window labelled "Notes View Window" opens. Locate will now look for an information data file that belongs to the current entry selected. If it is found then its contents will be displayed in this window.

---



You may leave this window open. If you do then Locate will look for an information file everytime you select and entry in the main list. This may slow things down more than usual because Locate is accessing the disk and attempting to load data everytime you select a new entry. So you have the option of leaving the window opened or closed.

In the Notes View Window is a button labelled "Edit." This is an important button because this enables you to create or alter data in the information file(s). Locate will call an external editor of your choice to edit this data. You may instruct Locate to use your favourite editor in the [Configuration](#) . If the editor doesn't detach itself from the current process then Locate will wait till you finished editing and then update the new information in the Notes View Window. Locate waits for the editor to return, but some editors detach themselves from the current process and return control. This mean that Locate will not know when to update the Notes View Window. You can easily update it by reselecting the entry in the main list.

Locate will look for the information files in a particular drawer. The location of this drawer is user-configurable and is saved with the Address file. See [Configuration](#) .

Locate provides a method for determining at a glance whether or not an Address has any external information notes. You can do this be setting the [List Format](#) option with column 7 (col=7) showing. Locate will fill this column with either a " ", ".", "-", or "\*" meaning:

" " (Blank) - Locate has not searched for an external notes data file for this entry.

." - Locate has looked and not found a found the data file for this entry.

.-" - A data file exists for this entry but it is empty

.\*" - A data file exists for this entry and it contains information.

To scan for information files you can use the "Scan Notes" menu option under "Functions". Alternatively, if you have column 7 showing as a default in your List Format then you might like Locate to scan for data files every time it loads the Address List. This can be accomplished with the "Scan Notes:" option in the [Configuration](#) .

## 1.9 Searching the List.

Searching the List.

Below the Address list is a string gadget labelled "Search:". Just type some text in here, press return and Locate will search the list for a matching entry. It will start searching straight after the Active entry or from the top if nothing is Activated in the list.If you want to deactivate the list just press DELETE from the keyboard or select "NEW".

Searching is performed using standard AmigaDOS pattern matching (see you AmigaDOS manual) so make sure you enter the search string correctly!

eg. a search string of "Ami" will NOT match "Amiga" because it is looking for an exact length match. Instead you would type "Ami#?" to match "Amiga". (Note: Some programs patch "#?" to allow you to use "\*" instead.)

The parameters for the search are set in the [Configuration](#) .

## 1.10 Using the Dialer

Using the Dialer.

Locate has a built in serial dialer attached to the phone data field. This means that if you have a modem you can use Locate to dial numbers directly from the Address list.

Just activate the entry from the list you would like to dial and select the correct phone number from the phone lister and press the "Dial" button. A requester stating what number is being dialed will appear. It will also ask you to close the requester (press return) when you have picked up the phone.

If you have your modem speaker turned on you should hear the modem dialing. When you hear the standard dial tone, you may pick up the phone and close the requester to hangup the modem.

A pre-dial string may be sent to the modem. This is set in the [Configuration](#) .

NOTE: The dialer will remove all alphabetical characters from the dial string before it sends it to the modem!

---

## 1.11 The Configuration Window.

The Configuration Window.

This window contains three pages of settings and defaults for Locate. Only those in the group labelled "Defaults" get saved in the configuration which are saved as tooltypes. Opening this window will not send the program to sleep.

Search Parameters

Case Sensitivity: {On, Off}

This controls the case sensitivity of list **searching** . If you turn this on then Locate will only find entries that exactly match the case of the search string.

Fields:

This cycle gadget allows you to select which Address fields are included when performing a search.

File Settings

Password:

Locate enables you to set password protection on an Address list so that a password must be entered correctly everytime the file is loaded into Locate. The password may be up to 30 characters long. To enter a password just activate the string gadget, type the password and then press return. Now the gadget will clear and become active again. It is waiting for you to verify the new password by retyping it. Once this is accomplished successfully, the password will be displayed in a standard "hidden" form.

Now just save the current Address file in the standard way from the pull down menus. Everytime the file is loaded you will be required to retype this password. You get two tries. If you fail then the file will NOT be loaded at all.

File Encryption: {On, Off}

Password protection wouldn't be too much use if you could still read the Address file in a standard text viewer. This option will Encrypt the Address file upon saving and Decrypt it when loaded successfully. With this option, no-one will be able to read the file from a text viewer. So don't forget your password 8).

Field Names:

This button allows you to alter the **field names** .

List Format: default ", "

This is a simple string gadget that enables you to change the format of the Address list. This is a standard feature of MUI lists and you can do things like change the order of the columns.

The default list display is - Last Name, First Name

You could change this to - First Name, Street by entering "col=1, col=2" in the List Format gadget. "col" stands for Column and represents the columns of the default display above. Entering just ", " (the default) means to just use the default columns. The ", " (comma) represents the spacer between each column.

The columns are assigned as follows: (using the default english **field names** )

0 = Last Name

1 = First Name

2 = Street

3 = Suburb

4 = Post Code

5 = Phone entry 1

6 = Notes String

7 = External Notes Info.

Scan Notes: {On, Off}

---

If you are using column 7 (External Information **Files** Info) then this option will enable scanning the files upon loading of an Address list. This way 'Column 7' will immediately indicate if an entry has an external data file.

Defaults

Default File:

This pop-file gadget enables you to set a default file that will be loaded everytime Locate is run from Workbench. This is a useful option considering most people only have one Address list. So don't forget to set it.

If you hold down the left mouse button after running Locate, it will not load the default file.

Notes Dir:

This pop-drawer gadget lets you choose the source drawer that Locate will look in for the external information files. This drawer must be set if you are to use external information files.

Note Editor:

This pop-file gadget lets you select the text editor of your choice for editing data in the information files. It basically tries to execute the given file and gives the relevant External Note Data File as an argument. You could call an ARexx script with 'c:rx <editor>'. The ideal Text Editor would be one that has REAL word wrap and doesn't fake it with annoying carriage-returns. If anyone finds one, please let me know 8). (such a basic thing)

Pre-Dial:

When using the **Dial** function you may want to send a string to the modem first. This gadget represents this optional string.

Confirm Remove: {On, Off}

This check mark gadget enables user-confirmation requests when performing a "Remove:" function on the list.

Window Quit: {On, Off}

If you place Locate in your WBStartup drawer then the program will load everytime you reboot. In such a case you would most likely have Locate start Iconified. This can be accomplished by altering the MUI settings for the program. Normally Locate will quit when you close its Main Window. By turning OFF the Window Quit option you can have Locate Iconify instead of quitting. To quit the program you can use the 'Quit' menu-item in the 'Project' pull-down menu.

The buttons "Load" and "Save" affect the defaults.

## 1.12 Merge Loading an Address File.

Merge Loading an Address File.

This option is enable from the "Project" menu.

It allows you to merge another standard Locate Address file into the current list. The file you select to Merge Load will NOT be affected, only the currently loaded list will be changed.

Whenever it finds an entry in the merge file that has the same Last and First Names as an entry in the current list it will open a requester displaying the information and ask you what you would like to do with it.

Add:

Adds this new entry into the list. It will NOT replace the matching entry in the currently loaded list.

Replace:

Replace the entry in the currently loaded list with the entry found in the Merge file.

Skip:

Skips over the entry found in the Merge file.

Abort:

Abort the Merge procedure all-together.

---

## 1.13 Importing ASCII files

Importing ASCII text files.

It is possible to import a certain type of structured ASCII file into Locate. If you have a list setup in another program and wish to import it into Locate you may use this function found in the "Project" menu.

To successfully import an ASCII file, Locate expects it to be in the following format:

- ° A RETURN (ASCII 13) or LINEFEED (ASCII 10) character terminates a single line in the ASCII file.
- ° Each line will be interpreted as a single record to be imported.
- ° The line will be formatted such that it contains individual pieces of data separated by TAB (ASCII 9) characters.

As each line in the file is interpreted by Locate, it fills out an Address entry using this data. When it comes to the end of the line it checks to see if there is **enough data** to create a new entry. If there is, then it will insert this new entry into the list, otherwise this information will be discarded and the next line read from the input file.

The ASCII format that Locate expects is exactly the same as the output when **printing** in Column format. So you could export using that function.

At the moment Locate will not check the format or contents of the file before it imports it. So please be certain that it is of the correct type.

## 1.14 Printing the Address List.

Printing the Address List.

Printing can be enabled in the "Project" menu.

A window with various features will appear.

Output:

This string gadget allows you to select the destination for all print output. The default is "PRT:" representing the preferences setup Printer.device. If you wish the output to be placed directly into a disk file then just type its path and file name.

Header:

The header string you would like to appear at the top of the output.

Print:

This cycle gadget selects what entries are to be printed. "Selected" will print only entries that are selected in the list and "All" will print all entries.

Perspective:

This cycle gadget selects either "Vertical" or "Horizontal" printing.

Vertical prints all data for an individual entry down the page.

Horizontal prints all data for an individual entry across the page. This appears as columns. Each column is separated with TAB characters. The latter method is probably more useful for exporting into another program or even a word processor where you can format the columns.

Print Field Names:

This option will cause the names of each field to be printed. If you have Vertical perspective selected then the fields will be printed on the left of each piece of data as it is printed successively down the page. With Horizontal perspective selected then the field names will appear as column titles.

Field Activation Check-Mark Gadgets:

There is a check-mark gadget for each Address field. You may turn them on and off individually for the printing process.

---

## 1.15 Altering the Field Names.

### Altering the Address Field Names

You may give different descriptions to the Address fields within Locate. This enables you to tailor a Locate file for a specific purpose. If you live in America you may just want to change the field name "Post Code:" to "ZIP:". It's all up to you.

The defaults for the english language are:

\*1 - Last Name:

\*2 - First Name:

3 - Street:

4 - Suburb:

5 - Post Code:

6 - Phone:

7 - Notes:

\* However, it is important to realise that the first two fields are the SORT fields. This means that Locate uses these two fields to keep an entry sorted alphabetically in the list. The first field has the greatest priority while the second has the least. This means that the first field is used for sorting and the second field is examined only if two of the first fields match exactly.

## 1.16 Why not register today

This application uses

MUI - MagicUserInterface

(c) Copyright 1993/94 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US\$ 20.-

to

Stefan Stuntz

Eduard-Spranger-Straße 7

80935 München

GERMANY

---

## 1.17 Thanks to...

Thanks to...

Thanks to Dave Cottrell for running this on his machine and letting me know when strange things occur.

I started programming with C in April '94 and I would like to thank Peter Jackson for his invaluable help with learning C on the Amiga when I had (and have) an extreme lack of documentation.

A big thanks to Stefan Stunz for creating and maintaining the MUI system.

Thanks to YOU for taking the time to read the docs 8).

## 1.18 In the Future...

Maybe In the Future...

- I would firstly like to sincerely apologize for the fact that this program is not fully object oriented in design. Locate has 4 different classes but the back bone of the program is still not split up into classes. I am relatively new to C programming (April '94) and I don't know any local experienced programmers to confer with (have no direct Internet access either!) With no technical manuals and very little documentation, I am in this alone. I am currently trying to teach myself many things, one being OO program design. Hopefully in the near future this program will be fully OO in design. If I stay with the Amiga then I will be committed to eventually produce OO designed programs. So please be patient and hold back the abuse for now 8). Hopefully I'll stay with Amiga.

- Possible language catalogs (catalog definitions are a real bore! 8). I'm part of the way there. All the Help bubble strings have definitions. Catalogs depend on whether anyone else is interested in my little insignificant programs apart from me 8).

- Any suggestions?

Wish List

- Some ACTUAL Beta testers who actually use the program 8) (no joke).

## 1.19 Disclaimer.

The following notice applies to the program "Locate" and everything contained within its original distribution archive.

° The program was written by Mathew Wilson.

° The program is classified FreeWare, which means it is:

- Freely distributable as long as its original archive remains intact.

- Copyrighted by its author and you are NOT permitted to modify the program(s) and documentation in any possible way.

° No warranties are given. No responsibility is taken for any possible use or misuse of the program(s) by its users.

The Author has tried to prevent errors but he can't guarantee that it is 100% bug free. You therefore use this program and any other associated program at your own risk.

° This package must not be sold directly or indirectly without prior written consent given by the Author. Although, it may be placed on any media used in the distribution of free software without consent from the Author.

The **MUI** system is Copyright Stefan Stunz.

---

## 1.20 arexxscripts

### User-Definable ARexx Scripts

Locate allows you to execute external ARexx scripts from within the program. In which case you must have ARexx operational on your system (refer Amiga manuals). Locate will not execute the scripts itself but tell ARexx to perform the scripts (standard procedure). The essential difference is that the default Host Port for the script will be Locate.x.

In the ARexx pull-down menu you will find the following items.

#### Run New Script...

A file requester will open. Select an ARexx script and close the requester. The file will be executed as an ARexx script.

#### Run Last Script

This calls the script previously invoked with the 'Run New Script' menu command. If no ARexx script exists then a file requester will open.

#### Install ARexx Script (1 to 4)...

These four menu items are where you can 'install' four different ARexx scripts. They can then be called quickly via the menu or menu shortcut key.

After selecting one of the four menu items, a file requester will open and you can select the script to install. The script will not be performed at this stage. Now, if you look at the menu item again, you will notice that the previous text has been replaced with the name of the file you selected. The file will appear without its path name.

#### Clear Script...

This menu command allows you to remove one of the scripts you previously installed with the 'Install ARexx Script' command. A requester will appear, showing all the currently installed scripts. Just select the one to remove and it will disappear from the ARexx pull-down menu. Of course the script itself will not be deleted, just the reference to it.

The four scripts installed will be saved along with the Main List (Not in the Configuration!). This allows you to install different scripts for each Locate data file.

You may invoke the execution of one of the four scripts via ARexx as well. See [RX command](#).

## 1.21 ARexx Commands

### Locate ARexx Command Set.

Apart from the built in MUI ARexx commands (see MUI documentation) there are many others to help you control Locate.

The commands return their result in the ARexx RESULT variable. Any errors will be reported in the RC variable (0 = success, any other value indicates an error).

The ARexx port is LOCATE.<x> where <x> is the number of the current program (like most MUI applications, you may run Locate more than once).

You can install four [ARexx scripts](#) for easy reference as well.

- A -

**About** (NO ARGS)

**Active** ENTRY/N,NOACTIVE/S

- C -

**ClearAll** (NO ARGS)

**ClearList** (NO ARGS)

- D -

**Dial** NUMBER,ACTIVE/S

---

- G -

**GetActive** (NO ARGS)

**GetEntries** (NO ARGS)

**GetField** NUMBER/N

**GetFileName** (NO ARGS)

**GetExtNoteDir** (NO ARGS)

**GetExtNoteName** (NO ARGS)

**GetSelect** ACTIVE/S

- L -

**ListFormat** FORMAT/F

**Load** FILE,MERGE/S

- N -

**NoteDelete** (NO ARGS)

**NoteEdit** (NO ARGS)

**NoteExamine** (NO ARGS)

- P -

**PhoneActive** ENTRY/N

**PutField** NUMBER/N,VALUE/F/A

- R -

**Remove** (NO ARGS)

**RX** FILE,SCRIPT/N/K

- S -

**Save** (NO ARGS)

**SaveAs** FILE,REQUEST/S

**Scan** (NO ARGS)

**Search** CASESENS/S,FIELD/N/K,STRING/F/A

**Select** ENTRY/N,NOSELECT/S,ALL/S

- V -

**View** STATE/N,TOGGLE/S

## 1.22 rx\_about

About

Purpose: Displays the About requester.

Syntax : About (no parameters)

Example: about

---



## 1.23 rx\_active

Active

Purpose: Set the active entry in the list. The data fields will be updated as well.

Syntax : Active ENTRY/N,NOACTIVE/S

ENTRY/N - The number of the entry.

NOACTIVE/S - Flag to de-activate the list (eg. no entry active). All data fields will be cleared.

Example: active 5

## 1.24 rx\_clearall

ClearAll

Purpose: Removes all entries from the current list and resets field names to their defaults.

Syntax : ClearAll (no parameters)

Example: clearall

## 1.25 rx\_clearlist

ClearList

Purpose: Remove all entries from the current list.

Syntax : ClearList (no parameters)

Example: clearlist

## 1.26 rx\_dial

Dial

Purpose: **Dial** a telephone number using the SER: device.

Syntax : Dial NUMBER,ACTIVE/S

NUMBER - The telephone number as a string of characters. (Non numeric characters will be stripped).

ACTIVE/S - Flag to dial the number from the currently active entry in the list.

Example: dial active

## 1.27 rx\_getactive

GetActive

Purpose: Return the number of the currently active entry in the list.

Syntax : GetActive (no parameters)

Example: getactive

active = result

---

## 1.28 rx\_getentries

GetEntries

Purpose: Return the current number of entries in the list.

Syntax : GetEntries (no parameters)

Example: getentries

entries = result

## 1.29 rx\_getfield

GetField

Purpose: Return the contents of a data field.

This data comes from the currently active entry in the list.

Syntax : GetField NUMBER/N

NUMBER/N - The number of the field to return.

The following are the field numbers (using the default names for the fields):

1 - Last Name

2 - First Name

3 - Street

4 - Suburb

5 - Post Code

6 - Phone (entry 1)

7 - Phone (entry 2)

8 - Phone (entry 3)

9 - Phone (entry 4)

10 - Notes

Example: getfield 2

firstname = result

## 1.30 rx\_getfilename

GetFileName

Purpose: Return the filename of the current list.

Syntax : GetFileName (no parameters)

Example: getfilename

file = result

### 1.31 rx\_getextnotedir

GetExtNoteDir

Purpose: Return the current External Note Directory path name.

Syntax : GetNoteDir (no parameters)

Example: getextnotedir

path = result

### 1.32 rx\_getextnotename

GetExtNoteName

Purpose: Every entry that exists in Locate's Main List has its own unique identification number. This number *\*never\** changes. Locate uses this number in conjunction with a standard format to name all its External Note Data Files.

The format is: Text#.ln (where # is the ID number).

There may be instances where you would like to know what the External Note file name for the currently Active entry is. This ARexx command will return the string.

Syntax : GetNoteName (no parameters)

Example: getextnotename

filename = result

### 1.33 rx\_listformat

ListFormat

Purpose: Set the format for the Main List.

Syntax : ListFormat FORMAT/F

FORMAT/F - A set of MUI column format commands.

For some reason only the Column format commands work.

see List Format in the [Config Window](#) .

Example: listformat 'col=0,col=5,col=2'

listformat ',,,'

listformat ''

### 1.34 rx\_load

Load

Purpose: Load a Locate data file.

Syntax : Load FILE,MERGE/S

FILE - Full name and path of the file to be loaded.

MERGE/S - Flag to cause [Merge-Loading](#) .

Example: load ram:YourAddresses.loc merge

---

### 1.35 rx\_notedelete

NoteDelete

Purpose: Delete the current entry's (active) external note data file.

Syntax : NoteDelete (no parameters)

Example: notedelete

### 1.36 rx\_noteedit

NoteEdit

Purpose: Invoke the chosen editor (in configuration) to edit the current (active) entry's external note data file.

Syntax : NoteEdit (no parameters)

Example: noteedit

### 1.37 rx\_noteexamine

NoteExamine

Purpose: Reload the current (active) entry's external note data file.

This is only useful if the Note Window is open.

Syntax : NoteExamine (no parameters)

Example: noteexamine

### 1.38 rx\_phoneactive

PhoneActive ENTRY/N

Purpose: To activate of of the entries in the Phone List.

Syntax : PhoneActive ENTRY/N

ENTRY/N - The number of the pop-list entry to select [1...4].

Example: phoneactive 2

The 'Phone:' data-field will now show the contents of the active entry in the pop-list.

### 1.39 rx\_putfield

PutField

Purpose: Put the supplied text string into a field.

This also simulates the effect of pressing return afterwards.

Syntax : PutField NUMBER/N,VALUE/F/A

NUMBER/N - The field to act on.

The following are the field numbers (using the default names for the fields):

1 - Last Name

---

- 2 - First Name
- 3 - Street
- 4 - Suburb
- 5 - Post Code
- 6 - Phone (Change the phone entry number using the **PhoneActive** command)
- 7 - Notes (is *\*not\** the external note file!)

VALUE/F/A - The string to put in the specified field.

You should use this argument(s) last and use the command 'VALUE' before hand.

The rest of the line after the command is used as the string, so spaces are allowed.

Example: 'putfield 3 value=new street'

This is the standard way of entering Event data via ARexx.

If you wish to insert a new Event then just deselect the list with the 'active noactive' command then proceed to insert data into the fields with the 'putfield' command.

## 1.40 rx\_remove

Remove

Purpose: Remove all currently selected (or single Active) entries from list.

Syntax : Remove (no parameters)

Example: remove

returns the number of entries removed.

## 1.41 rx\_rx

RX

Purpose: Invoke an ARexx script from within Locate.

Syntax : RX FILE,SCRIPT/N/K

FILE - Full name of the file to be executed.

SCRIPT/N/K - The number of a previously installed **ARexx Script** to be executed.

Example: rx 'rexx:test\_script.rexx'

rx script 2

## 1.42 rx\_save

Save

Purpose: Save the list using the current filename.

Syntax : Save (no parameters)

Example: save

### 1.43 rx\_saveas

SaveAs

Purpose: Save the list using a specified filename.

Syntax : SaveAs FILE,REQUEST/S

FILE - The new file name.

REQUEST/S - Flag to cause a file requester to appear.

Example: saveas ram:test request

### 1.44 rx\_scan

Scan

Purpose: Scan the current External Notes drawer for note files.

Syntax : Scan (no parameters)

Example: scan

### 1.45 rx\_search

Search

Purpose: **Search** for a string in the contents of the list.

If a match is found then this list entry will become active.

Syntax : Search CASESENS/S,FIELD/N/K,STRING/F/A

CASESENS/S - Flag to cause a case sensitive (exact case matching) search.

FIELD/N/K - Number to indicate what field(s) to search.

The field numbering matches that used in the **GetField** ARexx command above, but if you wish to include all fields in the search then specify a number of 0.

STRING/F/A - An AmigaDOS search pattern.

Example: search Wils#? casesens field 1

search Wison field 0

### 1.46 rx\_select

Select

Purpose: Cause an entry in the list to become selected (does not activate it).

Syntax : Select ENTRY/N,NOSELECT/S,ALL/S

ENTRY/N - Number of the entry in the list to select.

NOSELECT/S - Flag to deselect ALL entries in the list.

ALL/S - Flag to select ALL entries in the list.

Example: select 10

select noselect

---

## 1.47 rx\_view

View

Purpose: Open and close the external note view window.

Syntax : View STATE/N,TOGGLE/S

STATE/N - A value of 0 will close the window, whereas any other value will open it.

TOGGLE/S - Flag to cause window to toggle between open and closed.

Example: view state 1

---