Appendix B. Programs available for the Run command

The programs listed in this appendix are available for use with the **Run** command (section 3.6). These programs are supported by the Laurel group and are available on the <Laurel> directory on your neighborhood file server.

You may wish to keep these programs on your local disk, or you may retrieve them "on the fly" using the **Run** command. When programs have been retrieved in the latter manner, they are copied to your disk and are left there even after you are finished with them. You may wish to include a RunPath: entry in your Laurel profile to make this process a little simpler (section 5).

Many programs that run via the **Run** command use the composition region of Laurel as a typescript. This typescript is limited to 60,000 characters, i.e., only the last 60,000 characters are retained for any session. When you are finished with such a program, this typescript remains in the composition region, where you may edit it, file it, deliver it, etc. Normally, this typescript replaces the previous contents of the composition region. It may be restored after the **Run** command is finished by invoking U (CANCEL). If you begin another **Run** command immediately after finishing a **Run** command (no editing in between the two **Run**'s), then the typescript (if any) for the subsequent program is appended to the previous typescript (if any).

Many of the programs listed here allow you to invoke other Laurel screen commands. The commands that are typically enabled are **Copy** in the lower menu and all commands in the upper and middle menus except for **Quit** and **Hardcopy**.

Shifted selection is always available as an alternative to type-in. This makes acting on lengthy requests contained in a received message considerably easier.

In the modeless editor, the PASTE key reverts back to a LF key when the **Run** command is executing. It is restored to a PASTE key when the **Run** command finishes.

B.1. Chat

Chat.laurel provides a teletype interface to servers that support the Telnet protocol. Chat.laurel is similar to Chat.run, etc. in its effect, but it does not support the variety of commands available in the other Chat programs. Chat.laurel interacts with you via a typescript in the composition region of the Laurel screen.

Starting and stopping Chat

When you start Chat (by invoking the **Run** command with program name Chat), it greets you with the lines:

> Laurel Chat of *date*
> (CTRL DEL closes connection and returns to Chat command level.)
> C(onnect to), L(ogin to), or Q(uit)?

A blinking caret appears after the question mark, inviting you to begin your Chat session. At this point you are at *Chat command level*. Only the three characters C, L, or Q are valid inputs at this level (upper or lower case does not matter).

A "Q" typed at command level will terminate both the Chat program and the **Run** command.

A "C" typed at command level will produce the prompt:

> Connect to host:

with a blinking caret following the colon, inviting you to type the name of a server machine (terminated with a CR) to which you wish to connect. If you type a DEL character before the CR, then you will return to Chat command level. Once you have connected to the server machine, all interactions are as defined by that server.

An "L" typed at command level will produce the prompt:

> Login to host:

with a blinking caret following the colon, inviting you to type the name of a server machine (terminated with a CR) to which you wish to connect. This command behaves identically to the "C" command, except that "L" also issues an automatic Login command to the server after the connection is established. The name and password used for this Login command are the most recent ones entered via the **User** command (section 3.1.1), or the ones on your local disk if you haven't used the **User** command yet in this Laurel session.

Note: characters are only placed on the screen when they are echoed by the server to which you are connected. Most servers will not echo your password. Also, if the server is heavily loaded, there may be a slight delay between the time that you type characters and the time that they are displayed.

When you are connected to a remote server, if you type CTRL-DEL, then your connection is closed, and you return to Chat command level. If your connection is closed explicitly, e.g., by issuing a "Logout" command to the server, then you will also return to Chat command level. Maxc often takes a significant amount of time to close your connection after you have logged out. Strike CTRL-DEL to notify Chat that you don't wish to wait.

Whenever you return to Chat command level, the one line prompt:

C(onnect to), L(ogin to), or Q(uit)?

is displayed. From this point you may make another connection or quit as described above.

Chat usage tips

The Chat program significantly extends the range of operations you may perform while still inside of Laurel. Some of these are listed here.

Most servers support remote printing in a natural way. IFS servers, for example, support a "Press" command that prints files stored on that IFS on a printer you designate without your having to retrieve those files to your local disk. If you receive a message that mentions the name of a press file stored remotely, using Chat plus shifted selection to type the file name to the "Press" command, is the most convenient way to have that file printed.

Command files of input directed to a remote server are possible using shifted selection from the message display region. Once you SHIFT-select text, all of it will be input to Chat as if you typed it yourself at the appropriate times. Be careful to ensure that you include all CR's, confirmations, abbreviations. etc. just as the server would expect them as if you had typed the text yourself. To learn how to get arbitrary text into your mail file and thus into your message display region, see section B.3, InsertMail.laurel.

B.2. Maintain

Maintain.laurel is useful for interacting with Grapevine servers for various purposes including changing your password, updating public distribution lists, etc. The Maintain program is only useful to you if you are in a registry served by Grapevine (currently only PA and ES, most others will be soon).

This section is an abbreviated guide to the use of Maintain. It contains descriptions of the commands needed by the casual user, i.e., a person who is not an owner of a list or a registry. A more complete explanation of Maintain will appear in the near future. A thorough technical description (not intended for casual users) of the facilities provided by Grapevine is contained in the file [Ivy]<DMS>Interface.press.

Using Maintain

When you start Maintain (by invoking the **Run** command with program name Maintain), it greets you with the lines:

> Grapevine Registration Server Maintenance Program
> Version of *date*
> Login *Name.registry* ...
>
> GV:

A blinking caret appears after the GV:, which is the command prompt, inviting you to type a Maintain command. If you type a question mark, a complete list of commands will be typed out, and the GV: prompt will be issued again. Most of these commands are quite technical; we will deal with only a small subset of them here.

To issue a command to Maintain, you must type only the initial letters that uniquely distinguish that command from the others. As soon as enough of a command word (usually only one letter) has been typed, Maintain will complete that word for you. If you type any subsequent letters in that word beyond the unique prefix, those letters will be used for the rest of the command, which almost always results in an error. In the list of commands that follow, the complete command name will be given first, with the unique letters you should type to issue that command following in parentheses.

Most commands require names as arguments. Maintain will generally suggest a name for each argument based on the names you have entered to previous commands. If the name is correct, just confirm it by typing a space or CR. Otherwise, type in the name (shifted selection works) and terminate with a space or CR (may be included in the shifted selection). All names given to Maintain must be fully qualified, i.e., they must include the registry extension.

If you type a DEL character at any time during command input, that command will be cancelled and a new GV: prompt will be given.

Maintain commands

The following commands may be given in any order, with the exception that you must be properly logged in to give any command other than ?, Login, or Quit. When Maintain is started, it tries to log you in automatically with the current **User** name and password. If that fails, then you may log in using the Login command in Maintain.

The commands listed here are in alphabetical order.

Add Member (AME)

> First type the name of a person, including the *.registry* suffix, to be added to a public distribution list. Maintain will respond with *to group*. Now type the name of the list (remember to include the registry). If you are allowed to add that person to that list (there are several controls on this), then the requested change will be made. If not, Maintain will flash the screen and type out the reason for the rejection. In general, you are allowed to add yourself to general interest lists, but not allowed to add anyone else.

Login (L)

> Maintain will prompt with *Your name please:* . Type your name, including the *.registry* suffix. Maintain will then prompt with *Your password:* . Type your password. Maintain will echo a * for each letter in your password. Grapevine maintains its own copy of your password, and the password you give here must be the one the Grapevine knows. If you give your password successfully, Maintain will respond with *ok*. Once you are logged in, you may proceed to issue other Maintain commands. If you cannot log in, because you have forgotten your password or for some other reason, then you will have to ask an administrator for your registry to issue a Set Password command for you (see below).

Quit (Q)

> This command must be confirmed with a CR or the letter Y. When confirmed, this will terminate Maintain and the **Run** command.

Remove Member (RME)

> First type the name of a person to be removed from a public distribution list. Maintain will respond with *from group*. Now type the name of the list (remember to include the registry). If you are allowed to remove that person from that list (there are several controls on this), then the requested change will be made. If not, Maintain will flash the screen and type out the reason for the rejection. In general, you are allowed to remove yourself from general interest lists, but not allowed to remove anyone else.

Set Password (SP)

> Maintain will prompt with *to be*. Type your new password. Maintain will echo a * for each letter in this password. After terminating the password with a space or CR, Maintain will prompt with *for individual*. Type your name, including the *.registry* suffix. When

you terminate this name with a CR or space, Maintain will store this new password in the Grapevine name data base. From now on, this is the password you must use to access your new mail or to send mail. Be very careful when typing your password. If you make a mistake you may type DEL to cancel this command. Once you type the terminating CR or space for the entire Set Password command you are committed to whatever password you have typed. If you can't reproduce this password later, you will have to obtain the help of an administrator for your registry to correct it.

After setting your password with the Set Password command, you will need to re-invoke the Login command with your new password if you wish to continue using Maintain. Also, when you are back in Laurel, you should re-invoke the **User** command to allow access to your new mail.

Type Entry (TE)

Maintain will prompt with *for R-Name*. Type the name of the individual or public distribution list for which you wish to see information. For individuals, you can discover the current in-box server(s) and forwarding established for that individual. For lists, you will see the purpose of the list (in the Remark: field), the owners of the list (in the Owners: field), and the people who are allowed to add or remove themselves from that list (in the Friends: field).

Type Members (TM)

Maintain will prompt with *of group:*. Type the name of the public distribution list whose members you wish to see. Maintain will list the members of this group, some of which may be lists themselves. To see the members of those lists, extra Type Members commands may be used. Type Members of the special group "Groups.*registry*" produces a listing of all the distribution lists available in *registry*. The Type Members command can only access public distribution lists held in the Grapevine name data base. The **Get** command (section 3.5.1) can access any distribution list to which you can deliver a message.

B.3. InsertMail.laurel

InsertMail is a program that inserts the contents of the message composition region as a message into your current mail file. The message is inserted as is; unlike the **Deliver** command, no From: or Date: fields are added in this process. If you wish to include such fields in the inserted mail, you must edit them in yourself prior to running InsertMail. The T (COM-T) command is useful in preparing a Date: field, should you wish to do so.

InsertMail inserts the message *after* the last selected entry in the table-of-contents. If no messages are selected in the table-of-contents, then the message is inserted at the beginning of the mail file, i.e., it becomes message number 1.

When the message is inserted, the table-of-contents is reformatted, and the newly inserted message is selected and displayed. If the message as inserted does not conform to legal message header rules, or if some header fields are missing, then the table-of-contents display for that entry may have some parts missing or filled in with question marks.

InsertMail does not use the composition region for a typescript. It does its job, terminates, and leaves the text in the composition region as it was.

Uses for InsertMail

Have you ever wanted to annotate or otherwise edit a message in your mail file? The simple way is to move that message into the composition region (**Display** the message, use E (COM-E), and replace the contents of the composition region with a shifted selection of the entire message display region--use multi-clicks to select it.) Then, edit the message and **Run** InsertMail. If you haven't moved the table-of-contents entry, then the new version will be inserted immediately after the original version. If you wish, you may select the original version and invoke **Delete**.

To prepare a command file from which a shifted selection will be taken as input to some other runnable program, just prepare that command file in the composition region and InsertMail it into your mail file. You may wish to prefix the command file message with a portion of a message header, say the Date:, From:, and Subject: fields, to have that command file appear reasonably in the table-of-contents.

To print text that you are preparing in the composition region, InsertMail it into your mail file and invoke **Hardcopy**.

B.4. Files

Files.laurel is a program that provides many of the functions on files that are provided by the Alto Executive.

When you start Files (by invoking the **Run** command with program name Files), it greets you with the lines:

> Laurel File Utility of *date*
> Type ? for help.
> >

A blinking caret appears after the greater-than sign, inviting you to issue a Files command. The ">" characters is the prompt character, and always indicates that Files is ready for another command.

When giving a command to Files, any unique prefix of that command may be typed (as in the Alto Executive). ESC command completion is not supported; typing an ESC will insert that (illegal) character in the command. Terminate command lines with a CR.

Typing CTRL-DEL will cancel commands in progress. This is particularly useful for interrupting long typeout from the List or Type commands.

Most commands allow a file pattern wherever a single filename would be acceptable. A file pattern is expressed using the # and * characters, meaning match exactly one and match zero or more characters respectively, as in the Alto Executive.

Files commands

Copy *NewFile _ FilePattern FilePattern . . .*

> The Copy command is similar to the Copy command in the Alto Executive. The contents of the old files to the right of the arrow will be concatenated and copied into the new file mentioned to the left of the arrow.

Delete *FilePattern FilePattern . . .*

> The Delete command is similar to the Delete command in the Alto Executive. The Delete command requests confirmation for each file before it actually deletes that file. A CTRL-DEL to cancel the entire Delete command will not be acted on until after you finish confirming or cancelling the current file.

FileStat *FilePattern FilePattern . . .*

> The FileStat command is similar to the FileStat command in the Alto Executive. The length and create, read, and write dates for each file specified are listed.

List *FilePattern FilePattern . . .*

The List command is used to list all files on the local disk that match the patterns specified. In the Alto Executive, this function is provided by the TAB character. TAB used in this way is not supported in Files.laurel; use the List command instead. The matching files are listed in the order in which they occur in your directory. Although the output may look alphabetized, that only reflects the periodic sorting done in the Alto directory by the Alto Executive. You may find files out of alphabetical order near the end of your List command output.

Rename *OldFile NewFile*

The Rename command is similar to the Rename command in the Alto Executive.

Quit

The Quit command terminates the Files program and the **Run** command.

Type *FilePattern FilePattern . . .*

The Type command types the contents of the specified files one after the other, without pausing. Each file is preceded by a short length description. You may terminate its output by typing CTRL-DEL.

B.5. SearchMail

The SearchMail.laurel program will search through your entire current mail file, looking for text that matches a pattern. The message number of any message in which a match is found will be displayed in the typescript, and that message is selected in the table-of-contents. When SearchMail is finished, it terminates the **Run** command, leaving in its typescript the message numbers of all messages that contain a match, and with all such messages selected.

The patterns that are allowed are the same as those allowed by the **Find** command (section 4.4.5). Only one pattern is allowed per run of SearchMail; each subsequent run of SearchMail begins a new table-of-contents selection. You must terminate the pattern with an ESC character. Once the pattern is terminated, the search begins.

After SearchMail is finished, you may operate on the set of matched messages in a variety of ways. You may invoke any of the commands in the middle menu to delete, move, or hardcopy the entire set. If the set of selected messages contains two or more messages, then the **Display** command (section 3.2.1) will display each selected message in turn, without destroying the table-of-contents selection.

SearchMail thus provides a rudimentary keyword search capability to Laurel. It is limited to searching for only one pattern (no combinations of patterns are allowed), and it will search within the current mail file only.

B.6. MailFileScavenger

In the unlikely event that your mail file becomes damaged, there is a remedy in the form of the MailFileScavenger.laurel program that usually restores the internal structure of your mail file to a reasonable state. It cannot deal with disk errors; use Scavenger.run first if you suspect disk errors. The MailFileScavenger copies the damaged mail file into a new scratch file as it operates, therefore you must have slightly more free disk pages available for this scratch file than the number of disk pages that your damaged mail file occupies. The MailFileScavenger will warn you if there is not enough room.

To run the MailFileScavenger, just invoke Run with the program name MailFileScavenger. The MailFileScavenger will ask you to type the name of the mail file to be scavenged. Terminate this name with a CR. If you type a name without a period, .mail will be added to the name automatically. MailFileScavenger will proceed to copy your mail into its scratch file (named MailFileScavenger.scratch$). After each fifth message, MailFileScavenger will print out that message number, just to let you know it is still working.

When anomalies are detected in your mail file, MailFileScavenger will print out a short message such as "Message 53: existing count was 231 bytes too small." These messages indicate that the formatting information present in the mail file that Laurel uses to distinguish individual messages was inconsistent with what the MailFileScavenger believes to be distinct messages. When the MailFileScavenger is finished, it is a good idea to check any messages it complained about; these messages may be missing several characters or be malformed in other interesting ways. You should also check neighboring messages--some of the characters in those messages might really be part of other messages.

After the MailFileScavenger has finished copying (and reformatting) your mail into its scratch file, it will pause and ask if it should copy that file back into the original mail file (Type CR or Y to confirm). If there have been few error reports, this may be done without any trepidation; the MailFileScavenger will do the copy, delete the scratch file, and terminate. You may at this point invoke **Mail file** on your mail file once again and gaze upon the damage. On the other hand, if there have been many error reports, you may wish to type a DEL character to deny the automatic copy, and examine the MailFileScavenger.scratch$ mail file before performing the copy yourself. You may invoke **Mail file** on the MailFileScavenger.scratch$ file merely by typing (or SHIFT-selecting) that name into the **Mail file** brackets.

The mail file that MailFileScavenger produces should always give you a readable mail file, i.e., one that Laurel will not complain about. This mail file may have "messages" that are fragments of messages in the original file and/or duplicate messages. As long as Laurel will allow you to read that file, you may repair the damage with the Laurel editor and the InsertMail program (Appendix B.3). If the file produced by the MailFileScavenger is not readable by Laurel, please send a message describing your difficulties to LaurelSupport.PA.