

Getting Started

Summary

1. Introduction

1a. Key Press Notation

2. Starting Up

3. Documentation

3a. Written Documentation

3a1. Unix

3a2. Emacs

3a3. A+

3b. Online Documentation

3b1. Unix

3b2. Emacs

3b3. A+

4. Exiting

4a. Ending a Workstation Session

4b. Ending an A+ Process

4c. Ending an Emacs Session

4d. Ending an Xterm Session

5. The Emacs/A+ Interactive Environment

5a. A+ Tutorials

5b. A+ Scripts

5c. A+ Sessions, Workspaces, and Log Files

5d. The A+ Buffer in Emacs

- 5d1. Setting the A+ Version
- 5d2. Setting the A+ Host Machine
- 5d3. Setting All Emacs Variables for A+ Mode

5e. A Sample A+ Session

1. Introduction

Xterm and Emacs entries, as well as section titles, are enclosed in double quote marks when it helps to distinguish them from the surrounding text. The font in which these tutorials appear on the screen is in fact an APL font (kaplgallant); the distinction between ordinary text and A+ expressions is usually clear from the context and the use of special symbols.

It is possible that at some point, as you work through this tutorial, you will find yourself out of step with the text. Most likely, you will be in an Emacs session at the time. If that happens, exit from the Emacs session and go back to a point in the tutorial where you can start over. The tutorial is not so long that this will be burdensome, and it is definitely worthwhile to complete it successfully.

1a. Key Press Notation

We will often use the notation Ctrl-<key>, which means press and hold the Control key and, while holding, press the key specified by <key>. For example, Ctrl-x means press and hold the Control key, and then press the x key. Similarly, Meta-<key> means to press and hold the Meta key, which is marked with a diamond, or "Left", or "Right." If you are using an IBM keyboard, use Alt for Meta.

If the dash is omitted, as in Esc x, press and release the first key (here Escape), and then press the the other key (here x).

For the mouse, to "click" a button means to depress it and release it quickly. To "press" a button means to depress it and hold it until you are told to release it. To "double-click" is to click twice in quick succession.

2. Starting Up

The purpose of this tutorial is to help people unfamiliar with the A+

system get up and running.

We are interested in two application windows in this tutorial. One is an Xterm. This is an interactive window for executing Unix commands. If you see an "Xterm" icon on the lower left, double-click on it and an Xterm window will open. Or, move the mouse pointer to the background area, and then press and hold the right mouse-button to see the main menu of applications. Move the pointer to Xterm and release the button, and an Xterm window will open.

You can start an A+ session in an Xterm session by entering:

```
/usr/local/bin/a+
```

Even though A+ started in an Xterm can be used interactively, the preferred use is to run applications. Interactive use of A+ is best done in Emacs.

The Emacs editor window is the other application window of interest here. Once again, bring up the main menu of applications as you did for Xterm. Only now, move the pointer to Emacs/A+ and release the button. You can also start an A+ session within Emacs. See "5. The Emacs/A+ Interactive Environment".

If you now have both an Xterm and Emacs window on your screen then most likely one overlaps the other.

3. Documentation

3a. Written Documentation

3a1. Unix

Unix is the operating system in which we work, and as an A+ programmer you will most likely have to know the rudiments of the system. The following reference is recommended:

The UNIX Programming Environment, by Kernighan and Pike

Prentice-Hall Software Series

It is also possible to display online documentation for Unix; see "3b.Online Documentation".

3a2. Emacs

The Emacs editor, which serves as the A+ application development environment, is described in:

GNU Emacs Manual, by Richard Stallman

Free Software Foundation

or

Learning GNU Emacs, by Cameron and Rosenblatt

O'Reilly and Associates, Inc.

3a3. A+

3b. Online Documentation

3b1. Unix

There is substantial online documentation for Unix. To view the documentation on a particular topic from within an Xterm, enter the command:

```
man topic
```

(man stands for manual pages). For example, information on the Unix command cp (for copying files) can be viewed as follows:

```
man cp
```

A formatted description of the topic will be displayed in the Xterm window. Press Ctrl-c to quit the display. To learn more about manual page command, enter:

```
man man
```

Another way to view manual pages is in a separate window, that can be initialized from an Xterm by entering

```
xman &
```

(See the command Emacs & in "6c. More on Xterm" for the meaning of &.) The effect of this command is to display a new window. Click on the "Manual Page" button in that window and a manual-page display will

appear, with help information in it. Press and hold down the "Options" button, drag the mouse pointer to search, and release. Enter the topic whose manual pages you want to see, and press the Enter key.

This display can also be obtained by pressing the right mouse-button while the pointer is on the background area, moving the pointer to "Tools" and then to the right and then down the submenu that appears to "Manual Pages", and releasing the button.

If for some reason you want a print out of the online documentation, e.g. on the Unix command `cp`, enter the command:

```
man -t cp
```

The output should automatically be printed on the default local printer in your area (see "6c. More on Xterm" regarding setting the printer name).

3b2. Emacs

Emacs has an online tutorial and online help. Press `Esc x`, which will move the keyboard cursor to the so-called minibuffer (the last line in the Emacs window) with the prompt `M-x`. Enter `help-` and press the space bar. That is, at the point where you should press the space bar, the minibuffer should look like:

```
M-x help-
```

After you press the space bar you will see displayed the full names of both the tutorial and online help. Complete the name of the one you want and press the Enter key.

3b4. The Window Manager

The windows on your display are under the control of a program known as a window manager. T

4. Exiting

4a. Ending a Workstation Session

You can either log off the workstation, so that the next time you will have to log on again and start from scratch, or you can lock the

workstation, which means that next time you simply unlock it and start where you left off.

4b. Ending an A+ Process

To end an A+ process without ending the Xterm or Emacs session in which it is running, enter:

```
$off
```

The next time you start an A+ process, the same log file (see "4c. A+ Sessions, Workspaces, and Log Files") will be active. If you are also going to end the Xterm or Emacs session, you may want to save the log before you end the A+ process.

4c. Ending an Emacs session

To end an Emacs session, press Ctrl-x followed by Ctrl-c. You will be prompted in the minibuffer to save any files that have been modified since they were last saved. In particular, if you have run A+ in this session you will most likely be asked whether or not your A+ session log should be saved:

```
Do you want to save .emacs_a? (Y/N)
```

You may want to answer Y to keep this log, at least for a while. If you answer N to any of these prompts, you will then be prompted as follows:

```
Modified buffers exist, do you really want to exit? (yes or no)
```

Answer yes. If you did not end your A+ session (see "4b. Ending an A+ Process"), you will then be prompted with:

```
Active processes exist; kill them and exit anyway? (yes or no)
```

Answer yes, and the Emacs window will disappear.

For a fast exit with no prompting, put the mouse pointer on the title bar at top of the Emacs window and hold down the right mouse-button. Move the pointer to "Quit" and release.

4d. Ending an Xterm session

To end an Xterm session, move the mouse pointer to that window, and press Ctrl-d. Or, as with Emacs, put the mouse pointer on the title bar at the top of the Xterm window and hold down the right mouse-button. Move the pointer to Quit and release.

5. The Emacs/A+ Interactive Environment

The Emacs editor serves as the application development environment for A+. Bring up a new Emacs window, as described in the introduction. The line next to the bottom line contains "Emacs: *scratch*". The area above this line is called a buffer. The title of this buffer is "*scratch*". The bottom line of the window is called the minibuffer.

Press the function key F4. The Emacs window should split into two buffers. The top buffer is now the scratch buffer, while the bottom one is an A+ session with the buffer title "*a*".

The keyboard cursor is the solid, character-sized rectangle that indicates where the next typed character will go. The buffer containing this cursor is called the "selected" buffer. Pressing the F6 key (see "5b. More on Emacs for information on the F5, F6 and F7 keys") repeatedly will toggle back and forth between the two buffers, first selecting one and then the other. Moving the mouse pointer into a buffer and pressing the left mouse-button will also select that buffer.

After you have pressed F4, the bottom buffer holds an interactive A+ session. Select this buffer and you can enter A+ expressions for evaluation. If you were developing an A+ application then typically you would select the top buffer and open an A+ script file for editing. Or, you might select an A+ tutorial instead.

To open a file in the selected buffer, press Ctrl-x followed by Ctrl-f. The minibuffer will automatically be selected and will contain the prefix "Find file: ~/". To open a file from your HOME directory, simply enter its name and press the Enter key. To open a file in a subdirectory of your HOME directory, say the script "filetest.+" in the directory "versiontests", enter "versiontests/filetest.+". To open a file in any other directory, enter its full path and name, starting with /. The initial / causes Emacs not to include in the path the directory it identified in its prompt. For example, to open this tutorial, enter

```
(Install_Directory)/doc/tutorials/GettingStarted
```

If the file you name is found, it will be opened and displayed in the top buffer. Otherwise, the top buffer will appear to display an empty file and, in addition, the message "(New File) "will be displayed in the minibuffer. In the latter case, the file will not be created at this point; it will be created if and when you save it for the first time.

Use the Home, PgUp, PgDn, arrow, and End keys to scroll through an open file.

It is also possible to open a directory, in order to see the names of the files it contains. Once again, press Ctrl-x followed by Ctrl-f, but in response to the prompt Find file: ~/ , enter the directory name. For example, to open the tutorial directory, enter

```
(Install_Directory)/doc/tutorials
```

You should then see a list of file names in the current buffer. Any file in the directory can now be opened simply by moving the keyboard cursor onto the file name and pressing the f key.

Both A+ scripts and the tutorials contain executable A+ expressions and function definitions. An A+ expression that stands alone on a line can be brought into the A+ buffer and executed simply by placing the keyboard cursor on the expression and pressing F2. (Once the top buffer is selected, the keyboard cursor can be moved to the line holding the expression with the arrow keys; or you can put the mouse pointer on the expression and press the left mouse button.) Note that pressing F2 also causes the keyboard cursor to advance one line, leaving you in position to bring the next line into the A+ session with F2.

To bring an entire function definition into the A+ session, place the keyboard cursor anywhere on the definition and press F3. (If the definition has only one line then F2 will do as well as F3.) Unbalanced punctuation anywhere in the buffer above the function definition, however, will prevent the function's being brought in.

5a. A+ Tutorials

The A+ tutorials are located in (Install_Directory)/doc/tutorials, where, for instance, you will find the online version of this tutorial:

By default the Install_Directory is /usr/local/aplus-fsf-x.xx
(where x.xx is version.release)

i.e. /usr/local/aplus-fsf-4.20/doc/tutorials/GettingStarted

Open the file for this tutorial in the top buffer and use the
down-arrow key to scroll down to this point in the file. You can then
bring the following examples into your A+ session using F2.

```
↳a↵3 5
@a
g{x}:x*2
g a
```

You will notice that when the first, second, or fourth lines are
executed, the result is displayed immediately below (not shown here).
The third line holds a function definition, and there is no result to
display when the line is executed. It is sometimes convenient to
display the result of an execution in the tutorial; you shouldn't
execute these lines. In general, the A+ expressions that should be
executed are indented like those above.

5b. A+ Scripts

A+ scripts define applications, toolkits, and utilities. They consist
of global variable and function definitions, as well as expressions
for initialization.

Scripts are loaded into A+ sessions using the \$load system command or
the _load system function. For example, once an A+ session is begun,
if you plan to use the screen management facilities, then load them as
follows:

```
$load s
```

Once a script is loaded, all the functions and global variables
defined in it are available, and all the executable statements it
contains have been executed. Scripts can themselves include \$load
commands and _load statements, so that applications can load the
toolkits and utilities they depend on, and need not incorporate them
directly in their scripts.

The effect of \$load is by and large the same as if the script file
were opened in a buffer, the keyboard cursor were placed on the top
line, and F2 were held down until every line was brought into the A+

session. (See the A+ Reference Manual for the differences.) In particular, script files are processed line-by-line, starting at the top.

When developing and maintaining an application you will usually open its A+ script in the top buffer and selectively bring parts of it into the A+ session using the F2 and F3 keys. F2 can also be used to bring a multi-line function definition into the A+ session, but line by line. This is particularly useful during debugging. Start with the keyboard cursor on the first line of a function definition and press F2 for each line. (Or, simply hold F2 down and it will automatically go from one line to the next.) As each line is brought in, its display in the A+ session has a prefix of *'s on the left, one for each level of unbalanced, nested parentheses, braces, or quotation marks (but not brackets).

For example, consider the function definition

```
penny x:{
  m←0.01;
  i←1;
  while (m<x) {
    ↓i,m;           Ⓚ Print the current values of i and m.
    i←i+1;
    m←m×2;
  };
}
```

If you place the keyboard cursor anywhere in the definition and press F3, the function will be loaded into the A+ session, with the message:

```
Ⓚ Loading: penny x
```

However, if you place the keyboard cursor on the top line of the definition and press F2 on each line, the following lines will appear in the A+ session:

```
      penny x:{
*    m←0.01;
*      i←1;
*      while (m<x) {
**      ↓i,m;           Ⓚ Print the current values of i and m.
**      i←i+1;
**      m←m←2;
**    };
*    }
```

Careful attention to the increasing and decreasing numbers of *'s on the left will usually reveal unbalanced parentheses, braces, or quotation marks that span more than one line.

5c. A+ Sessions, Workspaces, and Log Files

A+ sessions, whether running in an Xterm or Emacs, are interactive sessions in which you can enter expressions directly and see the results. In an Emacs session, you will often bring pieces of an application into the A+ session with the F2 and F3 keys and then select the A+ buffer to enter test expressions.

Any objects created during an A+ session are stored in the A+ workspace. Although the workspace starts out with a fairly limited amount of space, you can increase it as necessary. For example, start up an A+ session and enter:

```
ρ21000 1000 ␣ You type Meta-r Meta-i 1000 1000
```

You will see:

```
4000056: wsfull
*
```

The first line gives the number of bytes required to execute the expression, followed by a workspace full message, meaning that the workspace does not have the reported amount of space available. In Version 3 it is "␣[error] 4000056: wsfull". The * on the second line indicates suspended execution. (Asterisks for suspended execution and for unbalanced punctuation are intermixed: there is no visible difference between them.) Enter the left-pointing arrow (Meta-[) to the right of the star in order to resume execution:

```
* ← ␣ You type Meta-[
and you will see
```

```
1000 1000
```

which is the result of the expression. The workspace size has been increased and execution of the expression ρ21000 1000 has been completed.

If you are running A+ in an Emacs session, then everything you type and everything the A+ process reports is kept in a session log, which

can be saved at the end of your Emacs session -- or indeed, at any time. It will be continued in your next A+ session.

Both F2 and F3 can be used within the session log also. That is, you can use the Page Up and up-arrow keys and the mouse pointer and left button to place the cursor on a line, then either bring it forward for reexecution using F2 or redefine a function using F3 (or newly define a function, if, after the selected line was originally executed, the function was expunged or the A+ process was ended and begun anew).

If you simply press the Enter key on a line in the session log, that line will be brought forward but not executed. You can then just press the Enter key to execute it or, what is more usual, edit it and then press Enter. This is a very efficient way to try out a series of variations on a single expression.

You may find yourself first editing a line in the session log and then bringing it forward for execution. If you do this the session log will contain the edited line, not the original, so this practice must be avoided if you want a clean record of your work.

5d. The A+ buffer in Emacs

You can control the version of A+ that you run, the host machine on which it runs, and various other aspects of the A+ mode in Emacs by setting the appropriate Emacs variables. In general, there are two ways to do this (both within Emacs, of course). One way is to set the A+ mode variables individually by executing Emacs-LISP expressions and the other is to run an application that displays all the A+ mode variables with their current settings and provides a way for them to be changed. Any changes made in these ways will remain in effect for the duration of the current Emacs session.

It is rare to change anything other than the A+ version and host machine. Let's see how to change their Emacs settings by executing Emacs-LISP expressions. Another way to change these settings is described in "5d3. Setting All Emacs Variables for A+ Mode".

Start by pressing the Esc key twice. The cursor will move to the minibuffer line and will be positioned to the right of the prompt "Eval: " -- if you see any other prompt, press the space bar and Eval: should then appear. At this point, you are ready to enter Emacs-LISP expressions for evaluation, as described in the following two sections.

5d1. Setting the A+ Version

One way to specify the version of A+ you want to run is to place the cursor in the minibuffer with the "Eval: " prompt and enter, for example:

```
Eval: (setq a-prog "/usr/local/bin/a+")
```

5d2. Setting the A+ Host Machine

By default, an A+ process running under Emacs runs on the same host machine that is running Emacs. To specify the host machine on which A+ is to run, make sure the cursor is in the minibuffer with the "Eval: " prompt. Enter, for example:

```
Eval: (setq a-host "oba")
```

oba is the name of a workstation.

5d3. Setting All Emacs Variables for A+ Mode

To see all the A+-mode Emacs variables and possibly change their values, start by pressing Esc x. The cursor will move to the minibuffer and will be positioned to the right of the command prompt M-x. Enter "a-options", so the line looks like

```
M-x a-options
```

and press Enter. A new Emacs buffer entitled "*'a' Options*" will appear with a display of the A+-mode Emacs variables. Each variable takes up a three-line paragraph in the display, and the paragraphs are separated by empty comment lines (";;" at the beginning of the line, followed by blanks).

Each three-line paragraph starts with a line containing ";;" followed by the name of the A+-mode Emacs variable and ended by a colon. The second line shows the current value of the variable and the third line is a brief description of what it does. For example, the paragraph for a-prog, which defines the A+ version to be loaded when F4 is pressed, might look like:

```
;; a-prog:  
"/usr/local/bin/a+"
```

*Program to run for 'a' command

```
;; a-host:
  nil
*Host on which to run 'a'
```

("nil" represents the default machine on which to run A+, the local workstation). A discussion of the other A+-mode variables can be found in the A+ Reference Manual. Here it is enough to consider only a-prog and a-host. To change the value of either one, say a-prog, move the keyboard cursor to any line in its defining paragraph and press the s key. The minibuffer will be selected, where you will see the prompt "New a-prog: ". Enter the new name (including the double quotes around it) and return, and the a-options buffer will automatically reflect the change.

In navigating in the *'a' Options* buffer, you can press the n key to move to the next paragraph and the p key to move to the previous one. Press the q key at any time to end the a-option program.

5e. A Sample A+ Session

Let's start with a clean slate. If you have an active A+ process in an Emacs session, select the A+ buffer and enter

```
$off
```

to end the session. Then press F4 to start a new session. The A+ buffer should still be selected, and the keyboard cursor should be indented 5 spaces from the left border. Type in the following to make sure A+ is working, and press the Enter key.

```
  10 0 You type Meta-i 10
0 1 2 3 4 5 6 7 8 9
```

The line 0 1 2 3 4 5 6 7 8 9 that appeared after you pressed the Enter key is the result of executing the expression 10, which you typed in. We will continue to show both what you type and how the A+ process responds.

Type in the following square-root function definition:

```
sqrt x: x*0.5 0 You type this
```

See that the name of the square-root function is in the functions list:

```
$fns  $\square$  You type this
sqrt
```

Test the sqrt function:

```
sqrt 1 4 9 16 152399025  $\square$  You type this
1 2 3 4 12345
```

Type the following to produce an error and its resulting suspension:

```
sqrt ^2  $\square$  You type this (Meta-2 for ^)
*: domain
*
```

The * in the line " *: domain" indicates that the error occurred during the execution of the primitive power function denoted by *. In Version 3, this message is " \square [error] *: domain". The * standing alone in the next line indicates that execution is suspended. This is in the prompt area; the keyboard cursor should be on the same line. Type \$si on that line:

```
* $si  $\square$  You type this
sqrt ^2
.sqrt: x*0.5
*
```

\$si is called the state indicator. It shows the top-level expression that caused the suspension (sqrt ^2), and on the next line, the expression within the sqrt function where the problem occurred. In Version 3, a blank line appears after the display of the state indicator. Once again, a * appears to the left of the indented keyboard cursor because of the suspension. Execution can be abandoned, clearing the suspension, by entering a right pointing arrow:

```
*  $\rightarrow$   $\square$  You type Meta-]
```

You should now see the keyboard cursor indented with no * to the left. Moreover, if you check the state indicator you will see that it is clear:

```
$si  $\square$  You type this
 $\square$  Nothing is displayed
```

Sign off A+, and you will see a message indicating you have done so:

```
$off ↵ You type this  
Process 'a' finished
```