MacABC

The ABC interpreter & environment on the Apple Macintosh

(This document uses the fonts Chicago 12, New York 9 and 10 and Monaco 9.)
(Do not use Font Substitution on a LaserWriter.)

ABC is a powerful, easy-to-learn and easy-to-use interactive programming language, intended for personal computing and designed as a modern alternative to BASIC and an easy alternative to Pascal.

ABC is a progamming language and environment. This environment shows one face tot the user. You are always in the editor, so you can edit immediate commands and program input in exactly the same way that you edit commands and functions.

The current implementation for the Apple Macintosh is a rather straigthforward port of the Unix/MS-DOS version. This means that the user interface does not completely follow Macintosh standards. Also, the ABC philosophy sometimes collides with the Mac Guidelines. Some of these differences are explained below.

1. ABC WORKSPACES AND MACINTOSH FILES

Startup

When MacABC itself is opened from the Finder (by Double-Clicking it or by selecting it and choosing Open from the File-Menu) the ABC interpreter is started, using the ABC editor, in the last workspace used or in workspace 'first' if this is your first abc session. A workspace is kept as a group of files in a Macintosh Folder, with separate files for each how-to and location. The workspace folders themselves are kept by default in the folder 'abc' on the same disk as MacABC.

To use a group of workspaces in a different folder (e.g. from another floppy) open the file 'wsgroup.abc' in that folder. Opening (almost) any other MacABC file in a particular workspace folder, starts in that workspace instead of the last one used.
However, opening a file that holds the definition of a how-to without parameters makes the ABC interpreter execute the command that the how-to defines. Input for READ commands is taken from the keyboard. When the execution is finished MacABC quits to the Finder. (The name of a file holding a how-to definition ends in '.cmd').

Workspaces

To create a new workspace, or go to an existing workspace, type '>name'. To go to the last workspace you were in, type a single '>'. To get a list of workspace names, type '>>'.

Files

The following files are created by MacABC to hold ABC objects, and for internal administration.

copybuf.abc	copy buffer between sessions
wsgroup.abc	table mapping workspace names to folder names
position.abc	focus position of edited how-to's in workspace
perm.abc	table mapping object names to file names
suggest.abc	suggestion list for user-defined commands
types.abc	table with codes for typechecking between how-to's

*.cmd	command how-to's in this workspace
*.zfd, *.mfd, *.dfd	function how-to's in this workspace
*.zpd, *.mpd, *.dpd	predicate how-to's in this workspace
*.cts	permanent locations in this workspace
@p@r@l@o@c@a@b@c@	temporary file for printing locations

Special Tasks

The File menu differs from the Macintosh standard. It is used to provide functions that other ABC implementations invoke from the command-line.

The first three items provide an interface between ABC objects and Macintosh files from other applications. The next four enable you to print ABC objects. Then there are two items that enable you to recover from a machine crash. Finally, the only standard thing on the File menu is that the last item can be used to end your ABC session.

FILE MENU SUMMARY

Open TEXT file as table
  Fill an ABC table with text lines from any Macintosh TEXT file.
Save table as TEXT file
  Write text lines from an ABC table to a MacWrite TEXT file.
Save how-to's as TEXT file
  List the how-to's from the current workspace on a MacWrite TEXT file.

Page Setup
  As in any Mac application ...
Print location
Print how-to
Print workspace how-to's
  Printing ...
    You can also print ABC files from the Finder by selecting them and choosing Print from the Finder's FILE menu.

Recover current workspace
    Recover the current workspace when its index is lost: useful after a machine crash if the ABC internal administration files didn't get written out.
Recover workspace-group index
   Recover the index of a group of workspaces.

Quit
   You can also type 'QUIT' to finish an ABC session.

2. THE ABC EDITOR

Visit

To create a new how-to, just type the first line of the how-to. This creates the new how-to, and allows you to type the body. Use 'Exit' from the File menu to finish it (or use the [Enter] key).

To visit a how-to, type a colon, followed by the name of the how-to. Again, use 'Exit' to exit. To visit the last how-to again, or the last how-to you got an error message for, type a single ':'. To get a list of the how-to's in this workspace, type '::'.

To edit a location, type a '=' followed by the name of the location. To re-edit it, type a single '='. To get a list of the locations in the workspace, type '=='.

One of the differences in philosophy between the Mac and ABC concerns saving: once you finish editing a how-to definition or a location with 'Exit' the changes are written to the file; you will never be asked for confirmation.

Editor Usage

(This is necessarily a short description; see "The ABC Programmer's Handbook" for full details.)

When MacABC starts up using the ABC editor, it displays a prompt '>>> ?', and awaits input.

The ABC editor has a FOCUS, roughly resembling a text-selection in Macintosh applications: it is the part of the edited text this is highlighted. The blinking cursor is at the start of the focus. An 'empty' focus is shown as a question mark, signalling that text can be added here; this is called a HOLE.

TYPING AND SUGGESTIONS: as you type, the system tries to suggest a possible continuation for what you have typed; to accept the suggestion, press [accept] (by default this is bound to the [Tab] key; see table below). If you don't want to accept the suggestion, just carry on typing (you can always type character for character, ignoring the suggestions).
Usually the system knows where a letter must be capital and where not, and you usually don't have to use the shift key; however, in the few places where both a lower-case and an upper-case letter would be legal (for instance for AND), you have to type the letter upper-case.

When you type a control command, like WHILE, the system provides indentation automatically for the body of the command; to reduce the indentation one level, type [return].

In contrast with the Macintosh-style of text editing the ABC editor always inserts text *before* the focus; also, you must explicitly use the [delete] operation to dispose of text.

CORRECTING AND EDITING: the [undo] key (by default bound to backspace) undoes the last key you typed. Repeatedly typing it undoes more and more, up to a certain maximum number of keypresses.

To correct other parts, you must put the 'focus' onto the part you want to change. The focus is displayed by highlighting. [Widen] and [extend] make the focus larger, [first] and [last] make it smaller.

COPYBUFFER: [Copy] copies the contents of the focus to a buffer, or if the focus is not focussed on anything, copies the contents of the buffer back to where you are positioned.
[Delete] deletes the contents of the focus; if the copybuffer is empty the deleted text is put in the buffer.

The ABC copybuffer is not transferred to the Macintosh Clipboard.

MOVING THE FOCUS: [Upline] and [downline] focus on one line above or below. [Previous] and [next] move the focus left and right. [Up], [down], [left], and [right] move an empty focus around. [Goto] widens the focus to the largest thing at the current position.

OTHER OPERATIONS: [record] records all keystrokes until the next time you press [record] - [play] replays them. [Redo] redoes the last key(s) undone; [interrupt] interrupts a running command.

SUMMARY OF EDITING OPERATIONS

NAME	DEFAULT KEYS*	SHORT DESCRIPTION
Accept	[Tab]	Accept suggestion, focus to hole or end of line
Return	[Return]	Add line or decrease indentation
Widen	[Cmd-W]	Widen focus
Extend	[Cmd-E]	Extend focus (usually to the right)
First	[Cmd-F]	Move focus to first contained item
Last	[Cmd-L]	Move focus to last contained item
Previous	[Cmd-P]	Move focus to previous item
Next	[Cmd-N]	Move focus to next item
Upline	[Cmd-U]	Move focus to whole line above
Downline	[Cmd-D]	Move focus to whole line below
Up	[UpArrow]	Make new hole, move up
Down	[DownArrow]	Make new hole, move down
Left	[LeftArrow]	Make new hole, move left
Right	[RightArrow]	Make new hole, move right
Goto	[mouseclick]	New focus at cursor position
Undo	[Cmd-Z], [Backspace]	Undo effect of last key pressed (may be repeated)
Redo	[Cmd-U]	Redo last UNDOne key (may be repeated)
Copy*	[Cmd-C], [Cmd-V]	Copy buffer to hole, or focus to buffer
Delete*	[Cmd-X], [Clear]	Delete contents of focus (to buffer if empty)
Record	[Cmd-R]	Start/stop recording keystrokes
Play	[Cmd-T]	Play back recorded keystrokes
Help	[?]	Print summary of keys not shown on menus
Exit	[Enter]	Finish changes or execute command
Interrupt	[Cmd-.]	Interrupt command execution

* Notes:

[Cmd-X] means: hold the [Command] key (with the Clover-leaf symbol or an Apple) down while pressing x.

WARNING: To support Desk Accessories, the Edit-menu follows Apple's Guidelines as far as the names of the menu items is concerned. However, Copy and Paste actually invoke ABC's Copy operation; Cut and Clear invoke ABC's Delete. Here the ABC philosophy differs from the Mac Guidelines; see the explanation above.

The binding of editing operations to keys may be changed with a resource editor like ResEdit. This will be reflected in the menus. Do not change the Command-key equivalents for items in the Edit-Menu, however, since you might invalidate support for some Desk Accessories.

3. THE ABC INTERPRETER

Once the ABC interpreter is running, only the Pause menu is active. Pressing the mousebutton in its title temporarily stops execution (enabling you to read fast scrolling output). Choosing [interrupt] stops execution and returns to the ABC Editor. You can also [interrupt] a READ command in this way.

An overview of the ABC language proper is given in the Quick Reference at the end of this document.

4. THE MAC DISTRIBUTION

The ABC distribution for the Macintosh should contain at least four files:

- the application 'MacABC' itself;
- the TEXT file 'MacABC.help' that is used by MacABC when you invoke the Help-menu; keep this file in the same folder as MacABC or in your System Folder;
- the MacWrite file 'MacABC.doc' that contains this text;
- the MacWrite file 'ABCintro.doc' with a short introduction to ABC.

On a Hard Disk it is probably best to create a folder 'abc' (not in any other folder) and to copy these files to that folder.

Note for Macintosh guru's: MacABC uses 'Monaco 9' for the screen and 'Courier 10' for printing. You can change that easily with ResEdit, by editing the resource with type 'Conf' and ID '0'. The horizontal and vertical window-size and the window-title can also be adapted. To facilitate this, first Paste the 'TMPL' resource with ID '5189' named 'Conf' from MacABC to (a copy of) ResEdit. Beware that MacABC only works properly with Fixed-Width Fonts like Monaco and Courier.

5. EPILOGUE

See Also

Leo Geurts, Lambert Meertens and Steven Pemberton,
The ABC Programmer's Handbook, Prentice-Hall,
Englewood Cliffs, New Jersey, 1989, ISBN 0-13-000027-2.
Steven Pemberton,
An Alternative Simple Language and Environment for PC's,
IEEE Software, Vol. 4, No. 1, January 1987, pp. 56-64.
The ABC Newsletter. To keep in touch; available free from CWI.

Address

ABC Distribution, CWI/AA, Postbox 4079, 1009 AB Amsterdam, The Netherlands.
E-mail: abc@cwi.nl.

Authors

Frank van Dijk, Leo Geurts, Timo Krijnen, Lambert Meertens, Steven Pemberton,
Guido van Rossum.

ABC QUICK REFERENCE

COMMANDS

WRITE expr	Write to screen;
	/ before or after expr gives new line
READ address EG expr	Read expression from keyboard to address;
	expr is example
READ address RAW	Read line of text
PUT expr IN address	Put value of expr in address
SET RANDOM expr	Start random sequence for random and choice
REMOVE expr FROM list	Remove one element from list
INSERT expr IN list	Insert in right place
DELETE address	Delete permanent location or table entry
PASS	Do nothing
KEYWORD expr KEYWORD ...	Execute user-defined command
KEYWORD	Execute refined command
CHECK test	Check test and stop if it fails
IF test:	If test succeeds, execute commands;
commands	no ELSE allowed
SELECT:	Select one alternative:
test: commands	try each test in order
...	(one must succeed;
test: commands	the last test may be ELSE)
WHILE test:	As long as test succeeds
commands	execute commands
FOR name,... IN train:	Take each element of train in turn
commands	

HOW-TO's

HOW TO KEYWORD ...:	Define new command KEYWORD ...
commands	
HOW TO RETURN f:	Define new function f with no arguments
commands	(returns a value)
HOW TO RETURN f x:	Define new function f with one argument
commands	
HOW TO RETURN x f y:	Define new function f with two arguments
commands	
HOW TO REPORT pr:	Define new predicate pr with no arguments
commands	(succeeds/fails)
HOW TO REPORT pr x:	Define new predicate pr with one argument
commands	
HOW TO REPORT x pr y:	Define new predicate pr with two arguments
commands	
SHARE name,...	Share permanent locations
	(before commands of how-to)

Refinements (after the commands of a how-to)

KEYWORD : commands	Define command refinement
name: commands	Define expression- or test-refinement

Terminating commands

QUIT	Leave command how-to or command refinement, or leave ABC
RETURN expr	Leave function how-to or expression refinement, return value of expr
REPORT test	Leave predicate how-to or test-refinement, report outcome of test
SUCCEED	The same, report success
FAIL	The same, report failure

EXPRESSIONS AND ADDRESSES

666, 3.14, 3.14e-9	Exact constants
expr,expr,...	Compound
name,name,...	Naming (may also be used as address)
text@p	"ABCD"@2 = "BCD" (also address)
text|q	"ABCD"|3 = "ABC" (also address)
text@p|q	"ABCD"@2|1 = "BCD"|1 = "B"
table[expr]	Table selection (also address)
"Jan", 'Feb', 'Won''t!'	Textual displays (empty: "" or '')
"value = `expr`;"	Conversion of expr to text
{1; 2; 2; ...}	List display (empty: {})
{1..9; ...}, {"a".."z"; ...}	List of consecutive values
{["Jan"]: 1; ["Feb"]: 2; ...}	Table display (empty: {})
f, f x, x f y	Result of function f (no permanent effects)
name	Result of refinement (no permanent effects)

TESTS

x < y, x <= y, x >= y, x > y	Order tests
x = y, x <> y	(<> means `not equals')
0 <= d < 10	
pr, pr x, x pr y	Outcome of predicate pr (no permanent effects)
name	Outcome of refinement (no permanent effects)
test AND test AND ...	Fails as soon as one of the tests fails
test OR test OR ...	Succeeds as soon as one of the tests succeeds
NOT test	
SOME name,... IN train HAS test	Sets name, ... on success
EACH name,... IN train HAS test	Sets name, ... on failure
NO name,... IN train HAS test	Sets name, ... on failure

PREDEFINED FUNCTIONS AND PREDICATES

Functions and predicates on numbers

~x	Approximate value of x
exactly x	Exact value of x
exact x	Test if x is exact

+x, x+y, x-y, -x, x*y, x/y	Plain arithmetic
x**y	x raised to the power y
root x, n root x	Square root, n-th root
abs x, sign x	Absolute value, sign (= -1, 0, or +1)
round x, floor x, ceiling x	Rounded to whole number
n round x	x rounded to n digits after decimal point
a mod n	Remainder of a on division by n
*/x	Numerator of exact number x
/*x	Denominator
random	Random approximate number r, 0 <= r < 1
e, exp x	Base of natural logarithm, exponential function
log x, b log x	Natural logarithm, logarithm to the base b
pi, sin x, cos x, tan x, arctan x	Trigonometric functions, with x in radians
angle (x, y), radius (x, y)	Angle of and radius to point (x, y)
c sin x, c cos x, c tan x	Similar, with the circle divided into c parts
c arctan x, c angle (x, y)	(e.g. 360 for degrees)
now	e.g. (1999, 12, 31, 23, 59, 59.999)

Functions on texts

t^u	t and u joined into one text
t^^n	t repeated n times
lower t	lower "aBc" = "abc"
upper t	upper "aBc" = "ABC"
stripped t	Strip leading and trailing spaces from t
split t	Split text t into words

Function on tables

keys table	List of all keys in table

Functions and predicates on trains

#train	Number of elements in train
e#train	Number of elements equal to e
e in train, e not.in train	Test for presence or absence
min train	Smallest element of train
e min train	Smallest element larger than e
max train, e max train	Largest element
train item n	n-th element
choice train	Random element

Functions on all types

x<<n	x converted to text, aligned left in width n
x><n	The same, centred
x>>n	The same, aligned right

THE CHARACTERS

!"#$%&'()*+,-./0123456789:;<=>?	This is the order of all characters
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_	that may occur in a text.
`abcdefghijklmnopqrstuvwxyz{|}~	(The first is a space.)