HyperMacro 1.1

A HyperCard Scripting Utility

by

David P. Sumner

HyperMacro

Introduction
The HyperMacro INIT installs patches into the system that make it far easier to write HyperCard scripts. After installing HyperMacro, you will discover that you need much less effort to write HyperCard scripts. HyperMacro is intelligent and automatically responds to certain actions. For example, if you type

```
function ThisIsABigFunctionName x, y, z
```

Then as soon as you press the Enter key, HyperMacro will type

```
end ThisIsABigFunctionName
```

and will then move the insertion point up one line ready for you to begin entering the function definition. HyperMacro responds similarly to repeat, if-then, exit, pass, on, and others (explained later). In addition, there are numerous other shortcuts for entering script items — including ten paste buffers and glossary commands.

Installation
Simply place the HyperMacro INIT into your system folder. Under system 7, the INIT should be placed into the Extensions folder. That is all there is to it! You must now restart your computer before HyperMacro takes effect.

Automatic Macros
It seems strange that when you enter a function header, such as

```
function ThisIsABigFunctionName x,y,z
```

You also have to later type 'end ThisIsABigFunctionName' It seems that HyperCard ought to be smart enough to do that for you. That's where HyperMacro comes in. When you press Enter (note *not* Return!) after typing the line above, HyperMacro will provide the 'end ThisIsABigFunctionName' for you — automatically. It will also position the insertion point just above the end line—ready for you to enter the body of the function.

Similarly, if you type

```
on ThisIsABigHandlerName
```

Then when you press Enter, HyperMacro will insert 'end ThisIsABigHandlerName' and then properly position the insertion point.

Note: All the automatic macros provided by HyperMacro require the Enter key be pressed instead of Return. If you press Return, there is no effect. However, HyperMacro does have a feature that allows you to use the Return key in place of the Enter key. This feature is explained more fully in a later section.

Repeat
Along the same lines, if you type

repeat with i=1 to 100 (or any other repeat statement for that matter)

Then, after you press Enter, HyperMacro types ‘end repeat’ and positions the insertion point in the body of the repeat loop.

Also, if you type just ‘r’ at the beginning of a line and then press the spacebar, then HyperMacro will extend the ‘r’ to ‘repeat.’ See the section on *initial macros* for more details.

If-Then
If you type

if x<y then

after you press Enter, HyperMacro provides the ‘end if,’ and positions the insertion point properly.

If you enter something like

if x<y then put 6 into z

Then in this case an ‘end if’ is not appropriate and HyperMacro will *not* produce the ‘end if’ even if you press the Enter key.

Exit-Pass
If you type ‘exit’ or ‘pass’ and then immediately press Enter, HyperMacro will locate the name of the current handler and insert it immediately after ‘exit’ or ‘pass’. Also, if you type something like ' if x < y then pass ' and press Enter, HyperMacro will supply the appropriate handler or function name at the end of the line.

What if what you wanted was “Exit to HyperCard” ? In this case, you need only type ‘eh’ which when followed by a spacebar will expand to “exit to HyperCard.” This is an example of an *initial macro* explained more fully in the next section.

Note: You can press Enter *at any point* inside a line that produces an automatic action.
For example, suppose that you typed

if x < y then

and were about to press Enter to produce the automatic ‘end if’ response. Then you realize that the ‘y’ should have been a ‘z.’ So you back-arrow over to the ‘y’ and change it to a ‘z.’ You do not have to go to the end of the line before pressing Enter. Press Enter immediately after changing the ‘y’ to a ‘z.’

However, pressing Enter anywhere in a line that does not require an automatic reaction will behave just like pressing Return. So if you press Enter in the middle of the line

put 4 into x
the line will be split just as if you had pressed the Return Key.

Initial Macros

When you type an ‘r’ at the *beginning of a line* it is likely that you are about to enter a repeat statement of some kind (you could be reading from a file—but that is less likely, on average). If you just press the space bar after typing the ‘r,’ then HyperMacro will

finish the repeat for you. After entering the word ‘repeat,’ if you then type a ‘w’ followed by space, then HyperMacro will complete the ‘w’ to ‘with.’ Similarly, if you type a ‘u’ after a ‘repeat’, then HyperMacro will finish the ‘until’ you were about to type. If you type ‘wh’ then HyperMacro completes this to ‘while’.

If you type ‘v’ at the beginning of a line and then press the spacebar, HyperMacro will complete the ‘v’ to ‘visual effect’.

Other initial macros are: (In each case ‘|’ indicates the location of the insertion point.)

a	answer | with -- with the insertion point between the ‘answer and ‘with’
b	set the cursor to busy
c	close file
ch	choose | tool -- with the insertion point between the ‘choose’ and ‘tool’
chb	choose browse tool
cbt	choose browse tool
d	domenu
dr	drag from
e	exit
f	function
g	global
gc	go card
gn	go next card
h	hide
ita	if there is a | then
ire	if the result is empty then
irne	if the result is not empty then
l	lock
m	multiply | by
n	next repeat
o	open file
p	put | into
pb	put | before
pa	put | after
pe	put empty into
r	repeat
rd	read from file
s	set the
sc	set the cursor to
u	unlock
v	visual effect
w	write | to file

All these initial macros are built-in to HyperMacro. Future versions will allow you to modify these macros.

The Menu Initial Macros

Since menus play an important part in so many HyperCard stacks, HyperMacro has built-in capability for easily creating menu scripts.

If you type ‘ondm’ and then press the space bar (or the escape key — see below), then this mnemonic will expand to the following generic menu-handling script:

```
on domenu   theItem
            |
     pass domenu
end domenu
```

The insertion point will be placed ready for you to write the code for various items.
Also, this macro makes it impossible that you will forget to put the 'pass domenu' line at the end of the domenu handler. If you forget to do this, you will not be able to access any menus except those named in your domenu handler.

Also, in the domenu handler, you can type 'iii' at the insertion point and then after pressing the Space Bar or Escape key the iii will expand to:

```
if theItem is | then

    exit domenu
end if
```

After entering the name of the menuItem, just press the down-arrow key to move to the next line to continue entering the rest of the script.

Glossary Macros — via the Escape Key
There are a number of built-in glossary abbreviations that HyperMacro will expand to full text when you press the Escape key. Later versions of HyperMacro will let you define your own glossary (and other) macros. Like the initial macros, these will be user-definable in a future version.

For example if at any time you enter 'ml'(or 'ML'— case doesn't matter), and next press the escape key, then HyperMacro will replace the 'ml' by 'mouseLoc'.

Other Escape Macros

bf	bg field
cf	cd field
cl	clickLoc
cm	cantModify
ctn	charToNum() — and the insertion point is inside the parenteses
ds	dragSpeed
hi	hilite
hs	heapSpace
iw	is within
inw	is not within
lm	lockMessages
lr	lockRecent
ls	lockScreen
lt	lockText
mc	mouseClick
mh	mouseH()
ml	mouseLoc
mv	mouseV()

nf	numberFormat
ntc	numToChar
of	offset() — and the insertion point is inside the parenteses
pc	paramcount
ra	random
um	userModify
shi or sk	shiftKey
ent or ek	enterKey
ret or rk	returnKey
tab	tabKey
ctl or con	controlKey
cmd or com	commandKey
opt	OptionKey
rif	returnInField
eif	enterInField
si	set the icon of
tt	the target
ondm	on domenu theItem

```
          |
     if theitem

  pass domenu
end domenu
```

The Edit Point and Tilde Copy / Paste

HyperMacro provides the mechanism of an *Edit Point*. What this means is that you can quickly and easily move text from one part of a script to another — without losing your place in the script! You are no doubt aware of the phenomenon wherein you decide to copy some portion of your script and then paste it in at the current insertion point. But unfortunately, after you copy the text, you have to scroll around looking for the place you wanted to paste this stuff into. With HyperMacro you avoid this difficulty.

HyperMacro remembers the last location in which you typed a character. This point is referred to as the *Edit Point*. If you now select some text in your script and then press the `/tilde key (on the upper left of most keyboards), then the selected text will automatically be placed at the Edit Point and you will be moved back to that location to continue editing.

Note: Be sure and just press the backquote key — you do not need to type a tilde character and so you do *not* press the Shift Key for this operation. Because the tilde is the most outstanding character on this key, we refer to it as the tilde key. Should you need to type a backquote character, you need only hold down the control key while typing the backquote.

Try it on a large script. Type a space (say) in the middle of the script. The go a few pages up and select some text. Now press the tilde Key, and you will see the text automatically placed after the space you previously typed.

Note: The Edit Point is *not* changed by just clicking at some location in the text. You must type a character there first. You can also force a particular location to be the Edit Point without typing anything by just clicking there while holding down the control key.

Moving Around the Script
HyperMacro will let you move quickly from word to word. If you hold down the option key and type '→', then you will move to the beginning of the next word to the right. If you hold down the option key and type '←' then you will move to the end of the previous word to the left.

Note: You can move to the end of the current line by pressing command and '→' and you can move to the beginning of the current line by pressing command and '←'.

Repeat the Previous Line
If you hold down the Shift Key when pressing the Return key or the Enter key at the end of a line, then HyperMacro will automatically insert a copy of the previous line of the script. This is frequently useful since most scripts contain several lines that are very similar, and it is convenient to be able to simply enter a copy of one line and then edit it.

For instance if the variables box1, box2 and box3 are all to be initialized with the value 5 then the script would contain the lines

```
put 5 into box1
put 5 into box2
put 5 into box3
```

Here, after typing the first line, hold down the Shift Key when pressing the Return key or the Enter key and then change the 1 at the end to a 2. Hold down the Shift Key yet again when pressing Return and change the 2 in the resulting line to a 3.

Repeating Part of the Previous Line
If you hold down the Shift Key and press Return or Enter somewhere in the middle of a line, then only the portion of the line to the left of the insertion point gets repeated.

Double Quotes and Parenthetical Pairs
You seldom type just one quote in a script. Usually, if you type one quote you are intending to type a string and then follow the string by typing another quote. Both quotes require holding down the shift key. With HyperMacro you only need to press the single quote key — no Shift Key required. When you press the single-quote key, you will see two double quotes appear and the insertion point will be between them. Should you need to type a single-quote character (which will occur much less frequently), you need only hold down the control key while typing the single-quote. Also, holding down the Shift Key when pressing the quote key still produces just one double-quote.

Similarly, if you press the '\' key then you will get a pair of parentheses with the insertion point in-between them. Should you want to type a '\' character, just hold down the control key when typing the character.

Clipboards Galore
HyperMacro provides ten additional clipboards other than the standard cut/paste clipboard provided in the menu.

To copy something into a clipboard, just select the text and press Shift+Control + a number key. For example holding down the Shift and Control keys while typing a 5 will assign the selected text to clipboard 5. Clipboards are numbered from 0 to 9.

To paste the contents of one of the clipboards into your script you may either type a number key while holding down the Control key or you can type the number and then

press the Escape key. So typing Control+ the 5 key will insert the contents of clipboard number 5.

Note: The clipboards do not replace the contents of the paste buffer. You can still use cut and paste as always.

Return Key Option for Automatic Macros
If you like, you can have the Return key activate HyperMacro's automatic actions instead of the Enter key. Just press Shift + Control + delete. You will hear a beep. After that, you will discover that pressing Enter will close the script (like it did before HyperMacro was installed), and pressing Return after typing 'on ThisHandlerName' will produce the automatic 'end ThisHandlerName'. To switch back to using the Enter key, just press Shift + Control + delete again.

Caution
The down side to using the Return key instead of the Enter key is that you will occasionally press Return when you don't want an automatic macro to execute. This can be annoying. However, if the cursor is at the end of a line like 'if x<y then' and you don't want the automatic 'end if' to follow, then hold down the Control key when you press Return.

The Future of HyperMacro
Future versions of HyperMacro will allow the user to define glossary macros and modify the interface in other ways — such as defining which keys will trigger glossary items or clipboards. Other editing improvements are planned as well. The Function keys and keypad will be usable, too.

Your feedback as to what features you would like to see in future releases and what you like or don't like about the current product are greatly appreciated. Any suggestions used in future releases will be acknowledged. Of course bug reports (gasp!) are solicited as well (even though they make me sad).

HyperMacro 1.1 is a CommentWare product. If you decide to keep it, you should send your opinion of the software to one of the addresses below to become a registered user. Registered users get more support and information about the product. Future versions of HyperMacro may not be free.

For further information write to:
David P. Sumner
1009 Walters Lane
Columbia, SC 29209

or (better) Use E-Mail
AOL: DAVIDSUM
CIS: 75515,1507
Internet: sumner@math.scarolina.edu

Acknowledgements: A very large Thanks to the following people who provided bug reports and

suggestions that improved this and future versions of HyperMacro.
Brian Blood, Peter Stubbs, Jean Berthomieu, Marcel Blanchaer, Ewa Wojcicka and Malcolm Abel.

Tutorial

This tutorial is not intended as an example of good scripting; only as a simple example of how to use HyperMacro. For example, suppose that you wanted to enter the following script.

```
function isANumber x
global FlagValue
if FlagValue = 0 then exit isANumber
 put 0 into value
 if x="*" then put 10 into value
 if (chartonum(x) >47) and (chartonum(x)<58) then
            put 1 into value
 end if
 if x="+" then put 11 into value
 if x="-" then put 12 into value
 if x="/" then put 13 into value
 if x= "^" then put 14 into value
 if x="(" then put 15 into value
 if x=")" then put 16 into value
  return value
end isANumber
```

You would proceed as follows. First type f and spacebar to produce function. Then type isnum x and press Enter. Now type g followed by a spacebar to produce global. Then type FlagValue. Press Return. (If you expect to use the expression FlagValue a lot, you might choose to assign it to a paste buffer). Type if and then select FlagValue in the previous line by double clicking on it and press the tilde key to transfer a copy of FlagValue to the Edit Point. (If you had assigned FlagValue to (say) paste buffer 3, then you would just press Control + 3 here instead). The type = 0 then exit and press the Enter key. HyperMacro then finishes the line by entering isANumber. Press Return to continue entering the script.

Next type p and spacebar to produce put | into . At the insertion point type 0 and then command + right arrow to go to the end of the line and type value. Now hold down the shift key when you press return. Use command+left arrow to go to the beginning of the line and type if x = "*" then — don't forget that you can type the double quotes by pressing just the single quote key once. Now use option + right arrow to get to the 0 (zero) and type a 1 in front of it. Now use down-arrow to get to the next line.
On the next line type if \ which expands immediately to if (|). Then type ctn and press Escape and the line now reads if (chartonum(|)) So now type x right arrow and > 47 right arrow. Now this line looks like this:

```
if (chartonum(x) >47)
```

Type and and then go back and select the (chartonum(x) >47) and press the tilde key to move a copy to the current editing position. Now change > 47 to < 58. Now use command + right arrow to ensure you're at the end of the line and type then and press Enter. At the resulting insertion point type p followed by a spacebar and type a 1 at the insertion point. Use command + right arrow to go to the end of the line and type value.

Now use the down-arrow and/or return key to get to the line just below the end if. Type a space or click while holding down the Control key so as to set the Edit Point. Now select the third line if x="*" then put 10 into value and press the tilde key to place it at the edit Point. Change the "*" to

"+" and the 10 to 11.

Now, don’t just press return to go to the next line, but instead hold the Shift key down as you press Return. You can now quickly edit this new line and then continue pressing Shift+Return to produce the next several lines.

Finally on the last line type return value and you are finished.

As an option, if you realized that you were going to be using lines like

```
if x="*" then put 10 into value
```

Quite a bit, then you might choose to select that line and assign it to paste buffer number 0 and then whenever you wanted to type that line, you need only type Control + 0 or just type a 0 and then press the Escape key.

Changes in Version 1.1

1. Version 1.1 is not sensitive to the name the user assigns to HyperCard.
2. Version 1.1 still works even when the Watch Variable and Watch Message palettes are open.
3. CBT or CHB will expand to "Choose Browse Tool"
4. HyperMacro 1.1 works if the scripting messages are French or German.
5. A few technical changes and minor bug fixes.
6. Portions of the manual were redone.

Acknowledgements: A very large Thanks to the following people who provided bug reports and suggestions that improved this and future versions of HyperMacro.
Brian Blood, Peter Stubbs, Jean Berthomieu, Marcel Blanchaer, Ewa Wojcicka and Malcolm Abel.