Apple II
File Type Notes

Developer Technical Support

File Type: $BC (188)
Auxiliary Type: $4002

Full Name: Super Info module
Short Name: Super Info module

Written by: Jim Lazar & Matt Deatherage March 1991

Files of this type and auxiliary type contain modules for Super Info, a shareware desk accessory.

Super Info II is a shareware desk accessory distributed through online services and other electronic means. The package includes the Super Info II desk accessory with associated files, programming manuals and sample source code for Super Info modules. The package is also available directly from the author for $13 ($12 shareware fee plus $1 postage and handling).

For more information on Super Info, contact:

Jim Lazar
1109 Niesen Road
Port Washington, WI 53074
AOL: WinkieJim
GEnie: WinkieJim
CompuServe: 70401, 2677

Definitions

The following definition is used in this document in addition to those defined for all Apple II file types:

Rectangle A standard QuickDraw II rectangle—four **Words** indicating, in order: the top, left, bottom and right of the rectangle.

Version A **Word** indicating the version as defined in Appendix A, Volume 2 of the *Apple IIGS Toolbox Reference*. Only bit 15 is reserved for prototype identification, unlike system tool set version numbers defined in Apple IIGS Technical Note #100, VersionVille.

The File Format

All Super Info modules **must** begin with the Super Info module header, followed by the module itself:

offset	(+000)	**Word**	Offset from the start of the file to the page dispatcher, which is the code entry point.
version	(+002)	**Version**	The version number of this module.
SIVer	(+004)	**Version**	The minimum version of Super Info necessary for this module to execute. (For differences in versions, see "Super Info Versions" later in this Note.) The prototype bit (bit 15) in this **Version** must be clear. This note describes version 2.1 ($0201) of Super Info.
reserved	(+006)	**Long**	Set to zero.
menuKey	(+010)	**2 Bytes**	Two ASCII characters for the module's menu item key equivalents. Menu key equivalents will be assigned by the author; contact him at the address provided in this Note for assignments of menu keys. Available keys are limited; do not use key equivalents unless absolutely necessary.
string	(+012)	**String**	Pascal string with the module's menu entry. This string must not change while the Super Info window is open.
module	(+xxx)	**Bytes**	The code for the module.

How you write the module after the header is completely up to you.

The Super Info Page Dispatcher

The page dispatcher is the only entry point into your module. Super Info calls this entry point in full native mode. On entry, `A` contains the operation code, while `X` and `Y` contain a pointer to Super Info's global data (the low word is in `X`).

The action your module takes is determined by the operation code passed in `A`. These codes are discussed in detail later in this Note in the section "The Operation Codes."

Your module may use the data bank register freely, but it must preserve the direct page register. Super Info will provide up to a page of direct page space for your module's use, but you must explicitly set the direct page to it. The address is in Super Info's global data area.

The Global Data Area

The global data area contains all the information Super Info has to communicate with your module. You should copy the information to your own buffer, use it as you please and update Super Info's copy when you are finished. For version 2.1, the global data area is 28 bytes long:

NDAWIndow	(+000)	**Long**	Handle of Super Info's desk accessory window. Use this to make any changes to Super Info's window, such as forcing an update, clearing the window or

			creating a control. However, you must not close the window.
BackRect	(+004)	**Rectangle**	Rectangle enclosing the size of your module's visible page. The top left of this rectangle should always be zero. Your module will initialize this rectangle during module initialization, discussed later. If you change this rectangle later, be sure to call the Window Manager routine `SetDataSize` with the new values to update the scroll bars properly.
HexDec	(+012)	**Word**	0 = The user has chosen decimal in the Super Info menu 1 = The user has chosen hexadecimal in the Super Info menu. Always attempt to respect the user's wishes for number display formatting.
NDAPage	(+014)	**Word**	The Super Info page number of your module. Page numbers are assigned by Super Info starting with 6. You can add $4FFA to this value to get the menu item ID number of your module's menu item.
NDAMenu	(+016)	**Long**	The Handle to Super Info's menu bar. Be careful about changing anything in the menu bar.
NDAControl	(+020)	**Long**	The handle to the control that was selected if your module is called with the control operation. For other operations, this field is invalid.
OurDP	(+024)	**Word**	The address of a 256-byte ($100) memory buffer in bank zero that your module may use as direct page space. It is not defined for you by Super Info and may be used by you as you please. You **must** explicitly set the direct page register to this value to use this memory, and you must restore the previous value of the direct page register before returning to Super Info. **Never** use any of the direct page that is active when your module is passed control. You could also use this memory for direct page space for extra toolsets. The author recommends not having this direct page active while making tool calls.
WordWrap	(+026)	**Word**	0 = The user does not want text to wrap to the width of the Super Info window 1 = The user wants text to wrap to the width of the window

The Operation Codes

Version 2.1 of Super Info defines five operation codes; your module should ignore any codes other than these five.

Code 0: Page Initialization (`InitOp`)

The initialization operation performs any setup operations the module needs before the page is drawn. It is called every time your page is selected from the Info menu. Creating controls, reading information from disk, setting the size of the page, and examining system parameters are a few examples of tasks performed during the initialization operation.

The initialization operation requires that you set the size of your page's data area (with the Window Manager routine `SetDataSize)`, set the `backRect` rectangle in Super Info's global data area to the size of your page's data area, and invalidate the rectangle (with the Window Manager routine `InvalRect)`.

Note: The Super Info window is **only** set up for drawing when the draw operation is called. Before doing anything visible (such as invalidating rectangles), be sure to set up the drawing environment with `StartDrawing` and restore it on exit with `SetOrigin`(0,0).

Your module must keep track of all memory allocated (`DisposeAll` is not allowed on your memory ID) and all system parameters changed. When in doubt, remember that Super Info is a new desk accessory (NDA)—don't do anything a stand-alone NDA couldn't do.

Code 1: Page Closing (`CloseOp`)

The close operation allows you to dispose of memory, controls, or other things allocated in your module's other routines. It is called when switching to another module or when Super Info is closed with your module open.

In addition to cleaning up after yourself, your module's close operation **must** erase anything your module drew in the Super Info window. Use the QuickDraw II call `EraseRect` with the `backRect` to do this, after setting up the drawing environment (see the Note under Page Initialization).

Code 2: Page Control (`ControlOp`)

The control operation allows you to interact with any control you've created in your page. The `NDAControl` handle in the global data area contains the handle to the control the user selected (the value is not valid for any other operation).

Be sure to initialize the drawing environment properly if you draw in the window. (See the Note under Page Initialization.) If the user selects check boxes or radio buttons in your page, you must call `SetCtlValue` to change the state of the control. Pop-up menu controls are properly tracked; the currently selected item will be in the pop-up control's `ctlValue` field when your control operation is called.

This operation is not required by Super Info; your code may simply return without acting on it.

Code 3: Page Run (`RunOp`)

The run operation is called every two seconds (unless the system is busy, in which case it may take longer). Your module can use this operation to update information that may have changed since the page was last drawn. You can call your draw routine directly (after initializing the drawing environment; see the Note under Page Initialization) or by drawing in the window within the run operation.

Do not perform time-consuming tasks in this operation or the system will react sluggishly to the user's actions.

Code 4: Page draw (`DrawOp`)

This operation is the workhorse of your module—it handles (or can handle) all drawing in the window's content area. Super Info will erase the `backRect` and call the Control Manager routine `DrawControls` before passing execution to your module's draw operation.

Unlike the other operations, the drawing environment **is** set up for your draw operation. You must not change the current port. The correct port is already set on entry to your draw operation. Anything that is legal for an NDA update routine is legal in this operation.

A Sample Page Dispatcher

The following code is a generic page dispatcher and may be used in your code. Note that this is not the only way to accomplish these tasks, but is one way that's known to work.

```
Page       PHK               ;Set data bank register to
           PLB               ;current bank
           PHA               ;Save operation code
           STX   GlobeAdr    ;Save global data address
           STY   GlobeAdr+2  ;for later use
           PHY               ;Save global data address
           PHX               ;on stack
           LDA   #^GlobalData     ;Save address of our data buffer
           PHA               ;on stack
           LDA   #GlobalData
           PHA
           PEA   #0          ;Copy 28 bytes
           PEA   #28
           _BlockMove        ;(copies global data to our buffer)
           PLA               ;Get operation and multiply by
           CMP   #5          ;two to jump to routine
           BCS   Exit        ;unless the operation is more than
           ASL               ;4 in which case ignore it
           TAX
           JSR   (JmpTable,X)     ;Jump to operation routine
Exit       LDA   #^GlobalData     ;Save address of our data buffer
           PHA               ;on stack
           LDA   #GlobalData
           PHA
           LDA   GlobalAdr+2 ;Save address of original global data
           PHA               ;on stack
           LDA   GlobalAdr
           PHA
           PEA   #0          ;Copy 28 bytes
           PEA   #28
           _BlockMove        ;(copies global data
           RTL               ;back to original location)

JmpTable   dc    i4'Init,Close,Control,Run,Draw'

GlobalAdr  ds    4           ;Storage space for global data address

GlobalData EQU   *           ;Start of your global data buffer
```

```
NDAWindow  ds    4
BackRect   ds    4
VertSize   ds    2
HorzSize   ds    2
HexDec     ds    2
NDAPage    ds    2
NDAMenu    ds    4
NDAControl ds    4
OurDP      ds    2
WordWrap   ds    2
```

Super Info Versions

You can use the list of features in various versions of Super Info listed below to determine the minimum Super Info version your module needs to operate. The lower a minimum version your module can accept, the greater the potential audience for it is (as new versions of shareware products are often slow to distribute).

Super Info Version 1.x

- Did not support modules.

Super Info Version 2.0

- Direct page supplied by Super Info in `OurDP` was only sixteen ($10) bytes. Do not use more than sixteen bytes on the direct page.
- Word Wrap flag not available. Do not attempt to use it. Consequently, the global data area was only 26 bytes long.

Super Info Version 2.1

- Described in this Note.

Further Reference

- *Apple IIGSToolbox Reference*, Volumes 1–3