Developer Technical Support

Apple IIGS
#53: Desk Accessories and Tools

Revised by: Dave "Out of Phase" Lyons May 1992
Written by: Matt Deatherage, Jim Mensch, & Dave Lyons March 1989

This Technical Note discusses compatibility issues that can arise between desk accessories and applications. Where possible, it presents solutions.
Changes since March 1991: Updated information about QuickDraw Auxiliary and `StartUpTools` for System 6.0.

This Note presents guidelines to help applications and desk accessories work together smoothly.

Tool Sets

The greatest conflict between applications and desk accessories, especially NDAs, is the use of system tool sets. The *Apple IIGS Toolbox Reference*, Volume 1, defines the minimum collection of tools sets available to an NDA. The Desk Manager requires that an application start the following tool sets before calling `DeskStartUp`:

Tool Locator (#1)
Memory Manager (#2)
Miscellaneous Tools (#3)
QuickDraw II (#4)
Event Manager (#6)
Window Manager (#14)
Menu Manager (#15)
Control Manager (#16)
LineEdit (#20)
Dialog Manager (#21)
Scrap Manager (#22)

NDAs may assume that these tools are all present and running, so they do not need to check for their presence. NDAs can also use the following tool sets without special consideration for starting them up: Desk Manager, Scheduler, Apple Desktop Bus, and Integer Math.

In addition to the tool sets applications must start to support NDAs, Apple recommends that applications start the following tools:

QuickDraw Auxiliary (#18) (see discussion under QuickDraw Auxiliary)
Font Manager (#27)

These two additional tools are so widely used by desk accessories that they should be present. NDAs may not **assume** their presence, but it is reasonable to write an NDA that **checks** for them, with the assumption that they usually turn out to be available.

NDA Guidelines

Which Tool Sets Can An NDA Use?

- In general, NDAs can use the tool sets which have already been started up by the host application, even tools that are not guaranteed to be started up. Using other tool sets is trickier (see below).

- In general, NDAs should not start up tools that are already started up. (The Resource Manager is an exception.)

- The Resource Manager must be started separately by each client. See Apple IIGS Technical Note #71 for detailed information on using the Resource Manager from an NDA.

- Sound tools are an exception to the rule of freely using a tool which is already started. See the section "Sound Tools" sections later in this note.

- Some tool sets are easily started up each time they are needed, if they are not already present.

  Standard File is an excellent example. If an NDA needs to use Standard File, it should check to see if the tool is already running. If it is not running, the NDA must use `LoadOneTool` to load it, then it must allocate a page of direct-page space and start the tool before using it. When finished with the tool, the NDA must shut it down, deallocate the direct-page space, and unload it with `UnloadOneTool`. (A tool is already running if its `xxxStatus` function returns `TRUE` and does not return an error.)

  The important thing here is that the NDA shuts down Standard File immediately after using it, if it was not already started. This does not cause conflicts with the host application or with other NDAs.

  Note that by pre-initializing the result space of an `xxxStatus` call to zero, you can avoid caring whether the tool is present but not started or simply not present.

  ```
  pea $0000
  _SFStatus
  pla                ;A is nonzero if Standard File is started
  ```

  From a high-level language, you may not be able to pre-initialize the result space. Instead, you need something like the C statement:

  ```
  StdFileActive = ( SFStatus() && !_toolErr);
  ```

  or the Pascal statement:

  ```
  StdFileActive := (SFStatus<>0) AND (ToolErrorNum=0);
  ```

- It is impractical or impossible to start up certain tool sets each time they are needed. These include the Font Manager, Scrap Manager, and Text Edit.

  If an NDA needs to start up a tool and **keep** it started while letting the application continue to run, things get interesting. (There is a risk that the host application will later try to start up the tool set itself and not be able to deal with the tool already being started.)

  In practice, the safest thing you can do for a tool you need to leave running is:

—When your NDA is opened, check the tool set's status. If it is not available, use `LoadOneTool`, allocate any needed direct-page space, start up the tool set, and set a flag indicating that your NDA started the tool set.

—When your NDA's Init routine is called at `DeskShutDown` time (Accumulator equal to zero), check the flag set above. If your NDA started a tool set, shut it down, dispose of any direct-page space you allocated for it, and call `UnloadOneTool`.

(Keep in mind that your NDA can be opened and closed many times before `DeskShutDown` is called when the application finally quits. If you have started a tool and set a flag on an open, be sure not to disturb the flag on a future open, when the tool is already available because you started it! You still need to shut it down at `DeskShutDown` time.)

—Do **not** shut down tool sets when your NDA is closed. To see why, consider what would happen if two NDAs just like yours were used at the same time. If the NDAs were closed in any other than the exact opposite order they were opened, some NDAs would have tool sets shut down from underneath them.

StartUpTools

- `StartUpTools` in System Software 5.0.4 and earlier is designed to be called only by an **application**, not a desk accessory. Unexpected things happen if your NDA calls `StartUpTools` (for example, you may get a second copy of the application's resource fork open in your NDA's private resource search path; this wastes RAM and can interfere with an application's attempt to write to its own resource fork).

- See the System 6.0 Toolbox documentation for information on using `StartUpTools` from an NDA. There are new flag bits you need to know about.

TLStartUp and TLShutDown

- Do not call `TLStartUp` or `TLShutDown` from a desk accessory.

- You may call `MMStartUp` at any time to get your desk accessory's own memory ID. This does not allocate a new ID; it just tells you what ID you already have (it returns the memory ID of the block the `MMStartUp` call is made from).

User Tool Sets Belong to the Application

- A desk accessory must not install user tool sets, because there is no arbitration of user tool set numbers. User tool sets are the sole property of the current application.

  A desk accessory should not call user tool sets even if it determines that the host application has installed a certain tool set, because that limits future system software options. For example, consider a hypothetical multiple-application environment. If DAs call user tool sets and the system automatically switches between separate collections of user tool sets, there would be no way for the system to know which set to switch in before giving control to a desk accessory.

Bank Zero Memory and Error $0201

- If you get error $0201 (unable to allocate memory block) while trying to launch a ProDOS 8 application, it is probably because your NDA allocated some memory in bank 0 or bank 1 and failed to dispose of it at DeskShutDown time (when the NDA's Init routine is called with the accumulator equal to zero). GS/OS needs to allocate all of this memory for ProDOS 8 to use.

QuickDraw Auxiliary

- In System 6.0 and later, QuickDraw Auxiliary is always available to an NDA, because the Window Manager automatically loads and starts QuickDraw Auxiliary (because it's needed for `AlertWindow`, for example). To prevent problems, duplicate `QDAuxStartUp` and `QDAuxShutDown` calls are tolerated, and `QDShutDown` automatically calls `QDAuxShutDown`.

- Before System 6.0, starting QuickDraw Auxiliary when the application has not started it can be a problem. An application that correctly implements switching between 320 and 640 mode calls `QDShutDown` and `QDStartUp`. QuickDraw Auxiliary depends heavily on QuickDraw, and restarting QuickDraw while QuickDraw Auxiliary is active will fry big-time.

Sound Tools

- A desk accessory **cannot** use any of the sound tools if they are already started. This is contrary to the rule for other tool sets, but it is required because there is no memory management of the sound RAM (or "DOC RAM"). If the Sound Tools (#8) are started, the application has exclusive control of the 64K DOC RAM used to play sounds. Anything your desk accessory might put there could overwrite information the application needs.

  Saving and restoring DOC RAM around desk accessory usage is **not** sufficient. Many of the sound functions are interrupt driven, altering the contents of DOC RAM only during sound interrupts, so your desk accessory might attempt to replace parts of DOC RAM which are being played. Since there is no memory management of DOC RAM, desk accessories must avoid the sound functions of the IIGS if the application is already using them.

Application Guidelines

For best compatibility with NDAs, applications should follow the following guidelines.

- Be careful about when your application starts and shuts down tools. A highly compatible approach is to start tools at the beginning of your application and leave them started. For certain tools, like Standard File, it is reasonable to load and start the tool set each time it's needed (you may want to check whether it's already started, in case some impolite NDA started Standard File and left it started).

  Note that `UnloadOneTool` followed later by `LoadOneTool` does not necessarily cause disk access or ask the user to insert the boot disk. `UnloadOneTool` calls `UserShutDown` to put the tool set into "zombie" state, where it can be restarted from memory if none of its segments have been purged. Unloading tools while they aren't in use is a Good Thing—if the user has plenty of RAM, there's no noticeable performance hit, but if RAM space is tight then doing extra disk access still is preferable to actually running out of memory.

For maximum compatibility, an application should not shut down any tools which were ever active when it called `SystemTask` or `TaskMaster` (until quitting time, of course, when it shuts down everything, starting with the Desk Manager). The application can start more tools, but it should not shut down those which are already active.

If your application is going to start a tool and not keep it started, use it and then shut it down immediately, without allowing desk accessories to be opened during that time.

- Don't just start the Scrap Manager—use it! Many desk accessories support cutting and pasting to exchange text and pictures with your application, but they can do it only if you use the Scrap Manager. If you have a need for your own private scrap internally, you should still **also** use the Scrap Manager to exchange text and pictures with other applications and DAs.

- Provide an Edit menu, and when an NDA window comes to the front enable the menu and the Undo, Cut, Copy, Paste, and Clear items.

- Applications should never make a `Close` call with reference number zero at file level zero. (If you need to use `Close` with reference number zero, use `GetLevel` and `SetLevel` to avoid closing files you did not open.)

  DAs written recently can open their files at an internal file level, as documented in GS/OS Technical Note #13, but applications still need to avoid closing all files at level zero for compatibility with older desk accessories.

- An application with some memory to spare can save NDAs time by providing them the additional tools which they are most likely to use.

  The most common tools which desk accessories require besides those available in the standard Desk Manager set are QuickDraw Auxiliary (#18), the Print Manager (#19), Standard File (#23), the Font Manager (#27), and the List Manager (#28).

- When you call `TaskMaster` or `GetNextEvent`, or `EventAvail`, be sure bit 10 is turned on in the event mask, to enable "desk accessory" events. If you turn this bit off, users will not be able to get to the Classic Desk Accessory menu by pressing Apple-Ctrl-ESC.

CDA Guidelines

- CDAs are nearly always modal, but by using the HeartBeat interrupt queue or other mechanisms, they can get control when the user is no longer "in" the CDA. The list of guaranteed tools for NDAs does not apply to CDAs, and CDAs must be prepared to deal with the ProDOS 8 environment as well as GS/OS.

- Under ProDOS 8, a CDA will not be able to allocate any bank 0 space through the Memory Manager; it can only use page 0 and page 1 safely (the stack is in page 1).

- Do not call `TLStartUp` or `TLShutDown` from a desk accessory.

- You may call `MMStartUp` at any time to get your DA's own memory ID. This does not allocate a new ID; it just tells you what ID you already have (it returns the memory ID of the block the `MMStartUp` call is made from).

Further Reference:

- *Apple IIGS Toolbox Reference*
- *Programmer's Introduction to the Apple IIGS*
- Apple IIGS Technical Note #71, Desk Accessory Tips and Techniques
- Apple IIGS Technical Note #83, Resource Manager Stuff