

Date: January 9, 1986

Author: John Worthington

Subject: Classic Desk Accessories

Document Version Number: 00.10

---

Revision History:

00.00	(11/26/85)	Initial Release.
00.10	(01/09/86)	Major Changes to the Data Structures

## 1. About the Classic Desk Manager

The Desk Manager provides the user access to desk accessories. A desk accessory is a "mini-application" that can be run at the same time as a Cortland application.

The Desk Manager provides support for two types of desk accessory's: Classic Desk Accessories (CDA) and New Desk Accessories (NDA). This document will concern itself with CDAs and the parts of the desk manager that support these accessories. This information will eventually be included in a comprehensive document describing both type of desk accessories.

### 1.1 What is a Classic Desk Accessory?

Classic Desk Accessories are desk accessories that are designed to execute in a non-desktop, non-event based environment. Unlike NDAs, a classic accessory gets full control of the machine during what is basically an interrupt state (generated by a keypress). The desk accessory is responsible for saving any of the applications memory that it uses as well as handling all I/O.

### 1.2 Using Classic Desk Accessories

The user opens a CDA only through a menu activated by key-press (OPEN APPLE-CONTROL-ESCAPE). When a desk accessory is activated, the 80-column screen is saved as well as the system direct page. MouseText characters are enabled and the desk accessory is activated via its open routine. When the desk accessory is closed it should RTL to the Desk Manager.

## 2.0 Desk Manager Routines

These are the desk manager routines that are applicable to classic desk accessories. The complete set of desk manager routines will be documented in (\*\*\*) to be completed (\*\*\*) Desk Manager, the ERS.

FUNCTION ChooseCDA : INTEGER;

ChooseCDA will save the 80-column text screen and system direct page, switch in main memory, enable mousetext characters and

display a menu of the available classic accessories. Control is returned to the application when the desk accessory has completed execution.

The value returned by ChooseCDA will be zero and the carry flag will be cleared if the accessory successfully completed execution. If there was an error, carry will be set and the error returned.

FUNCTION CloseCDA : INTEGER;

Normally CDAs will close and exit through the ChooseCDA call. CloseCDA is a "emergency" call that will force desk accessories to close. A CDAs close routine should perform whatever shutdown in necessary and RTL to the Desk Manager. CloseCDA returns values in a manner similar to that of ChooseCDA.

FUNCTION ActiveCDA (daHandle: HANDLE) : INTEGER;

ActiveCDA tells the desk manager that a particular CDA should be considered active even after it is closed. This is useful for a desk accessory such as a clock that might require periodic service. If the desk accessory is not "active" then its control routine will not be called.

Note that there can only be one active desk accessory at a time. To de-activate the accessory, call ActiveCDA with NIL. An error code is returned if appropriate.

FUNCTION InstallDA (daType : INTEGER, daHandle : HANDLE) : INTEGER;

InstallDA is called to place the desk accessory in the desk manager's chain. This must be done in order to use the desk accessory. daType is an integer specifying the type of the desk accessory. The types are:

<u>Bit</u>	<u>Type</u>
0	set if Classic Desk Accessory
1	set if New Desk Accessory
2-15	reserved

FUNCTION RemoveDA (daHandle : HANDLE) : INTEGER;

RemoveDA removes the DA that is pointed to by daHandle from the desk managers chain. The removed DA is marked as purgable.

FUNCTION RestScreen : INTEGER;

RestoreScreen will restore the screen area saved by the desk manager. This should be called prior to closing the CDA and returning control to the application. Note that the screen is automatically restored for most CDAs.

FUNCTION SaveScreen : INTEGER;

SaveScreen will save the 80-column text screens in bank 00, 01, E0 and E1. This new image of the screen will be used for subsequent calls to RestScreen.

## 2.5 Performing Periodic Actions

PROCEDURE SystemTask;

For each open desk accessory, SystemTask causes the accessory to perform the periodic action defined for it, if any such action was defined and if the proper time period has passed since the action was last performed. For example, a clock accessory can be defined such that the second hand is to move once every second; the periodic action for the accessory will be to move the second hand to the next position, and SystemTask will alert the accessory every second to perform that action.

For classic desk accessories SystemTask should be called only when the accessory should perform its periodic action. Calling System Task will cause a CDA to perform the periodic action defined for it each time that SystemTask is called. Note that if the event manager is not being used then the CDA is responsible for setting up a periodic interrupt to call SystemTask. See the Misc. Tool Manager for details.

## 3. Writing Desk Accessories

This section assumes that the desk accessories have somehow been loaded into memory. The details of that mechanism will be described in the (\*\*\*)to be completed(\*\*\*) Desk Manager ERS.

A desk accessory begins with a link to the next CDA in the chain, followed by few words of flags and other data, offsets to the routines that do the work, an optional specification for a save area, followed by an optional title and finally the routines themselves.

byte	0	daLINK (long)	link to next DA
	4	daFLAGS (word)	flags
	6	daOPEN (long)	offset to open routine
	10	daCLOSE (long)	offset to close routine
	14	daCONTROL (long)	offset to control routine
	18	daNAME (byte)	length of DA name
	19	daNAME+1 (bytes)	characters of DA name

The first item in the definition for a desk accessory is a handle to the next CDA in the desk managers list. The desk manager maintains a linked list for each type of desk accessory. Desk accessories are added and removed from the list by tool calls. The list itself should not be manipulated.

The daLink field is initially empty. The desk manager will install the link to the next accessory. The field should be NIL for the last accessory.

The daFLAGS are defined as follows:

<u>NAME</u>	<u>BIT</u>	<u>FUNCTION</u>
dNeedPro16	0	set if CDA runs under ProDOS/16
dNeedPro	1	set if CDA runs under ProDOS/8
dNeedDOS	2	set if CDA runs under DOS 3.3
dNeedPascal	3	set if CDA runs under Pascal
dSaveArea	4	set if CDA doesn't want entire screen saved
dRestoreScr	5	set if Desk Manager shouldn't restore screen
dPascalEntr	6	set if CDA uses Pascal I/O hooks
(reserved)	7	
(reserved)	8	
(reserved)	9	
(reserved)	10	
(reserved)	11	
(reserved)	12	

dNeedTime	13	set if CDA needs time for periodic actions
dNeedLock	14	set if CDA will be locked in memory as soon as its opened
dStayLock	15	set if the CDA is non-relocatable

A CDA may run only under a particular operating system. The first 4 flags specify the operating system(s) that the CDA operates under. If the CDA operates under a particular operating system then the flag for that system should be set. A CDA that doesn't operate under any of the listed operating systems may not be opened.

Flag bits 4 and 5 deal with the state of the screen. If bit 4 is clear then the desk manager will automatically save the 80-col text screens in bank 0 and E0. If bit 5 is clear the desk manager will restore the text screen before returning control to the application. If bit 5 is set then the desk manager will not restore the screen area prior to returning control to the application. This might be useful for accessories such as a clock that wish to remain on the screen.

If bit 6 is set then the desk manager will initialize the Pascal hooks to the 80-column firmware. The desk manager itself uses the Basic entry points.

Desk accessories may need to perform predefined actions periodically. For example, a clock desk accessory may want to change the time it displays every second. If SystemTask is called, the dNeedTime flag is set, and the desk accessory is active then the CDA's control routine will be called regardless of the tick count.

Whenever possible the CDAs should be relocatable. This will allow for more efficient memory management. If dStayLock is clear then the desk manager will handle locking and unlocking the desk accessory as necessary. If it is set, then the desk accessory will remain locked all the time, possibly fragmenting memory.

Following these four words are pointers to the desk accessory's routines and a title for the desk accessory (preceded by its length in bytes). A desk accessory's title should be no longer than 20 bytes.

(note)

A practical size limit for desk accessories is about 8K bytes.

### 3.1 The Desk Accessory Routines

Only three routines exist for classic desk accessories: the open, close, and control routines.

The open routine opens the desk accessory:

When the open routine is called, the screen area has been saved, the system direct page is saved, the main/aux mem soft switches are set correctly, and the mousetext characters are enabled.

The CDAs open routine should display the accessory, and is at this point has control of the entire machine. Of course, one should not take undue advantage of this. The desk accessory should save anything it is going to use (other than areas saved by the desk manager). Remember, you are a guest in some one else's house. Be polite!

When the desk accessory is closed, you should RTL back to the desk manager. The desk manager will then restore the the machine state, along with any areas it saved prior to returning to the application.

The close routine is an emergency shutdown routine for desk accessories. When this routine is called, the CDA should restore the machine state to what it was when it was opened. The CDA should then RTL back to the desk manager.

Currently, a CDA is executed only with interrupts off. This means that practically speaking the close routine will not be called. We will not promise that this will always be the case. For compatibility with future systems, the close routine should be supported by all desk accessories.

The operation of the control routine is application specific. It is called if the desk accessory is active and the SystemTask is called. Note that "active" implies that the desk accessory has called the desk manager ActiveCDA to tell the desk manager to call you.

The control routine will be called each time the SystemTask is called.