SetEntries Set a group of color table entries for the current gDevice

#include <Quickdraw.h> **Color Manager**

```
void    SetEntries(start, count, aTable );
short   start ;     beginning position for specified number of entries.
short   count ;     total number of entries in group
Ptr     aTable ;    cSpecArray
```

SetEntries sets the enumerated color table entries for the current gDevice.

start is the given position where the specified number of entries begin.

count yields the total number of color table entries in the group being set.

aTable is a cSpecArray, not a color table

Returns: none

Notes: **SetEntries** sets many color table entries to the current gDevice. Its *aTable* parameter points to a cSpecArray (rather than a color table) in which the colorSpec.value field of the entries has to fall within the logical range for the gDevice's assigned pixel depth. For example, if the gDevice lists a 4-bit pixel size, the colorSpec.value fields must be in the 1 to 15 range and, if the device has 8-bit pixels, the colorSpec.value fields range from 0 to 255. All values are zero-based so setting three entries means that you pass a 2 in the count parameter.

Since **SetEntries** lists its positional information in terms of logical space instead of memory locations as used by the gDevice, specifying a color table change won't necessarily get you a change in the hardware's color table--but it will be correctly reflected in the colors on the screen.

There are actually two **SetEntries** modes--**index mode** and **sequence mode**.
For the latter, the distinguishing characteristic is a specified start position and a length. Index mode **SetEntries**, on the other hand, leave the specifying of where the data will be installed up to the cSpecArray (which handles such chores on an individual basis for each entry).

While sequence mode loads new colors in the same order they appear in the aTable, with clientID fields for changed entries copied from the gDevice's gdID field, index mode installs each entry based on a position specified by its colorSpec.value field in the cSpecArray. Then, in the gDevice's color table, all changed entries' colorSpec.value fields get the gdID value. You initiate the index mode by passing -1 for the start position, with a valid count and pointer to the cSpecArray.

Changing a color table entry invalidates all cached fonts as well as the seed number, meaning that the inverse table will be rebuilt by the next drawing operation. If any requested entry is protected or out of range, you get a protection error and nothing else. If a requested entry is reserved, the gDevice's gdID has to match the low byte of the intended colorSpec.value

field before the entry can be changed.