WinCTab structure

```
#include <Windows.h>

typedef struct WinCTab {
  long        wCSeed;          4    0    Reserved field
  short       wCReserved;      2    4    Reserved field
  short       ctSize;          2    6    Total table entries minus 1
  ColorSpec   ctTable[5];      4    8    Address of a five-element color table
} WinCTab;                     12

typedef WinCTab *WCTabPtr;
typedef WinCTab **WCTabHandle;
```

Notes: Both the wCSeed and wCReserved field have been set aside by Apple and their values are currently zero.

ctSize describes the number of elements in the color table and the value is always 4 for a standard definition procedure since the standard color table always contains five elements, ie, definitions for the colors of: content area, frame, text, highlight and title bar. If you decide to build a custom definition procedure, you can make color tables as large or as small as you want and define their parts any way you want. The only reservation you have to bear in mind is that part identifier numbers 1 through 127 belong to the system.

The color table field (ctTable) whose address is given by cSpecArray contains 5 integers--each identifying a part of the color window. Associated with each of these PartIdentifier fields are value fields for the red/green/blue components that comprise that part's color. The whole **WinCTab** for a standard window can be described as a structure that looks like:

wCSeed= 0	
wCReserved= 0	
ctSize= 4	
Color Table = Window Part & values	Component Colors for Each Part (values supplied by application)
content = 0	red green blue
frame = 1	red green blue
text = 2	red green blue
highlight = 3	red green blue
title bar = 4	red green blue

Except for the 32-bit unsigned long in the wCSeed field, all the other field's values are described by 16-bit integers. The default values for the RGB fields (known individually as, partRGB) produce a black and white window. Individual applications supply their own color component values from the awOwner and awCTable fields of the auxiliary window record, AuxWinRec.

The default color table is defined as the one pointed to by the awCTable field of the last AuxWinRec on the auxiliary window list. It is recognized as the last (and therefore the default) by NILs in its awNext and awOwner fields.

If you're setting up custom color tables, you should still use indices 0 to 4 to provide the values for the wContentColor, wFrameColor, wTextColor, wHiliteColor and wTitleBarColor, partIdentifier fields, respectively. By sticking with the standard index values you will maintain compatibility with the default mechanism.

Alternatively, you can bypass the default if you allocate an explicit AuxWinRec for every window your custom definition function creates. Either way, you have to coerce the cSpecArray to a WCTabHandle to access the nonstandard window color table.

The 'wctb' resource is a duplicate of the color table, with partIdentifier and partRGB fields taking on the names, colorSpec.value and colorSpec.RGBColor.