Custom Search and Complement Procedures

Custom search and complement procedures allow an application to override the inverse table matching code. The desired color is specified in the RGBColor field of a ColorSpec record and passed via a pointer on the stack; the procedure returns the corresponding pixel value in the value field of the ColorSpec.

A custom search procedure can provide its own matching rules. For instance, you might want to map all levels of green to a single green on a monitor. To do this, you could write and install a custom search procedure that is passed the RGB under question by the **Color Manager**. It can then analyze the color, and if it decides to act on this color, it can return the index of the desired shade of green. Otherwise, it can pass the color back to the **Color Manager** for matching, using the normal inverse table routine.

Many applications can share the same graphics device, each with its own custom procedure. The prodedures are chain elements in a linked list made up of elements of type SProcHndl. The list begins in the gdSearchProc field of the gDevice port.

The interface is as follows:

```
pascal Boolean SearchProc (RGBColor *rgb, long *position);
```

Note: There is an error in *Inside Macintosh* in the Pascal definition of this function. The *rgb* parameter should be a VAR parameter. So if you are writing a searchProc in Pascal, the interface should be:

```
FUNCTION SearchProc (VAR rgb: RGBColor; VAR position: LONGINT):
    BOOLEAN;
```

When attempting to approximate a color, the **Color Manager** calls each search procedure in the list until the Boolean value returns TRUE. The index value of the closest match is returned by the position parameter. If no search procedure installed in the linked list returns TRUE, the **Color Manager** calls the default search procedure.

The application can also supply a custom *complement procedure* to find the complement of a specified color. Complement procedures work the same as search procedures, and are kept in a list made up of elements of type CProcHndl beginning in the gdCompProc field of the gDevice.

The default complement procedure simply uses the 1's complement of the RGB color components before looking them up in the inverse table. The interface is as follows:

```
pascal void CompProc (RGBColor *rgb);
```