

SetStdProcs

Set graphProcs field to point to custom routines

#include <Quickdraw.h>

Quickdraw

```
void      SetStdProcs(ll_procs );
QDProcsPtr ll_procs ;      address of an empty QDProcs structure
```

SetStdProcs stores the addresses of the standard Quickdraw procedures into a structure intended to be used in a GrafPort.

ll_procs is the address of a 52-byte QDProcs structure. Upon return, all fields of the structure have been set to contain the addresses of the standard low-level routines used by Quickdraw.

Returns: none

Notes: **SetStdProcs** is used by applications that want to intercept selected low-level routines (e.g., the picture-comment handler) while continuing to use the other standard routines.

Since Quickdraw lives up to its name, most applications won't need to replace its code. If you do wish to install a custom Quickdraw function handler (sometimes called a "bottleneck" routine), follow these steps:

- Create a function that accepts the same parameters in the same order as one of Quickdraw's **StdXxx** functions.
- Open a GrafPort (**OpenPort** or **NewWindow**)
- Create a standard QDProcs structure by allocating it and then using **SetStdProcs** to initialize it.
- Store the address of your custom procedure into the appropriate field of the QDProcs structure.
- Store the address of your QDProcs structure into the grafProcs field of the desired GrafPort.

Now, when your application invokes a Quickdraw function that passes through the intercepted bottleneck, your custom handler will get control.

You need not replace all the functionality of a bottleneck-you may choose to simply pre-process the parameters passed to you and then invoke the original handler, as illustrated in the following example:

Example

```
#include <Quickdraw.h>
#define MY_COMMENT 1234

QDProcs      myQDProcs;          /* uninitialized data structure */
pascal short myCommentProc(void); /* declare the function */

SetStdProcs( &myQDProcs );      /* initialize with defaults */
myQDProcs.commentProc = (QDPtr)myCommentProc; /* change one proc */
thePort->grafProcs = &myQDProcs; /* install the change */
```

```
.  
:  
/* Now each time DrawPicture is called, and a PicComment  
   is encountered, the comment will be sent to the custom handler. */
```

```
pascal short myCommentProc(short dataKind,short dataSize, HandledataHandle )  
{  
    if ( dataKind = MY_COMMENT ) {  
        /*... do something special with this comment ... */  
    }  
    else  
        /* pass it to the original handler */  
        StdComment( dataKind, dataSize, dataHandle );  
}
```