WStateData structure

#include <Windows.h>

typedef struct **WStateData** {

		Size	Offset	Description
Rect	userState;	8	0	Size & position when zoomed IN
Rect	stdState;	8	8	Size & position when zoomed OUT

} **WStateData**; 16

typedef WStateData ***WStateDataPtr**;
typedef WStateData ****WStateDataHandle**;

Notes: The WStateData structure is not used in any system call. The Window Manager transparently maintains this structure of all zoom-able windows (ie, zoomDocProc and zoomNoGrow windows; see Window Types). For these types of windows, the dataHandle field of the WindowRecord contains a handle leading to this 16-byte structure.

Initially, the stdState field is set to fill most of the primary screen. It does not get changed unless you do it manually (see below).

Whenever the user changes the size of the window (via your call to **SizeWindow**), the resulting rectangle is stored in userState.

It's becoming a common practice to "remember" the user's preferred userState for documents (by saving it in a document resource), and manually put that value in place when the document is loaded. This would be most handy in a system which has two or more screens:

```
WindowPtr     myWin;
WindowPeek    myWinPeek;
WStateData    **hWSD;

myWin = GetNewWindow(DOC_WIN_ID, nil, (WindowPtr) -1)
myWinPeek=(WindowPeek)myWin;
hWSD =(WStateData **)myWinPeek->dataHandle;

**hWSD->userState = saveUsrRect; // rect used last time saved
```