ResError Find if an error occurred in a resource operation

#include <Resources.h> **Resource Manager**

short **ResError( );**
returns an Error Code from last operation (0=no error)

All resource-related functions store a result code in a low-memory global. You can use **ResError** to read that code and see if the most recent operation caused an error, and if so, what the error was.

Returns: an integer; the Error Code of the most recent resource-related operation. It may be a file system error or one of the following resource error constants:

noErr	(0)	No Error (this constant is defined in MacTypes.h)
resNotFound	(-192)	Resource not found
resFNotFound	(-193)	Resource file not found
addResFailed	(-194)	AddResource failed
rmvResFailed	(-196)	RmveResource failed
	(-197)	(not used)
resAttrErr	(-198)	Attribute does not permit operation
mapReadErr	(-199)	Error reading **resource map**

Notes: **ResError** is functionally equivalent to reading the low-memory global, ResErr; i.e., the following are the same, except the latter generates less code and is faster:

```
if ( ResError() )   { ... an error occurred ... }
if ( ResErr )       { ... an error occurred ... }
```

ResError may return other system errors, for instance, dskFulErr or memFullErr. See Error Codes for a full list.

A few **Resource Manager** functions indicate errors by returning a NIL handle (e.g., **GetResource**). When these calls fail, **ResError** returns noErr, so **be sure to check for NIL handles!**