

SetResAttrs

Set resource attributes (purgeable, locked, etc.)

#include <Resources.h>

Resource Manager

```
void      SetResAttrs(rHandle, rAttrs );
Handle    rHandle ;      handle of an existing resource
short     rAttrs ;      desired resource attribute (bit record)
```

SetResAttrs sets resource attributes in the **resource map**. The modified attributes will not take effect until the next time the resource is loaded.

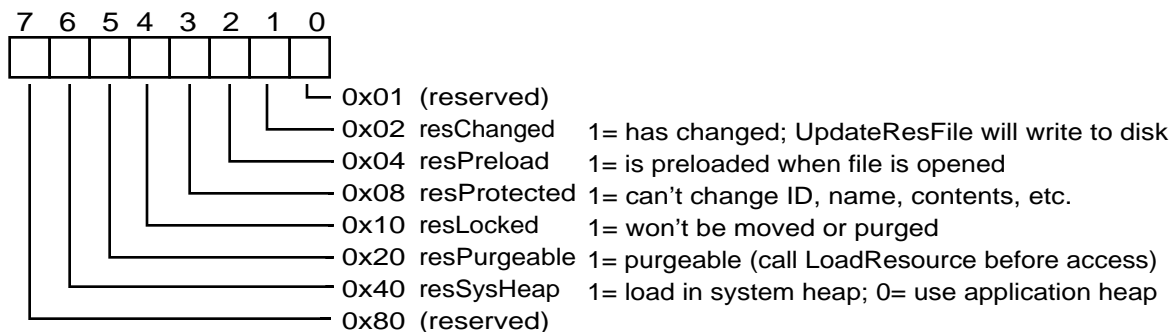
rHandle is a resource handle. It is a handle obtained via **GetResource**, **GetIndResource**, et. al.

rAttrs is a 16-bit resource attribute word - a bit record which specifies how the resource is to be handled when loaded subsequently (see below).

Returns: none

Notes: **SetResAttrs** is normally needed only by resource-management utilities such as ResEdit; it is a rare application that needs to modify attributes by using this function.

The *rAttrs* parameter specifies resource attributes as follows:



The new setting of resProtected takes effect immediately, so make sure you have already written any changes out to disk; other attributes take effect the next time the resource is loaded. We are warned specifically against modifying the state of the resChanged attribute directly. Use **ChangedResource** to flag a resource for update.

A normal sequence is to use **GetResAttrs** to find the current settings, modifying one or more bits (without changing bit 1), then use **SetResAttrs** to update the **resource map**. This is illustrated in the following example:

Example

```
#include <Resources.h>
```

```
short      theAttr;  
Handle      rHandle;
```

```
rHandle = GetResource( 'DanR', 128 );  
if ( rHandle == 0 ) { /* ... an error occurred ... */}  
theAttr = GetResAttrs( rHandle );
```

```
SetResAttrs( rHandle, theAttr | resLocked );      /* set as locked */  
LoadResource(rHandle);                          /* put into effect */
```

```
/* --- following code would force changes to be written to disk ---*/
```

```
ChangedResource( rHandle );                      /* flag for update */  
ReleaseResource( rHandle );                      /* discard for now */  
rHandle = GetResource( 'DanR', 128 );           /* this time it's locked */
```