SetResFileAttrs   Set resource file attributes

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

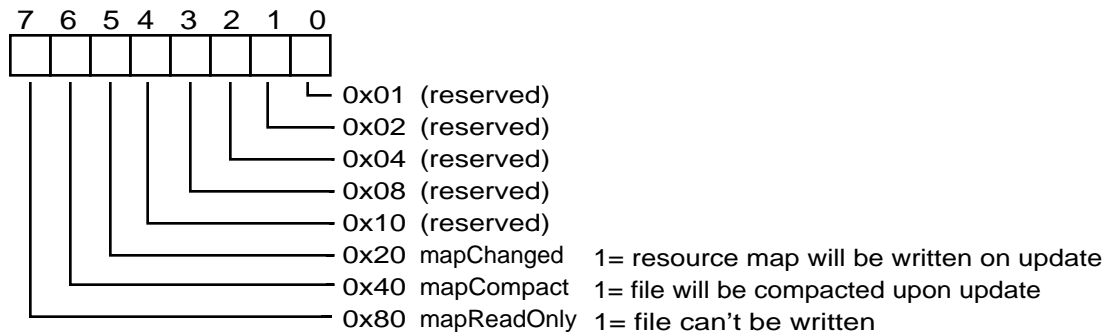| void	**SetResFileAttrs(**_rfRefNum_, _rfAttrs_ **)**;	
short	_rfRefNum_ ;	reference number of an open resource file
short	_rfAttrs_ ;	desired new attributes for the file

SetResFileAttrs sets resource file attributes. It specifies whether a resource file is marked as read-only, has changed, or needs compacting. This is needed rarely, since file attributes are modified automatically by the **Resource Manager**.

> _rfRefNum_ identifies the resource file whose attributes you wish to modify. It is a value obtained from **OpenResFile**, **HomeResFile**, **CurResFile** or **OpenRFPerm**. A value of 0 refers to the system resource file.

> _rfAttrs_ identifies the desired attributes you wish to apply to the resource file. See below..

Returns: none (use **ResError** to check for success/failure)

───

Notes: Resource file attributes are defined as follows:

```
7  6  5  4  3  2  1  0
┌──┬──┬──┬──┬──┬──┬──┬──┐
│  │  │  │  │  │  │  │  │
└──┴──┴──┴──┴──┴──┴──┴──┘
                    └─ 0x01  (reserved)
                 ──── 0x02  (reserved)
              ─────── 0x04  (reserved)
           ────────── 0x08  (reserved)
        ───────────── 0x10  (reserved)
     ──────────────── 0x20  mapChanged   1= resource map will be written on update
  ─────────────────── 0x40  mapCompact   1= file will be compacted upon update
 ──────────────────── 0x80  mapReadOnly  1= file can't be written
```

See **GetResFileAttrs** for related information.

In the 64K ROMs, it was necessary to set mapCompact manually if you wanted to recover the file space that had been occupied by a resource after it was shortened. Since this was fixed, there was little reason to modify resource file attributes.