**UseResFile**                    Make specified resource file the "current file".

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

void                **UseResFile(** *rfRefNum* **);**
short               *rfRefNum* ;        resource file reference number

   **UseResFile** selects a different file (already open) as the current resource
   file.  On subsequent resource requests, the specified file will be searched first
   and none of the more recently opened resource files will be searched.

      *rfRefNum* is a resource file reference number; typically a value obtained from
               **OpenResFile**, **HomeResFile**, or **CurResFile**.  Use 0 to specify
               the system resource file.

      **Returns**:  none (use **ResError** to determine success/failure)
   _____

 Notes:   Open resource files are arranged as a linked list; the most recently opened
          file is at the end of the list and is the first one the **Resource Manager**
          searches when looking for a resource. **UseResFile** lets you start the search
          with a file opened earlier; the file(s) following it in the list are then left out
          of the search process. When a new resource file is opened, it's added to the
          end of the list; this overrides any previous calls to **UseResFile**, causing
          the entire list of open resource files to be searched. For example, assume
          that there are four open resource files (R0 through R3) and the search
          order is R3, R2, R1, R0. If you call **UseResFile**(R2), the search order
          becomes R2, R1, R0. Note that R3 is no longer searched. If you then open a
          fifth resource file (R4), it is added to the end of the list and the search
          order becomes R4, R3, R2, R1, R0.

          **UseResFile** does not re-order the resource file list; it causes resource
          searching to start at a specified file and work backwards (chronologically)
          down the list if it fails to find a resource in the current file.

          For instance, after **UseResFile**(0), calls such as **GetResource** or
          **GetPicture** will search only the system resource file.

          The application's resource file is implicitly set as the current file when an
          application is started.  The **OpenResFile** function also sets the current
          resource file, thereby overriding any previous call to **UseResFile**.

          The resource search order is affected by the setting of the low-memory
          globals RomMapInsert and TmpResLoad.  These affect whether ROM-based
          resources are considered to be in the normal lookup list.