Chapter 12 Resources & the Resource Manager

From: tim@hoptoad.uucp (Tim Maroney)
Subject: Re: Creating a Resource Fork

In article <6956@hubcap.clemson.edu> mikeoro@hubcap.clemson.edu (Michael K O'Rourke) writes:

>No, i don't just want to know how to create a resource fork. Obviously,
>i'd just use CreateResFile. What I am trying to do is open the resource
>fork for an existing file, but a file which currently doesn't have a
>resource fork. Therefore, if i use OpenResFile or OpenRFPerm, they return
>-1 because there isn't a resource fork. If i call CreateResFile, it thinks
>I want to overwrite the existing file and it returns an error.

What error? I use CreateResFile to create resource forks for existing files and it works fine. This is actually the recommended way of using CreateResFile, due to some weirdness involving the Poor Person's Search Path. See some tech note or other (OK, I'll check -- it's 101). Of course, you have to frame it in volume-setting stuff, like:

```
GetVol -- get old volume
SetVol -- set volume of existing file
CreateResFile -- with name of existing file
SetVol -- back to old volume
```

--
Tim Maroney, Mac Software Consultant, sun!hoptoad!tim, tim@toad.com

●●●

From: mjohnson@Apple.COM (Mark B. Johnson)
Subject: Re: Interesting problem whe GetResource('dctb',id) [*LONG***]**

The following is an original posting and followups to it concerning a problem with GetResource. It is long, so press 'n' now if you aren't interested. Posted on behalf of Jim Reekes, MacDTS Ethics Officer...

- - -
dolf,

Unfortunately, you've run across a patch to _GetResource that is there to solve a Dialog Manager problem. The dctb and actb are color tables for windows. This color table resource handle cannot be purged from memory, but all other Dialog Manager resources are supposed to be purged. Normally such resource DITLs, DLOGs, etc. are copied by the Dialog Manager. The problem is that the Dialog Manager didn't copy the dctb or actb, and this lead to dialog window's color table being purged.

So, a patched to _GetResource was put into place. When a 'dctb' or 'actb' is fetched, the Resource Manager performs a _HandToHand and returns this copy. This is bad because the Resource Manager has returned a valid handle that is not a resource, and calling _GetResInfo will fail with resNotFound! The work around suggested below works, but I do not recommend it.

```
SetResLoad(FALSE);
hDCTB := GetResource('dctb', foobar); { <-- this is bad, due to patch}
SetResLoad(TRUE);
LoadResource(hDCTB);
```

Setting ResLoad to false will not load the resource data but returns a valid handle that is empty. (The handle's master pointer is NIL.) When the patch code is called to perform the _HandToHand, the routine fails. The Memory Manager will set D0 to nilHandleErr when _HandToHand is passed an empty handle.

If you followed _GetResource with _MemError, you will get the result nilHandleErr. (BTW The Resource Manager isn't too consistent returning errors, and if you called _ResError you would get noErr.) Therefore, the above code does do what you wanted. It even looks as if there's nothing wrong, but it is not a good idea. It will cause the Resource Manager patch to pass an empty handle to _HandToHand, and calling _MemError will verify this.

What I recommend is to use _Get1Resource instead of _GetResource. The former does NOT have the patch code mentioned above. This is because the Dialog Manager doesn't call _Get1Resource and _Get1Resource doesn't call _GetResource. So, the Dialog Manager patch code containing _HandToHand will not be called if you use _Get1Resource.

This patch is contained in Macs with color and applies to the dctb and actb only. I believe this patch was introduced in System 6.0x.

Jim Reekes E.O., Macintosh Developer Technical Support

--
Mark B. Johnson AppleLink: mjohnsonDeveloper Technical Support domain: mjohnson@Apple.com

●●●

From: han@apple.COM (Byron Han, Project Scapegoat)
Subject: Re: Puzzling Resource Problem

In article <1765@thumper.bellcore.com> sdh@thumper.bellcore.com (Retief of the CDT) writes:

> On a reply.good,
> it tries to open the file with OpenResourceFile() (I believe that's the
> call I may have mistyped it here), sending in the filename from the
> standard file routine. It fails and returns fnfErr (file not found) no
> matter what.
>
> So what's the deal? How do I get a resource file opened from a name
> selected from a standard file dialog?

Well, you need to go the following (pseudo code)

```
err := GetVol(@fooName, savedVRefNum);
err := SetVol(NIL, theReply.vRefNum);

newResourceFile := OpenResFile(theReply.fName);

err := SetVol(NIL, savedVRefNum);
```

This is do guarantee that you open the file from the proper working directory. SFxxxFile returns in theReply.vRefNum the working directory reference number of the folder. The WD refnum is an integer that specifies both a volume reference number and a directory ID.

Hope this helps.

Byron Han, CommToolbox Scapegoat

●●●

From: han@apple.COM (Byron Han, Project Scapegoat)
Subject: Re: Interesting problem with GetResource('dctb', id)

In article <232@fwi.uva.nl> dolf@fwi.uva.nl (Dolf Starreveld) writes:

> I have written a general purpose "CopyResource" routine. The routine itself
> was extensively tested and seemed to work, but .....
> The copying of the 'dctb' resource was only recently added and things
> started to go wrong. After lots of debugging I found out that after executing
> the following statement:
>   inHandle = GetResource('dctb', 10000);
> "inHandle" contains a valid handle, but not a handle to a resource. I entered
> TMON to find out more and:
> 1) Just before executing the statement the 'dctb' resource is not
>   loaded in the heap.
> 2) Just after the statement, the 'dctb' resource is loaded, but

```
>       immediately following it, the heap contains another relocatable
>       block of exactly the same size of the resource (48 bytes).
> 3)     The GetResource call returns a handle to this last block instead
>       of to the actually loaded resource.
```

Well, this is kind of a strange quirk in the operating system.

Do this instead:

```
        SetResLoad(FALSE);
        hDCTB := GetResource('dctb', foobar);
        SetResLoad(TRUE);
        LoadResource(hDCTB);
```

This should work in place of a simple GetResource.

Byron Han, CommToolbox Scapegoat

●●●

From: alan@Apple.COM (Alan Mimms)
Subject: Re: Interesting problem with GetResource('dctb', id)

In article <232@fwi.uva.nl> dolf@fwi.uva.nl (Dolf Starreveld) writes:

```
>started to go wrong. After lots of debugging I found out that after executing
>the following statement:
>          inHandle = GetResource('dctb', 10000);
>"inHandle" contains a valid handle, but not a handle to a resource. I entered
>TMON to find out more and:
>1)        Just before executing the statement the 'dctb' resource is not
>          loaded in the heap.
>2)        Just after the statement, the 'dctb' resource is loaded, but
>          immediately following it, the heap contains another relocatable
>          block of exactly the same size of the resource (48 bytes).
>3)        The GetResource call returns a handle to this last block instead
>          of to the actually loaded resource.
>
>--dolf
```

SOME version of the system software introduced a "feature" in which 'dctb' and 'actb' resources are CLONED when they're retrieved by GetResource and related calls. A colleague of mine discovered this recently. You MAY be able to get around this with either Get1Resource calls or with the sequence:

```
        SetResLoad (false);
        GetResource (...);
        SetResLoad (true);
        LoadResource (...);
```

Anyone in System Software care to comment?

Also, please note that you really SHOULD go ahead an upgrade to 6.0.4 final since there were a few bugs fixed in the last few versions of 6.0.4 after b15 (I THINK).

Hope this helps.
--

Alan Mimms   Communications Product Development Group

●●●

From: Steve M. Hoffman

Subject: Re: Altering the Resouce fork

I've been doing that for years. I think I read somewhere that you should never change the resources of the running program, but I've NEVER had a problem.

What I do is create a resource called PREF for my preferences. ID can be anything but I start at 0. I do a GetResource("PREF",0) and if NIL is returned then you create a default resource and write it. Then when you quit your program, call ChangedResource followed by WriteResource. You can find all of this in IM vol. 1 chapter 5. As for the data type of a PREF. It's whatever you want:

```
        myPrefHandle=^myPrefPtr;
        myPrefPtr=^myPrefRec;
        myPrefRec=RECORD
                .. your fields here ..
                end; (* record *)
```

It's really a neat thing, but do some checking after GetResource in case someone has been poking around with ResEdit where they shouldn't. If you get back an invalid PREF, just go back to the default.

Steve M. Hoffman

•••

From: lsr@Apple.COM (Larry Rosenstein)
Subject: Re: OpenResfile Question

In article <13325@unix.SRI.COM> mxmora@unix.SRI.COM (Matt Mora) writes:

> Now for the problem. OpenResFile doesn't seem to work right. It seems
> like OpenResFile wants a full pathname.

Yes.

On an HFS system, you can also use OpenRFPerm, which takes a vRefnum (could be a working directory refnum) as well as a file name. If all you have is a dirID, you'll have to open a working directory yourself.

> I tried to set the directory with PBHSetVol and then openresfile but that

Offhand, I don't know why this didn't work. There is a pitfall with PBHSetVol, in that if some piece of code subsequently calls GetVol, it will get back the root of the volume and not the folder you set. Perhaps that's what's happening here. (You can verify this by checking whether you can open a file at the root level.)

You might also consider setting ResLoad to FALSE before opening the file, just in case the application has resources that are marked preload.

Larry Rosenstein, Apple Computer, Inc.

•••

From: mxmora@unix.SRI.COM (Matthew Xavier Mora)
Subject: How to Openresfile [Summary]
Keywords: Openresfile,PBGetCatInfo,PBHSetVol,HOpenResfile

Thanks for all your help with my Openresfile question.
(Larry,Tom,John,Bill,Bruce and any others I forgot to mention)

What was I doing wrong? I wasn't using the correct dirId when I was using the PBHSETVOL function. On page 54 of the UMPG Tim Maroney writes:

In article <37560@apple.Apple.COM> keith@Apple.COM (Keith Rollin) writes:
>I think that what is happening is that your ioDirID field is being changed from

>MyCurID on the PBGetCatInfo call. Then, when you try the PBSetCatInfo call,
>you are using a DirID different from the original one.

"Yes. PBGetCatInfo changes the dirID it returns, apparently to the file ID.(Yet

> another place where this pointless feature is a pain in the ass.) It is necessary to save the dirID before calling PBGetCatInfo, then to reset the ioDirID field to this saved value before calling PBSetCatInfo. And of course, this is completely undocumented, unless you count the mysterious two-headed arrow before ioDirID in the description of the PBGetCatInfo routine."

My error was that I was using the DirID that I received back from PBGetCatinfo when I found a file that I wanted to open. I thought that the DirID was the Directory the file was in. Apparently this is not true. It must be the File ID or something. What you need to do is when you test to see if it a file or a folder,if it is a folder save that dir id and vrefnum so you can use it later. The next PBGetCatinfo call will likely change those values.

What I was doing: {in pseduo code}

```
Addnametolist(myCPB)
   myPB.dirid:=mCPB.ioDirID;  {<----WRONG}
   PBHSetVol(mypb, false);
   OpenResFile(name);
end;

enumeratecatroutine(dirid);
  PBGetCatInfo(mypb, aSync);
    if a folder then
       enumeratecatroutine(dirid);
    else
      if a_file_I_want Addnametolist(MyCPB);
    end;
 end;
```

What I should have been doing:

```
Addnametolist(myCPB)
   myPB.dirid:=CurDirID;      {curDirId is a global for the current dir}
   PBHSetVol(mypb, false);
   OpenResFile(name);
end;

enumeratecatroutine(dirid);
 PBGetCatInfo(mypb, aSync);
  if a folder then
   CurDirID=dirid
   CurVrefNum=mycbp.iovrefnum
   enumeratecatroutine(dirid);
  else
    if a_file_I_want Addnametolist(MyCPB);
  end
end
```

What I'm doing now:

```
Addnametolist(myCPB)
   SetResLoad(false); {make sure not to read in  Preload set resources }
                      {thanks Larry}
   HOpenResFile(curVrefNum, CurDirID, name, perm);
end;

enumeratecatroutine(dirid);
 PBGetCatInfo(mypb, aSync);
 if a folder then
    CurDirID=dirid
    CurVrefNum=mycbp.iovrefnum
    enumeratecatroutine(dirid);
 else
```

```
    if a_file_I_want Addnametolist(MyCPB);
  end
 end
```

Apparently there are four ways to open a resource file.

1. Openresfile (fullpathname) Probably not the recommended way

2. PPBOpenWD(pb, aSync);
   OpenRFPerm(name, vRefNum, perm);   {don't forget stripaddress on filename}

3. GetVol(OldVol, vRefNum);
   PBHSetVol(pb, aSync);
   OpenResFile(name);
   SetVol(OldVol, vRefNum);

4. HOpenResFile(curVrefNum, CurDirID, name, perm);

The last one is a new High-level call documented in tech note #218. It does the equivalent of number 3.

Now here is another question. Which one do you think would be the fastest?

Thanks again.

--
Matthew Mora | my Mac Matt_Mora@sri.com

●●●

From: oster@well.sf.ca.us (David Phillip Oster)
Subject: Re: OpenResfile Question

Inside Mac Vol 4 documents a variant of OpenResFile called something like OpenRFPerm() that takes a vrefNum. If you have a dirId, you can create a temporary wRefNum for it you can use (remember to use your own application signature, and to dispose it after you are done with it.)

-- David Phillip Oster

●●●