

Programming With Resources

KansasFest 91

**Marc Wolfgram
July 19, 1991**

<u>Type</u>	<u>Resource Name</u>	<u>Document</u>	<u>Standard Use</u>
\$8001	rIcon	TB Ref 3	rControlTemplate
\$8002	rPicture	TB Ref 3	rControlTemplate
\$8003	rControlList	TB Ref 3	rWindParam1
\$8004	rControlTemplate	TB Ref 3	rControlList and rWindParam1
	Simple Button Control		
	Check Box Control		
	Icon Button Control		
	LineEdit Control		
	List Control		
	Picture Control		
	Pop-up Control		
	Radio Button Control		
	Scroll Bar Control		
	Size Box Control		
	Static Text Control		
	TextEdit Control		
\$8005	rC1InputString	TB Ref 3	
\$8006	rPString	TB Ref 3	rControlTemplate and rWindParam1
\$8007	rStringList	TB Ref 3	
\$8008	rMenuBar	TB Ref 3	(Menu Manager)
\$8009	rMenu	TB Ref 3	rControlTemplate and rMenuBar
\$800a	rMenuItem	TB Ref 3	rMenu
\$800b	rTextForLETextBox2	TB Ref 3	rControlTemplate
\$800c	rCtlDefProc	GS TN 86	(CDEV)
\$800d	rCtlColorTable	TB Ref 3	rControlTemplate
\$800e	rWindParam1	TB Ref 3	(Window Manager)
\$800f	rWindParam2	TB Ref 3	
\$8010	rWindColor	TB Ref 3	rWindParam1
\$8011	rTextBlock	TB Ref 3	rControlTemplate
\$8012	rStyleBlock	TB Ref 3	rControlTemplate
\$8013	rToolStartup	TB Ref 3	(Tool Manager)
\$8014	rResName	TB Ref 3	(HyperCard GS)
\$8015	rAlertString	TB Ref 3	(Window Manager)
\$8016	rText	TB Ref 3	rControlTemplate
\$8017	rCodeResource	GS TN 86	(CDEV)
\$8018	rCDEVCode	GS TN 86	(CDEV)
\$8019	rCDEVFlags	FTN \$C7	(CDEV)
\$801a	rTwoRects	TB Ref 3	
\$801b	rFileType		
\$801c	rListRef	TB Ref 3	rControlTemplate
\$801d	rCString	TB Ref 3	rControlTemplate
\$801e	rXCmd	GS TN 86	(HyperCard GS and CDEV)
\$801f	rXFcn	GS TN 86	(HyperCard GS and CDEV)
\$8020	rErrorString	TB Ref 3	(Window Manager)
\$8021	rKTranTable	TB Ref 3	
\$8022	WString		
\$8023	rC1OutputString	TB Ref 3	rControlTemplate
\$8024	rSoundSample	GS TN 76	
\$8025	rTERuler	TB Ref 3	rControlTemplate
\$8026	rFSequence		
\$8027	rCursor	GS TN 76	
\$8029	rVersion	GS TN 76	
\$802A	rComment	GS TN 76	
\$c001	rRectList	GS TN 76	

Section 1 - rDialog.lib Call Descriptions:

rDialog.lib contains only two functions to support custom dialog resources with the Dialog Manager calls, LoadDialogResource and ReleaseDialogResource. Please refer to Section 2 for the resource template details.

```
=====
LoadDialogResource
```

Returns a pointer to the memory resident image of a custom rDialog resource (type \$4401). The resource and all dependent resources are loaded from the current resource chain into locked handles. The handles are then altered to reflect standard dialog template pointer references. The handles remain the property of the resource manager.

```
| previous contents |
|-----|
|_   rDialog   _|
|   resource ID   |
|-----|
|                                     |<-sp
```

```
| previous contents |      A/X contain the dialog template pointer
|-----|          ~TOOLERROR contains an error value
|                                     |<-sp
```

```
#include <rDialog.h>
dlgTempPtr LoadDialogResource(long rDialogID);
```

```
Errors: $150A - inappropriate item type.
        $1E06 - resource not found.
        memory manager errors.
```

```
=====
ReleaseDialogResource
```

Releases all resources associated back from a dialog template pointer.

```
| previous contents |
|-----|
|_ dialog template _|
|   pointer   |
|-----|
|                                     |<-sp
```

```
| previous contents |
|-----|
|                                     |<-sp
```

```
#include <rDialog.h>
void ReleaseDialogResource(dlgTempPtr theDialogTemplate);
```

```
Errors: none
```

Section 2 - rDialog.rez Custom Dialog Resource Descriptor:

This file would be included in your REZ source to define the two custom dialog resources, rDialog and rDialogItem.

```
/*-----  
rDialog.rez...  
Resource REZ descriptor file for  
rdialog.lib
```

```
Copyright 1991 by Marc Wolfgram  
--- All rights reserved ---
```

```
This file and rdialog.lib may not be  
used for commercial purposes without  
the express written consent of:
```

```
Marc W. Wolfgram  
1808 Michael Drive  
Waukesha, WI 53186  
414/549-9261
```

```
v1.0b1 Kansasfest 1991 demo release
```

```
*/
```

```
/* custom type defines ----- */
```

```
#define rDialog          $4401  
#define rDialogItem     $4402
```

```
/* dialog item type defines ----- */
```

```
#define buttonItem      $000A  
#define checkItem      $000B  
#define radioItem      $000C  
#define scrollBarItem   $000D  
#define userCtlItem     $000E  
#define statText       $000F  
#define longStatText   $0010  
#define editLine       $0011  
#define userItem       $0014  
#define userCtlItem2   $0015  
#define longStatText2  $0016  
  
#define itemDisable    $8000
```

```
/* rDialog resource ----- */

type rDialog {
    rect;          /* rect defining dialog enclosing rect */
    integer;       /* boolean - visible if true */
    longint;       /* refcon */
    array {
        longint;   /* dialog item list */
    };
    longint = 0;   /* terminating null item */
};

/* rDialogItem resource ----- */

type rDialogItem {
    integer;       /* item id - must be unique within the dialog */
    rect;          /* rect defining dialog item location and size */
    integer;       /* type (button, check box, scroll bar, etc. */
    longint;       /* descriptor reference */
    integer;       /* value */
    integer;       /* bitflag - 0 for default */
    longint = 0;   /* colortable - I don't want to mess with this */
};
```

Sample 1 - ORCA/C Custom Dialog Example:

The following code segment is the function in the demo that uses the custom dialog resources with standard toolbox calls...

```
#include <rDialog.h>

#define ExampleDialogRID 0x0000FA0L

/*****
 *
 * doExample: An example of using custom dialog resources
 *
 * Inputs:    NONE
 * Outputs:   NONE
 * Calls:     NONE
 */

doExample()
(
    word      dItem;
    DlgTempPtr dTempPtr;
    DialogPtr  dPtr;

    dTempPtr = LoadDialogResource(ExampleDialogRID);
    if (!toolerror()) (
        dPtr = GetNewModalDialog(dTempPtr);
        do (
            dItem = ModalDialog(OL);
            switch (dItem) (
                case 3:
                    SetDItemValue(0, dPtr, 4);
                    SetDItemValue(1, dPtr, 3);
                    break;
                case 4:
                    SetDItemValue(0, dPtr, 3);
                    SetDItemValue(1, dPtr, 4);
                    break;
                case 5:
                    SetDItemValue(GetDItemValue(dPtr, 5) ^ 1, dPtr, 5);
                    break;
            )
        ) while (dItem != 1);
        CloseDialog(dPtr);
        ReleaseDialogResource(dTempPtr);
    )
)
```

Sample 2:

The following REZ code segment defines the custom dialog and item templates used in the demo...

```
#define ExampleDialogRID 4000
#define ExampleButtonRID 4001
#define ExampleRadio1RID 4002
#define ExampleRadio2RID 4003
#define ExampleCheckRID 4004
#define ExampleTextRID 4005
#define ExampleLTextRID 4006

resource rDialog (ExampleDialogRID, nocrossbank) (
    ( 20,24,115,616 ),
    1, /* visible */
    0,
    (
        ExampleButtonRID,
        ExampleTextRID,
        ExampleRadio1RID,
        ExampleRadio2RID,
        ExampleCheckRID,
        ExampleLTextRID
    )
);

resource rDialogItem (ExampleButtonRID, nocrossbank) (
    1, /* item id */
    ( 12,480,25,570 ),
    buttonItem,
    ExampleButtonRID,
    0, /* value */
    0 /* default */
);

resource rPString (ExampleButtonRID, nocrossbank) (
    "OK"
);

resource rDialogItem (ExampleTextRID, nocrossbank) (
    2, /* item id */
    ( 5,20,15,470 ),
    statText + itemDisable,
    ExampleTextRID,
    0,
    0
);

resource rPString (ExampleTextRID, nocrossbank) (
    "KansasFest '91 - Programming with Resources - Marc Wolfgram";
);
```

```
resource rDialogItem (ExampleRadio1RID, nocrossbank) (
    3,                                /* item id */
    ( 20,20,30,220 ),
    radioItem,
    ExampleRadio1RID,
    1,                                /* value - set*/
    1                                /* family */
);

resource rPString (ExampleRadio1RID, nocrossbank) (
    "She loves me"
);

resource rDialogItem (ExampleRadio2RID, nocrossbank) (
    4,                                /* item id */
    ( 30,20,40,220 ),
    radioItem,
    ExampleRadio2RID,
    0,                                /* value - not set*/
    1                                /* family */
);

resource rPString (ExampleRadio2RID, nocrossbank) (
    "She loves me not"
);

resource rDialogItem (ExampleCheckRID, nocrossbank) (
    5,                                /* item id */
    ( 30,260,40,460 ),
    checkItem,
    ExampleCheckRID,
    0,                                /* value - not set*/
    0
);

resource rPString (ExampleCheckRID, nocrossbank) (
    "Apple II Forever!"
);

resource rDialogItem (ExampleLTextRID, nocrossbank) (
    6,                                /* item id */
    ( 45,20,95,574 ),
    longStatText + itemDisable,
    ExampleLTextRID,
    0,
    0
);

resource rText (ExampleLTextRID, nocrossbank) (
    "This is a standard Dialog, nothing more, nothing less. The remarkable secret\%0D"
    "about this particular dialog is that its template and the six item templates\%0D"
    "that define it are all custom resources. This text is a standard rText %8016\%0D"
    "%resource, and all the other strings are standard rPString %8006 resources...\%0D"
    "Try to use standard resources wherever possible... especially for text!"
);
```