

How to read data embedded in your program's resources

Why do it?

In a *previous article* we dealt with why and how to embed data from a file in your program's resources. If we do that we're going to want to extract it again.

How it's done

This article assumes that the data you embedded was in the *RCDATA* format. This is a user-defined data format so we also assume that you know how to decode the format.

Reading it back from the resource is very simple due to the *TResourceStream* class that ships with Delphi. You create a *TResourceStream* in one of two ways:

```
var
  RS: TResourceStream;
begin
  // Do this if the resource is named
  RS := TResourceStream.Create(
    HInstance,      // your app or DLL instance handle
    ResourceName,   // string containing resource name
    RT_RCDATA);    // identifies RCDATA resource type
  // ... or do this if it is identified by an ordinal
  // value (per my article on writing the data)
  RS := TResourceStream.CreateFromID(
    HInstance,      // as above
    ResourceID,     // Word identifier
    RT_RCDATA);    // as above
end;
```

Listing 1

You now just use the stream to read the data as if it was coming from a file. You can't write to a *TResourceStream* as data embedded in an executable file is read only.

To copy your embedded file into another stream (say *Stm* of type *TStream*) you can use this code:

```
Stm.CopyFrom(RS, RS.Size);
```

Listing 2

As an example, say we have a program that includes a resource file that contains an *RCDATA* resource which has a copy of a rich text file inside it. The program displays the embedded rich text in a rich edit component. This is the code we need:

```
var
  RS: TResourceStream;
begin
  // Create resource stream (resource id is 100)
  RS := TResourceStream.CreateFromID(HInstance,
    100, RT_RCDATA);
  try
    // Load the rich edit component
    RichEdit1.Lines.LoadFromStream(RS);
  finally
    // Free the stream
    RS.Free;
  end;
end;
```

Listing 3

Worked Example

You can download a *worked example* that demonstrates what we've described here – it uses the above code. The zip file contains a pair of projects. The first is a program that embeds a supplied rich text file in a resource file. The second program includes the resource file and displays the rich text in a rich edit component as above.

This article is copyright © Peter Johnson 2000-2005



Licensed under a *Creative Commons License*.

Copyright © Peter Johnson (DelphiDabbler) 2002-2020