

Функция **WriteComponentResFile** может быть использована для сохранения их свойств компонентов Delphi в одном файле ресурсов.

Сохранение изображений TImageList в один файл

Допустим, у Вас есть компонент **TImageList**, заполненный изображениями, каждое из этих изображений можно получить с помощью свойства

ImageIndex

. Эти изображения используются для различных компонентов, такие, как меню, пункты **TreeView, ListView** и т.д.

Если Вы хотите использовать изображения из **TImageList**, Вы можете заполнить **TImageList**

изображениями, сохранить эти изображения в одном файле ресурсов, а затем динамически загружать эти изображения из хранимого файла в динамически созданный **TImageList**

.

WriteComponentResFile

Чтобы сохранить свойства компонента (в нашем случае, это **ImageList1**) в файл, используйте метод

WriteComponentResFile

:

```
WriteComponentResFile('ImageList.dat', ImageList1);
```

ImageList1 - это имя компонента **TImageList**, который находится у Вас на форме и заполнен изображениями.

ReadComponentResFile

Чтобы прочитать свойства компонента из файла, используйте функцию **ReadComponentResFile**

Допустим, у Вас было четыре изображения в **ImageList1** и мы сохранили их в файл. Чтобы восстановить эти изображения программно, а также отобразить их в четыре **TImage**

, можно использовать следующий код:

```
procedure TForm1.Button1Click(Sender: TObject); var  imageList: TImageList; begin
imageList:= TImageList.Create(nil); try  ReadComponentResFile('ImageList.dat',
imageList);  with imageList do  begin    GetBitmap(0, Image1.Picture.Bitmap);
GetBitmap(1, Image2.Picture.Bitmap);    GetBitmap(2, Image3.Picture.Bitmap);
GetBitmap(3, Image4.Picture.Bitmap);  end; finally  FreeAndNil(imageList);  end; end;
```

Это сокращает время загрузки формы, так как изображения не загружаются вместе с формой, а будут загружены по мере необходимости. А также экономит ресурсы, так как после использования **TImageList** будет тут же уничтожен.