**Работа с инифайлами (\*.ini)**

Михаил Христосенко

В этой работе мы рассмотрим технику создания инифайлов их назначение и применение. Начнем с ответа на вопрос зачем же нужны эти инифайлы?! Предположим, что вы создали приложение, в котором пользователь может настраивать цвет фона, шрифт надписей и так далее. Когда он повторно включит вашу программу он очень сильно разочаруется, так как всего его старания по настройке интерфейса вашей программы пропали даром - программа будет иметь такой вид, который сделали вы при проектировании программы. Так вот чтобы эти настройки сохранять, лучше всего пользоваться инифайлами.

Одно из главных преимуществ инифайлов заключается в том, что эти файлы подерживают переменные разных типов (String, Integer, Boolean). В этих файлах очень удобно хранить различные настройки, например параметры шрифта, цвет фона, какие checkbox'ы выбрал пользователь и многое другое.

Теперь начнем разбираться с этими инифайлами. Для начала создайте новое приложение. Добавьте в секцию uses слово inifiles. Сохраните и откомпилируйте ваше приложение. Теперь сделаем, чтобы при каждом открытии программы форма имела такие размеры, какие установил пользователь последний раз. Для начала нам надо создать объект типа Inifile. Создается он методом Create(Filename:string); причем если в переменной Filename не указан путь к фалу, то он создаться в директории Windows, что не очень-то удобно. Поэтому мы создадим этот файл в директории нашей программы. Напишем это в обработчик события OnDestroy для формы:

procedure TForm1.FormDestroy(Sender: TObject);

var Ini: Tinifile; //необходимо создать объект, чтоб потом с ним работать

begin

Ini:=TiniFile.Create(extractfilepath(paramstr(0))+'MyIni.ini'); //создали файл в директории программы

Ini.WriteInteger('Size','Width',form1.width);

Ini.WriteInteger('Size','Height',form1.height);

Ini.WriteInteger('Position','X',form1.left);

Ini.WriteInteger('Position','Y',form1.top);

Ini.Free;

end;

Если файл с таким именем существует, то он откроется для чтения, а если нет - то он будет создан. Это очень удобно, так как не надо обрабатывать возможные исключительные ситуации, которые могут возникнуть при обращении к файлу.

Вот файл MyIni.ini после завершения работы программы (у вас естественно значения будут другими):

[Size]

Width=188

Height=144

[Position]

X=14

Y=427

Теперь подробно разберемся как записывать информацию в инифайлы:

После того, как вы создали инифайл, в него можно записывать три вида переменных: Integer, String, Boolean, это осуществляется соответствующими процедурами: WriteInteger, WriteString, WriteBool. У всех этих процедур одинаковые параметры. В общем объявление этих процедур выглядит так:

Ini.WriteInteger(const Section: string, const Ident:string, Value: Integer);

Здесь Section -это имя секции, куда будут помещены параметры и значения. В файле имена секций заключены в квадратные скобки. Обычно в секции объединяют схожие параметры.

Ident - это название параметра, которому будет присваиваться какое-нибудь значение.

Value - это собственно значение, которое будет присвоено параметру. В файле оно стоит после знака равно.

Теперь напишем обработчик события OnCreate для формы, в котором будем считывать значения из файла и изменять размеры формы в соответствии с полученными значениями. Код должен иметь такой вид:

procedure TForm1.FormCreate(Sender: TObject);

var Ini: Tinifile;

begin

Ini:=TiniFile.Create(extractfilepath(paramstr(0))+'MyIni.ini'); //открываем файл

Form1.Width:=Ini.ReadInteger('Size','Width',100); //последнее значение (100) это значение по умолчанию (default)

Form1.Height:=Ini.ReadInteger('Size','Height',100);

Form1.Left:=Ini.ReadInteger('Position','X',10);

Form1.Top:=Ini.WriteInteger('Position','Y',10);

Ini.Free;

end;

В этом коде все просто: открыли файл, прочитали из соответствующих секций необходимые параметры и присвоили их форме. Чтение значений из инифайла по сути ничем не отличается от записи в них. Указываете секцию, где хранится необходимый параметр, указываете параметр и читаете его значение. Как вы видите все просто!

Теперь я отвечу еще на один вопрос, который может появиться - почему не обычные текстовые файлы и не реестр? Отвечаю: из текстового файла очень сложно получить и обработать необходимую информацию. Многие рекомендуют для Win95/98/2000/Me, короче для всех 32-разрядных ОС использовать именно реестр, но лично я считаю, что инифайлы удобнее, так как при при переносе программы на другой компьютер, нужно перенести только один инифайл, а во-вторых, если вы что-нибудь в реестре случайно удалите, то может случиться каюк.