

## Delphi Lesson 2: Delphi Source Code

---

### Console Applications

A console application uses the command line interface instead of windows for interacting with the user (just like programs created in Pascal). To create one select New—Console Application, when it appears you will see that it has the `program` keyword at the top and this directive

```
{$APPTYPE CONSOLE}
```

you will see that the `{$` is quite common in Delphi for special directives. This application type simulates Pascal very well, however there are two important differences: the CRT unit is not available and the file handling procedures have been renamed.

### File Handling in Delphi

Some of the procedures used in Pascal for file handling have been renamed in Delphi. The changes made are `Assign( <file>, <filename> )` is now `AssignFile( <file>, <filename> )`, `Close( <file> )` is now `CloseFile( <file> )` and the `Text` variable type is now `TextFile`.

Sample code – Copying text files:

```
Var i,o : TextFile;
    s    : String;
Begin
  AssignFile(i,'text.in'); AssignFile(o,'text.out');
  Reset(i); ReWrite(o);
  While( Not EOF(i)) Do
  Begin
    ReadLn(i,S); WriteLn(o,S);
  End;
  CloseFile(i); CloseFile(o);
End.
```

### Units

Most of the code you will write in Delphi will not be in the main program but in units. Units are files which contain code for procedures and functions which can be used in Pascal programs.

Sample code – Basic Delphi unit structure:

```
Unit <unit name>

Interface
//list of procedures and functions to be shared

Implementation
//code for procedures and functions
//must include the list above, but can include others as well

Initialization //OPTIONAL
//the code here is executed when the unit is created by a program

Finalization //OPTIONAL
//the code here is executed when the unit is destroyed by a program

End.
```

Units can be used by programs and other units, using a unit in a program is fairly simple. In the **Uses** section specify `<unit name> [in <file name>]`; note that in Delphi the file name does not have to match the name of the unit, if they differ specify `in <file name>` after the unit name to indicate where the code is located.

There are two places a unit can request to use another unit: it has **Uses** subsections in **Interface** and **Implementation** sections. Requesting a unit in the **Interface** will mean that it is available in both sections, a request in the **Implementation** will only be available there. Suppose two units request each other in the **Interface**, this is referred to as a circular reference and Delphi will not compile the units, however if two units are requested in the **Interface** and the other in the **Implementation** then one unit can be compiled before the other and no conflict occurs. Also in the case that both units request in the **Implementation** no conflict will occur.

## File Types

Pascal used only one type of file (`.pas`), Delphi uses several. Delphi units are saved as `.pas` however a program is saved as `.dpr`. These DPR-files can use `.res` files with the directive

```
{$R *.res}
```

you will see this when the DPR is for a windows application, it will not be generated for a console application. Units can also make use of resources with the `{$R` directive, typically you will see

```
{$R *.dfm}
```

in units which contain the code for forms. The `.dfm` files contain the visual layout of your form.

All of these files are necessary for your Delphi project, so you will have to save all of them. Fortunately Delphi has a nice option in the file menu: Save All (Ctrl-Shift-S).