CREATING A CONSOLE APPLICATION

Console applications are pure 32-bit Windows programs that run without a graphical interface. When a console application is started, Windows creates a text-mode console window through which the user can interact with the application. These applications typically don't require much user input.

All the information a console application needs can be provided through

- a) Command line parameters,
- b) The *readin()* procedure or
- c) Text files. (e.g. input.txt)

All output will be via write(); and writeln();

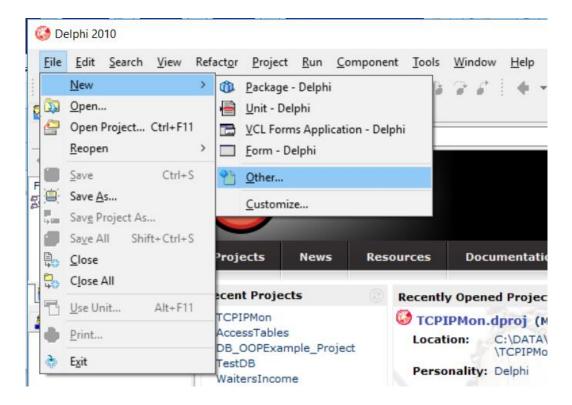
For new programmers, console applications simplify learning Pascal/Delphi - after all, the Pascal/Delphi introductory examples are nothing more than console applications.

STARTING A CONSOLE APPLICATION

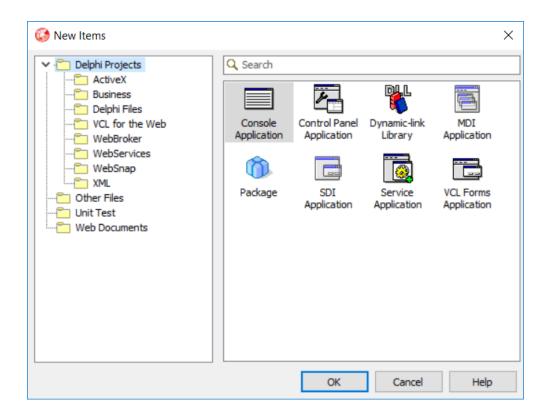
Here's how to quickly build console applications that run without a graphical interface. This assumes you are using the SA schools Delphi version 2010.

To create the application we use the Console Application Wizard by:

1. Going to File → New, this opens up a New Items dialogue, select Other



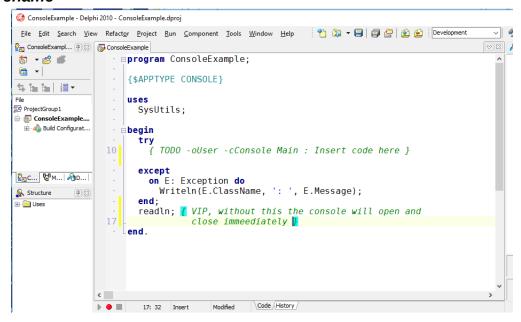
2. In the 'New items' window select the Console Application and press OK (or Double click the icon) and the wizard will set-up a Delphi project ready to be compiled as a console application.



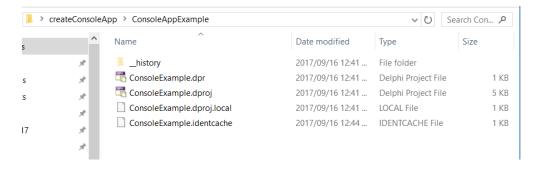
3. The console Application you will be presented with, will look like this:

```
Project1 - Delphi 2010 - Project1.dproj
                                                                            V 2 6
 File Edit Search View Refactor Project Run Component Tools Window Help Development
Project1.dproj - Pr... 📮 🖾 😭 Welcome Page 🕞 Project1
1 program Project1;
ち 智 智 温マ
                         {$APPTYPE CONSOLE}
물화 ProjectGroup1
                        uses
Project1.exe
                           SysUtils;
 Build Configurations
                      begin
[a] Pr... | Value | 1 Pr... | Value | 1 Pr... |
                             { TODO -oUser -cConsole Main : Insert code here }
                           except
& Structure
              783
                             on E: Exception do
⊕- 🧰 Uses
                               Writeln(E.ClassName, ': ', E.Message);
                           end;
                        end.
                   <
                   ▶ 🧼 ■
                                           Modified
                                                   Code History
```

4. Now select project, see highlight under ProjectGroup1, right click and select *'rename'*



- 5. Now add the *readin;* line 16 in the illustration above and save the project,
- To save press ctrl+shift+s, create the folder for your App and save the project in it.
 Note the console only has the .dproj and .dpr file (this holds the code).
 There is no .pas file.



7. Another way to use the console is to set up a template as follows. Note the readin;

