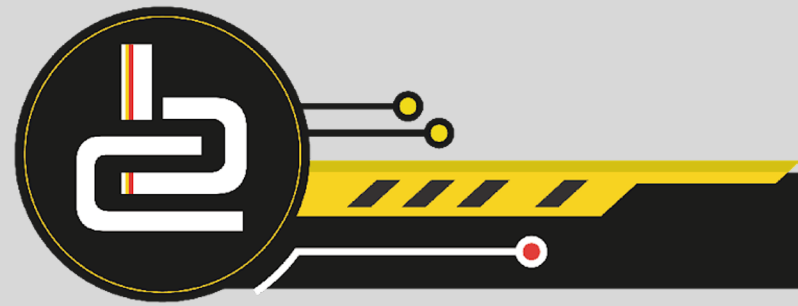




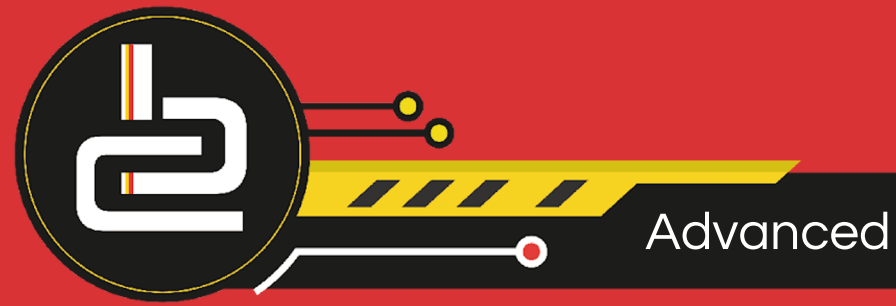
Delphi - Advanced

V. Pretorius



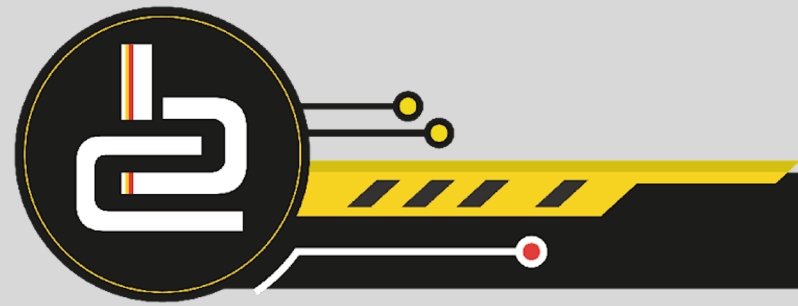
Lesson objectives

- Data Modules
- Multiple screens
- Multiple forms
- Advanced examples



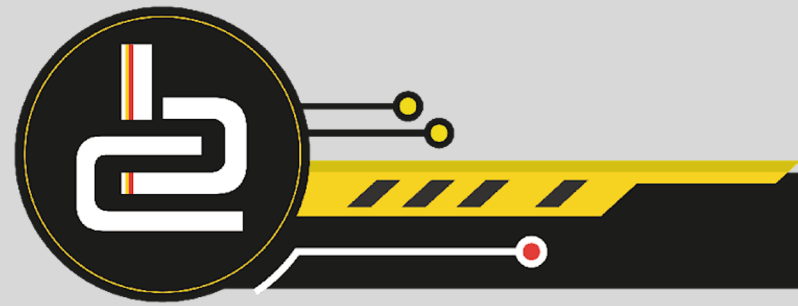
Quick Review

- In Part 1 we intensively discussed how to use the database and link it to Delphi.
- In this section we will look at it again.
- More specifically, we will look at how to group the invisible components together, so it will be easier to work with more than one screen and even use more than one form.



Data Modules

- The data module is an invisible component.
- The data module will group all the invisible components of the program.
- A data module is a specialised Delphi class used to centralise all the invisible components, and make the components available for the entire program.
- The data module ensures that you have all the access controls grouped together and it will not be necessary to duplicate it in every form.



Data Modules

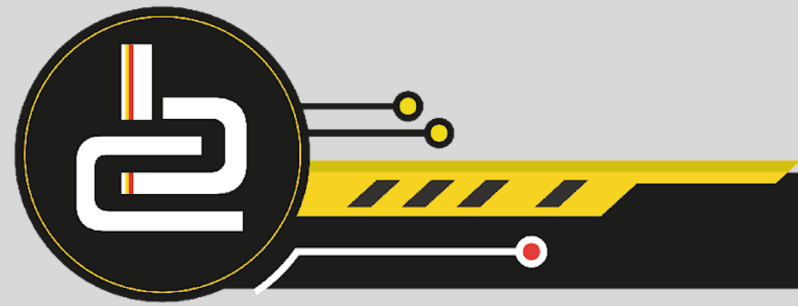
Get access to the data module as follows:

Go to File -> New -> Data Module

You will see that there is now an Additional window opened.

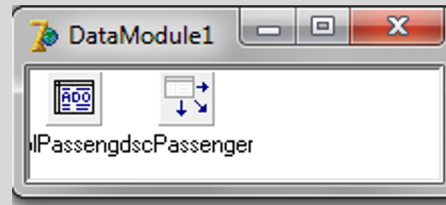
Move this window to the side so you can work comfortably

Now save the project (unit1 and project1) and the data module (unit2). Make sure you save it in such a way that you can identify it.

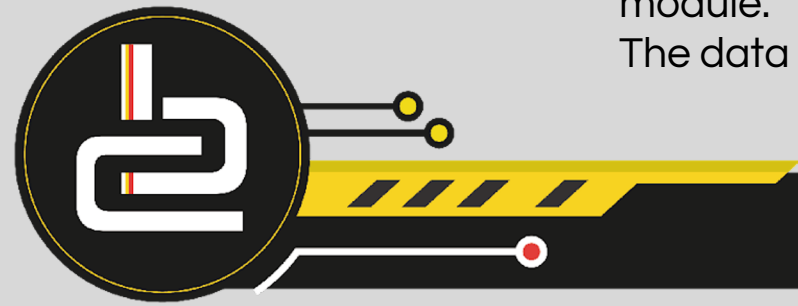


Make Use of Data Modules

- Put the ADO Table and the DataSource component on the data module.



- Change the necessary attributes of these components in the OI (as discussed in part 1)
- Now go to File and select the option Use Unit. Select the data module.
- These invisible components will now be available for this form of your project. If you want it to be visible to other forms in this project ... Stand on the new form. Then choose File -> New -> Use Unit. Select the data module. The data module will now be available for this form as well.



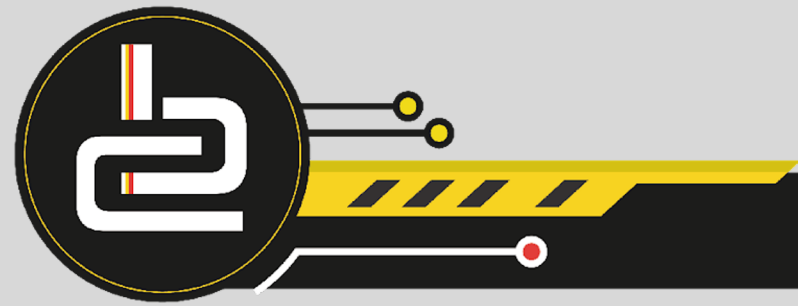
Make Use of Multiple Forms

There are several components that you can use to represent different screens, for example:

- Menu component -> find it on the Standard palette.
- The Page Control -> find it on the Win32.

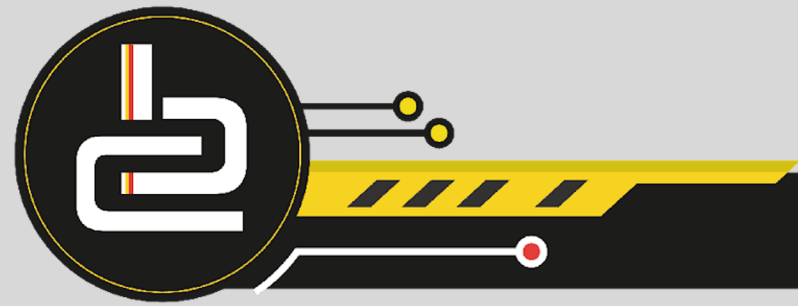


- There are several other components available to represent different screens, you can discover them yourself.



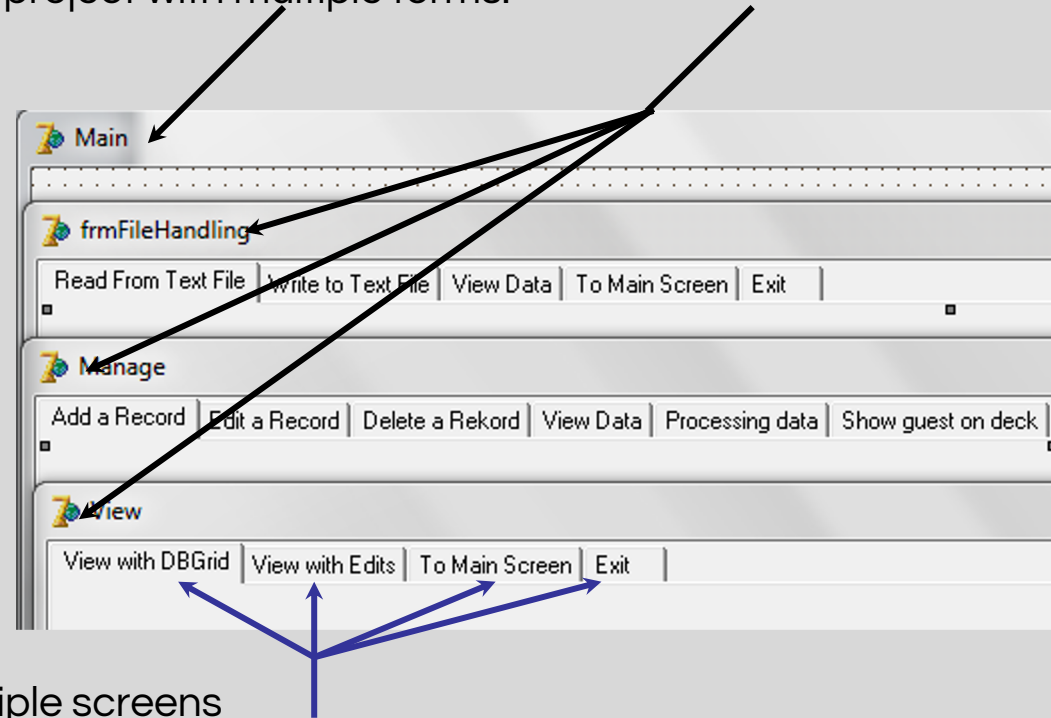
Make Use of Multiple Forms

- Open a new application.
- Save the project and its unit.
- Go to File -> New -> Form. You will see as you enter the Windows tab, your project now has two forms. Save the second unit.
- Stand somewhere in FORM1 and go to File-> Use Unit and then add the second.
- Place a button on form 1 and add the following code:
`Form2.Show;`
- Run the program. Once you click the button form 2 will appear.

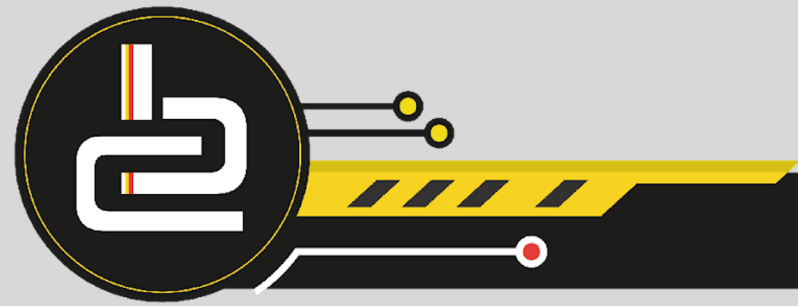


Example of Different Forms

- You now have ONE project with multiple forms.



- ONE form with multiple screens



Make Use of Multiple Forms

There are two methods to go to another form:

- `Form2.Show` this code takes you to the second form, but the form can still be accessed by the user.
- `Form2.ShowDialog` this code takes you to the next form, but the previous form is not accessible any more.

