

**Chapter
1**

**GETTING
STARTED**

*Get on the
Fast Track!*



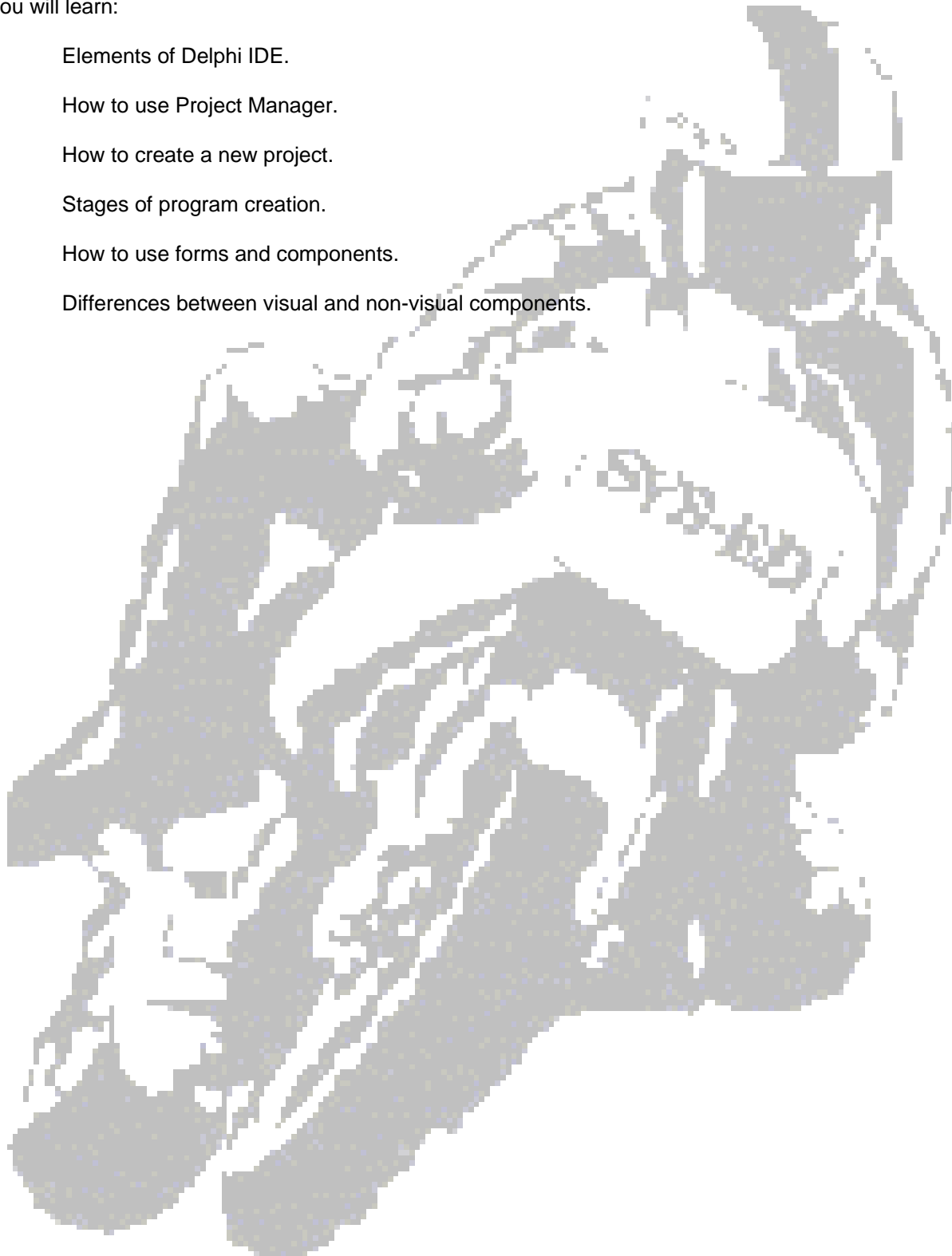
TM

**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives

You will learn:

- C Elements of Delphi IDE.
- C How to use Project Manager.
- C How to create a new project.
- C Stages of program creation.
- C How to use forms and components.
- C Differences between visual and non-visual components.



1 Delphi IDE

The Delphi IDE consists of the following elements:

C	Object Inspector Window	C	Main Window
C	Form Designer Window	C	Code Editor Window
C	Project Manager	C	Watch List Window
C	Call Stack Window	C	Breakpoint List
C	Object Browser		

Not all of these elements will be visible when Delphi is started.

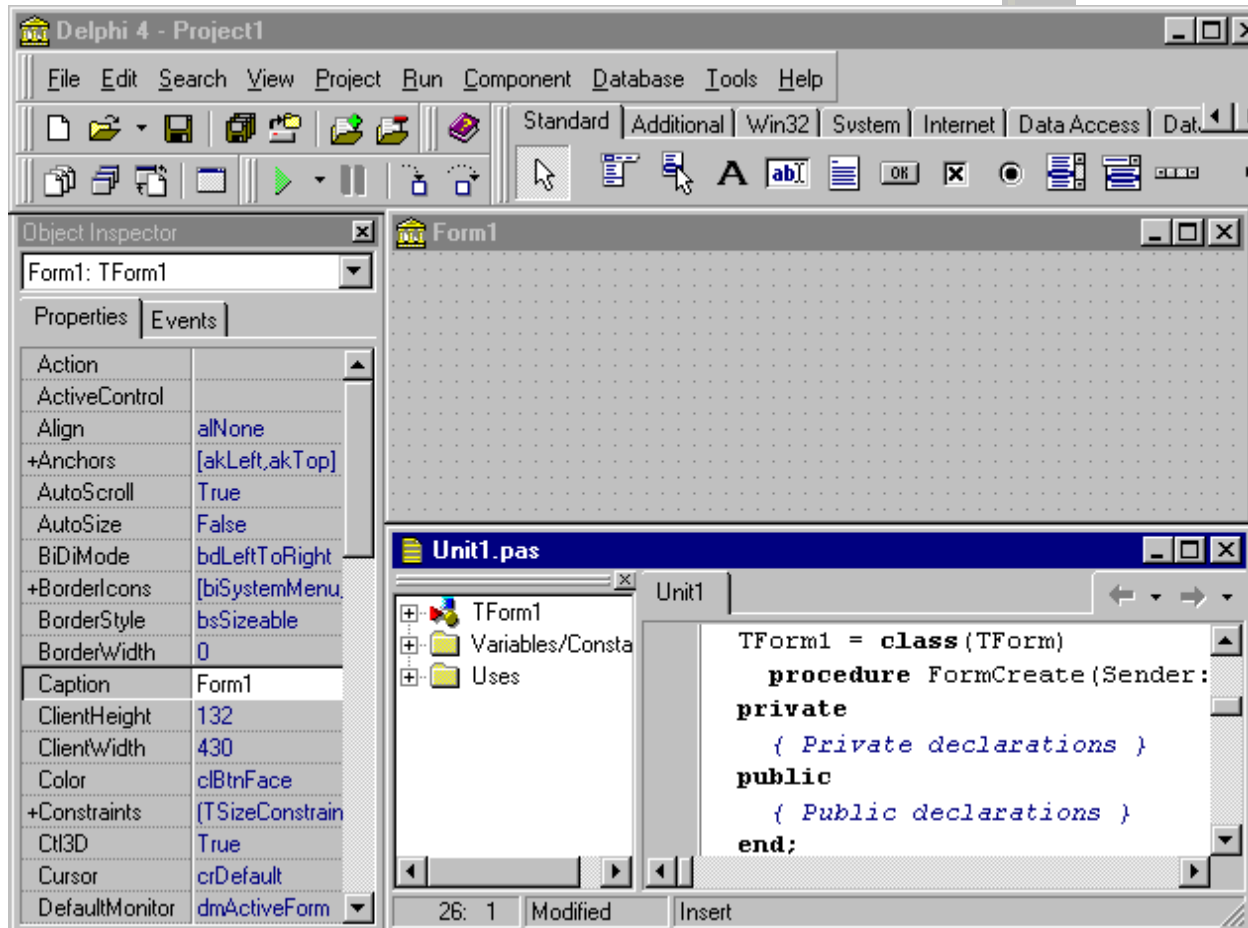
The Main Window, Object Inspector, Form Designer, Code Editor, and Project Manager typically will be the elements used most often.

1.1 Main Window

The main window is placed at the top of the screen. Minimizing the main window hides all of the other subordinate windows of the Delphi environment.

1.2 Form Designer Window

The Form Designer is the blank, gray window captioned Form1, covered with regularly arranged dots.



A form is the area of the display where various user-interface components, such as buttons, labels, and edit fields, will be placed. A form contains and owns the components placed on it and provides the framework in which cooperation among the subordinate components is achieved. Every Delphi program consists of at least one form.

The Form Designer is where all visual programming takes place. Programs are constructed by placing components selected from the Components Palette on the form. Codes are attached later to allow these components to communicate and cooperate with one another. The visual design drives the overall development process.

Even the simplest possible blank form has a number of properties that can be modified. By setting the values of these properties it is possible to customize many aspects of the form's appearance and behavior.

The current value of a property can be retrieved, and a new value for the property set during design time or run time. Setting property values at design-time provides their initial values.

1.3 Object Inspector Window

The Object Inspector displays its window along the left edge of the screen.

Inside the Object Inspector there is a two-page grid. The Object Inspector pages can be switched by clicking on one of the tabs near the bottom edge of the window, labeled Properties and Events, respectively.

1.4 The Properties Page

The Object Inspector is intimately tied to the Form Designer. The controls placed inside the Form Designer window can be selected and their properties examined in the Object Inspector.

A control is a visual component that can be placed on a form and is typically visible at run-time.

Standard Windows controls include:

C	Labels or static text	C	Pushbuttons
C	Radio buttons	C	Combo boxes
C	Edit boxes	C	Check boxes
C	List boxes	C	Scroll bars

The Properties page of the Object Inspector contains a two-column grid of names on the left and values on the right. Some of those values may appear empty initially. Each row of this grid represents one property of the currently selected component in the Form Designer.

1.5 Code Editor Window

Along with the visual design of the forms, the underlying program codes are also being developed.

Object Pascal is the underlying programming language of Delphi. Object Pascal codes are written using a text editor provided by Delphi.

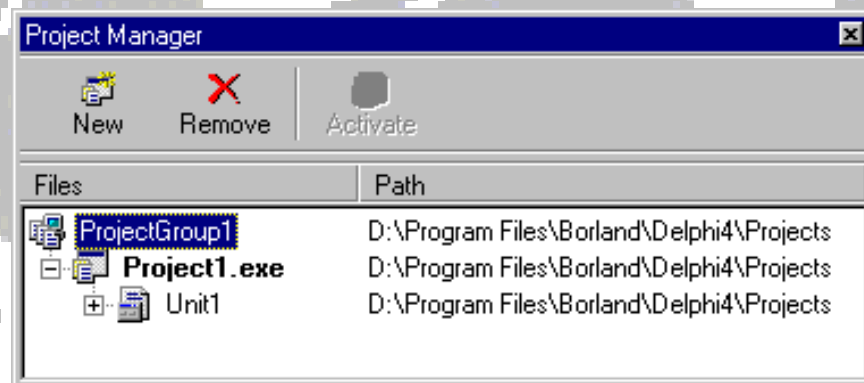
The integrated code editor is also a debugger; this allows for a programmer to:

- C single-step a program's source code.
- C examine the values of variables and properties at run-time.
- C set breakpoints.

When a new project is created or Delphi is run for the first time, a new Object Pascal source file is created in the code Editor window. This source file is an Object Pascal unit, by default named Unit1.

1.6 Project Manager

A project is the collection of all the units and forms required to develop an application. Only one project can be opened at a time, but a project can consist of a large number of units and forms.



To open the Project Manager window, select Project Manager from the Delphi main window's View menu (press Alt+V...P). The list of elements of the currently active project will be available.

Double-clicking with the mouse in the Unit column opens that unit in the corresponding form for visual editing.

The Project Manager displays a list consisting of three columns.

- C The Unit column lists all the Object Pascal modules (units) that belong to the project. Some of these units will have a visual form (window) associated with them.
- C The forms are listed in the next (middle) column, after the unit with which they are associated.
- C The third (right-hand) column shows the directory in which the corresponding Pascal source file resides.

2 Creating a New Project

To create a new project:

- C Select New Project from the File menu of the main Delphi window.

There will be some disk and display activity; the programmer is then presented with a blank form, Form1, inside the Form Designer's window, as well as a newly generated Object Pascal sourcecode unit, Unit1, which corresponds to the form, in the Code Editor window.

- C Although it is not required, the next step should be to establish a home directory for the new project by saving the project. The Delphi environment will then know where to store files it later generates.

To save the newly created project, select Save Project from the File menu. At this point, Delphi opens a File dialog box prompting for a name for the Object Pascal form unit, and the drive and directory where the form unit and the form itself will be stored.

The default name for a form is Form1. The default name for its corresponding Object Pascal unit is Unit1. These names are assigned by Delphi when a new project is created. Any default names Delphi assigns, can be changed.

Select Save Project from the File menu. Because this is a new project that has not yet been named, a Save Unit As dialog box first appears, in order for the form's unit to be specified.

Change the default unit name at this point from Unit1 to something more descriptive. The default extension .PAS will be supplied by Delphi.

After the form unit file has been renamed and the OK button of the Save Unit As dialog box is pressed, the project file itself has to be named.

Enter the name of the project in the File Name box of the Save Project As dialog box.

3 Project Files

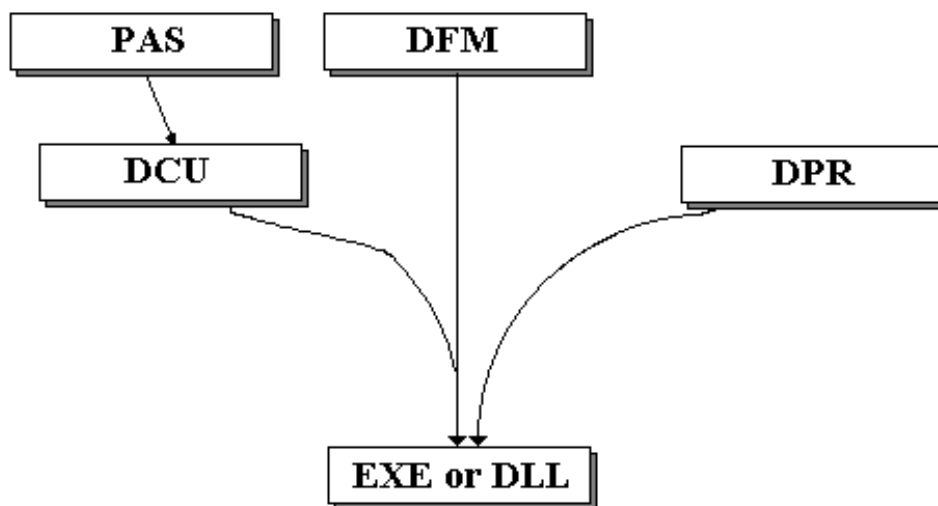
Delphi creates a number of files when a new project is saved.

File	Purpose
The main project file	This file has the extension .DPR ("Delphi Project"). It is the main Object Pascal source-code module for the project. There is only one .DPR file for each Delphi project. This file, among other things, lists the names of the other files comprising a Delphi project.
The form file	The form files have the extension .DFM ("Delphi Form"). These are binary resources files holding the definition of the visual forms as created inside the Form Designer. There may be many forms in a Delphi project, and each form has its own .DFM file.
Pascal unit file	This file has the extension .PAS and contains the Object Pascal code for the corresponding form or for a stand-alone code module. By default, there is only one such file in the project: the main form's unit. It corresponds to, and provides support for, the main form. As the projects grow, more .PAS files will be added with or without associated forms.
Project options file	The extension of .OPT marks a file that contains various Delphi settings that affect the way programs are generated and the way the Delphi IDE operates. There is one such file per project. This is a text file listing options and settings, one per line.

4 Object Files and Executables

The process of creating a complete Delphi application involves three distinct steps:

1. Creating the visual interface elements, in particular the main form of the program.
2. Writing supporting Object Pascal code to perform any needed actions on the form.
3. Compiling the Object Pascal source code and the form resources into an executable (.EXE) that can be run as a stand-alone Windows program.



These three steps will be performed iteratively many times during the course of building a project. A typical sequence of events would be to create some elements of the interface, write a little bit of code to connect them to other elements, and quickly compile and test them in a test run. This cycle of steps will be repeated each time elements are added to the interface.

4.1 Compiled Unit Files

Compiled unit files have the extension .DCU (for “Delphi Compiled Units”) and contain compiled object code of the corresponding unit (.PAS) files. The Delphi compiler creates them when either the RUN, Compile, or Build All command is issued from the Run menu. Delphi uses these .DCU files when it creates the final executable.

4.2 Compiled Program Files

Compiled program files have the extension .EXE. They are stand-alone, compiled Windows programs that can be run from the File Manager or from the Program Manager. They can also be installed inside the Program Manager’s groups.

4.3 Compiled Dynamic Link Library Files

Compiled dynamic link library files are the second type of the final executable file that Delphi is capable of creating. The default extension for this type of file is .DLL. They are compiled Windows programs.

5 Forms and Components

Forms are at the very foundation of visual programming with Delphi. Visual programming means developing programs interactively by visually placing components on forms and manipulating them with interactive, mouse-operated tools to obtain the desired and esthetically pleasing effects.

Writing code in such a scenario comes into play only after a component has been selected visually and placed on the form. This approach to programming is in contrast to the traditional code/compile/debug cycle in which the visual appearance of a program is not fully known until the first successful run of the program has been written.

Visual programming offers faster feedback to both the developer and the user. If something does not look right to the user of the program, it is much easier to change it using the interactive tools in the visual environment than it is to re-code it in the traditional fashion.

5.1 Forms

The form represents the main application window at run-time, and is a kind of “box” or “package” that holds the components of the application. A form is the owner of the components have been placed on it.

5.2 Components

A Delphi component can be placed on a Delphi form by first selecting the component from the palette and then dropping on the form in approximately the place where it should appear.

The Object Inspector can be used for rearranging the order and positions of the component on the form. This technique can also be used for changing the size and adjusting the properties of the component.

5.3 Component-Based Development

The form itself is a component. Like other components, it has a number of properties that can be changed. It is different from other components in that it is pre-created inside the Form Designer. A form can not be directly selected from the Component Palette.

A form is a very special component that, instead of being dropped on something, is being created after the New Form from Delphi's File menu has been selected.

All the other components must be explicitly selected from the Component Palette and placed on the form. The form acts as the top-level folder, or container, and the owner of all other objects placed there.

5.4 Creating Components

The Delphi environment serves a dual purpose as a software-development tool.

- C It provides for Windows programs to be built from a palette of prefabricated software elements.
- C Unlike other environments of its kind, Delphi provides for the creation of additional components that complement the existing components.

New components can be easily incorporated into the Delphi environment so that it can appear on the Component Palette just like the standard components that came with the product.

5.5 Visual versus Non-Visual Components

Non-visual components correspond to visual elements of the user interface. For instance, a Label, a button, and a Form are all visual components. They are all visible to the user when a program is running.

Delphi components do not necessarily have to be visual, however. The non-visual type of component serves other purposes. Examples may include a database, a serial communications port, a printer component, and other application elements that do not appear on the form, but rather perform services and execute actions. They are part of the project, are connected to other components via Object Pascal code, and cooperate with other components at run-time.

Even though the non-visual components cannot be seen, they are made part of the project by being placed on the form. This is done the same way as with the visual components. At design-time, Delphi shows a small, iconic, graphical representation of the non-visual component on the form. At run-time, the graphical representation is not visible.