

# Chapter 5

## Learning the Basics

---

### In This Chapter

The best way to learn about CA-Visual Objects is to use it, and this chapter lets you do just that by presenting a series of hands-on lessons that step you through the creation of an MDI, order entry application.

You will create the order entry application through completion of the following tasks:

- Build and execute the Standard Program to explore its built-in features
- CA-Visual Objects makes it easy to start developing applications by generating object-oriented code, called the *Standard Program*, each time you create a new GUI application. The Standard Program is fully functional, and it will serve as the basis for the order entry application that you will create.
- Define data servers for the new application
- A data server provides a way to interact with a database. You'll create data servers by importing a library that contains a predefined data server and associating the library with the new application. You'll then use the DB Server Editor to add a second data server to that library.
- Create a data window using the Window Editor and link it to the data servers defined in the previous step
- Add two new methods to display the data in either numeric or alphabetical order

- Customize the menus provided by the Standard Program to add three new menu items: one to open the data window created earlier and two more to implement the methods for switching between orders
- Build and execute the customized order entry application

Working through this chapter, you will create a full-featured application—complete with menus, toolbars, status bar, and event and error handling—with a minimum of time and effort. You will also gain an understanding of the skills and concepts needed to get a quick and productive start with CA-Visual Objects.

This tutorial assumes you have read through the first part of this guide, and describes only those features of the IDE that are required to create the sample application. For a complete description of all CA-Visual Objects features, please refer to the *IDE User Guide*. For more information on the programming concepts introduced here, refer to the *Programmer's Guide, Volume II*.

This tutorial is lengthy, but you can stop at any point you like. Just make sure you save your work and shut down any editors that are currently in use. Later, you can pick up where you left off. You may find the beginning of each new lesson a convenient point for taking a break.

***Important!*** Before beginning this tutorial, make sure you have installed the sample files in your CA-Visual Objects \SAMPLES\GSTUTOR directory. If the files are not installed, refer to “Installing and Starting CA-Visual Objects” in this guide for information on how to install them.

## Lesson 1: A Tour of the Standard Program

CA-Visual Objects provides a flexible, three-stage process that guides you in creating new applications. This process encourages you to think about basic design considerations by prompting you to make certain choices.

For example, do you need a single document interface (SDI) or an MDI application? Do you want to permit traditional Xbase programming techniques, such as the use of undeclared variables? What type of database access does your application require: object-oriented data server access, traditional Xbase access, or none at all?

### The Standard Program

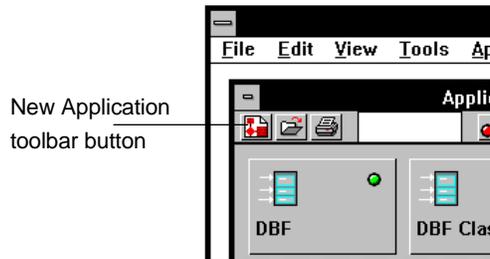
During this process, CA-Visual Objects tracks the choices you make and generates code based on those choices. The result is the Standard Program, a basic application that supports the selected styles and serves as a starting point for your own, customized application.

The purpose of the Standard Program is to form a framework upon which you build your application. When you gain more experience, you can customize the built-in standard behavior and eliminate parts of it that are irrelevant to your users.

In this lesson, you will start the process of creating a customized order entry application by generating the code for the Standard Program. Then, after running the Standard Program to see its built-in features and exploring its design, you will go on to the next lesson in which you begin customizing the application.

## Creating an Application: The Three-Stage Process

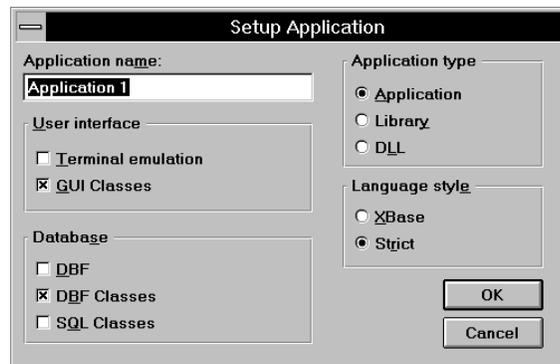
To create a new application, click the New Application toolbar button:



Clicking this button is the same as choosing the File New menu command (or pressing Alt+F, N). In CA-Visual Objects, almost every window contains toolbar buttons that serve as shortcuts for commonly used menu commands. (To quickly find out what a toolbar button does, position the mouse over it—a description appears in the status bar.)

### Stage 1

Clicking the New button displays the Setup Application dialog box, which represents the initial stage in the three-stage process:



The first stage asks you for the name of the application and its type—application, library, or DLL.

It also invites you to think about, and decide upon, basic design considerations, such as what type of user interface and database access your application will require and what language style you will use when writing the code for the application.

For this application:

1. Type **Order Entry** in the Application Name edit control.

This name will appear in the Application Browser button created for this application.

2. In the Database group box, turn on the SQL Classes check box.

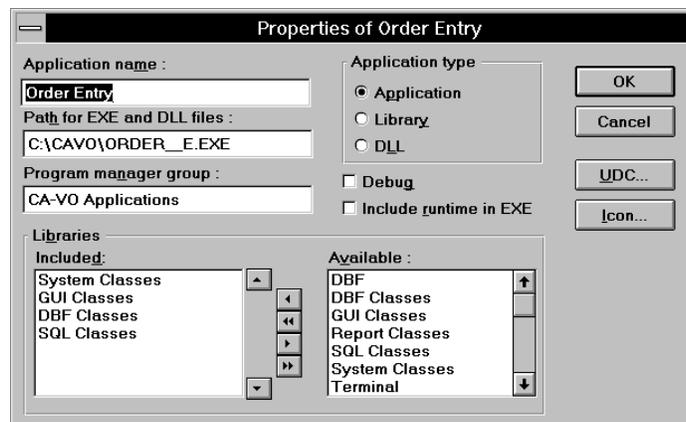
By default, the Application Type option is already set to Application—if you were creating a library or DLL, you'd have to change it here.

The default settings for the other options are also appropriate for the application you are creating. Basically, they tell CA-Visual Objects that this application will need access to the GUI Classes and DBF Classes libraries and that it will use the Strict language style (for example, all variables in the application will be declared and strongly typed).

3. Choose OK.

## Stage 2

Next, the Properties dialog box is displayed, which represents the second stage in the process:



**Note:** The Path for EXE and DLL Files may not be set to C:\CAVO (i.e., if you installed your system to another directory).

The second stage allows you to set additional options for the application, like the path and file name for the resulting .EXE file and the name of the Program Manager group to which it will belong.

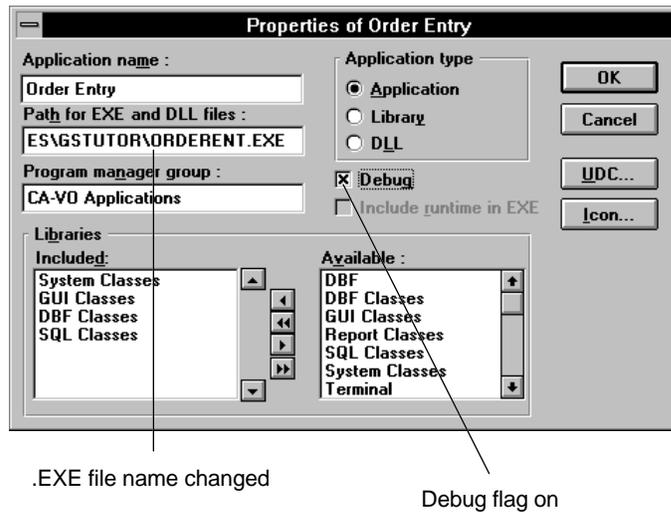
It also provides you with the opportunity to change which libraries are included in the application's search path. Including a library in an application's search path helps CA-Visual Objects resolve references during compilation. For example, if the compiler encounters a reference to a function or a constant that is not contained within any of the application's modules, it searches through the libraries defined in the application's search path before returning an error message.

In the first stage, you made some basic design choices that determined what libraries would be included in the new application's search path, which you can see reflected in this dialog box. In this stage, you can add more libraries if desired. You can also remove libraries that are already included. For example, if you are planning to display reports designed with the Report Editor in your application, you will want to add the Report Classes library.

For this application, most of the default settings in this dialog box are correct. All you need to do is:

1. In the Path for EXE and DLL Files edit control, replace the text ORDER\_\_E.EXE with the following:  
**SAMPLES\GSTUTOR\ORDERENT.EXE**
2. Turn on the Debug check box.

The Properties dialog box should now look like this:

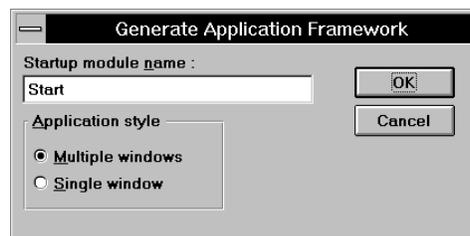


3. Choose OK.

**Note:** The choices that you make in stages one and two of this process are not necessarily permanent. They simply influence how the compiler and application options will be set *initially*. If you want to, you can change any of these choices using the Properties and Compiler Options commands on the Application menu.

Stage 3

Next, the Generate Application Framework dialog box is displayed, which represents the third and final stage in this process:



To help set up an initial framework for your application, you are prompted to name the module that will contain the startup code for this application. It is at this stage that your application acquires both an initial framework and a starting point.

For this application, all the default settings in this dialog box are correct:

1. Choose OK.

CA-Visual Objects automatically opens a Module Browser for the new application.

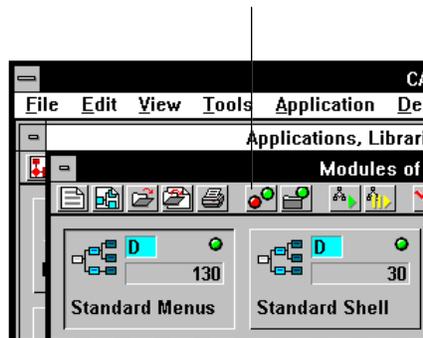
2. Choose the Window Cascade menu command to arrange the open windows in the CA-Visual Objects desktop.

## Building and Running the Standard Program

The few steps that you have taken to generate this application have given you a practical and useful starting point. Without writing even one more line of code, what you have is a working MDI application. To fully understand this application, we will provide an in-depth overview of the generated source code. However, before starting that, let's take a quick look at the application you've created.

First, click the Build toolbar button to compile the entire application:

Build button

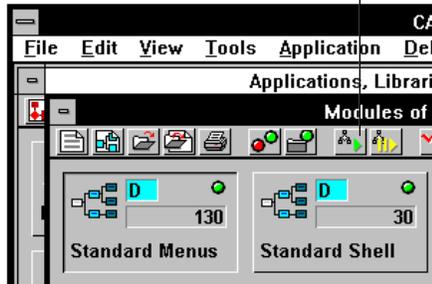


The dialog box that is displayed allows you to stop the build process at any time by clicking an Abort button. As each new entity is compiled, its name appears in the status bar.

Note that when you first build a new application, CA-Visual Objects compiles *every* entity in the application. After a successful build, a message appears in the Module Browser status bar and the compilation status indicators change from yellow to green.

You have created your first CA-Visual Objects application with just a few mouse clicks and in just a few minutes! After the build is complete, click the Execute button to run the application:

Execute button



After a few seconds, CA-Visual Objects opens a new application window titled "Standard MDI Application." Later in this lesson, we will run the application again and explore it more fully, but for now, choose the File Exit menu command, and we'll take a closer look at the CA-Visual Objects desktop and customize the application a bit.