

## Lesson 2: Setting Up the Data Servers

Up until now, we've called this new application the Standard Program because, after all, it has hardly anything to do with order entry yet. In fact, the only code you've written so far is to change the title bar caption of the shell window. Starting with this lesson, however, the changes you will make will be more significant; and we will, therefore, refer to the application as Order Entry from now on.

In this lesson, you will set up data servers for two database files: CUSTOMER.DBF and ORDERS.DBF. The concept of a data server was mentioned briefly in the previous lesson because the Standard Program uses them to interact with the self-configuring data windows that it creates.

Basically, a data server is a high-level, abstract entity designed to give you a consistent OOP interface for your database files and, more importantly, to allow them to interact with data windows. The data server acts as a database describer, defining its file (or table) name, specifying the order in which it is accessed, and providing a mechanism for extending field definitions beyond the basic name, type, and length information stored in the database file structure (this last part is accomplished using a *field specification*).

As you work your way through the development cycle, you can make on-the-fly changes to a data server (for example, changing the validation or formatting rules for one or more fields). CA-Visual Objects ensures that these changes are reflected in all appropriate places, such as a window that is associated with that data server or another data server that is sharing the same field specification.

The purpose of this lesson is to provide the basis for creating a customized, master-detail data window in the next lesson. The information stored in these data servers will be automatically picked up by the data window, and you will see the results when you connect that window to a menu command and run the new application.

### Importing the OE Data Servers Library

To make this lesson as quick and simple as possible, the data server for CUSTOMER.DBF is already defined for you. It resides in a

separate library that is stored on disk as an Application Export Format (.AEF) file, rather than in the repository.

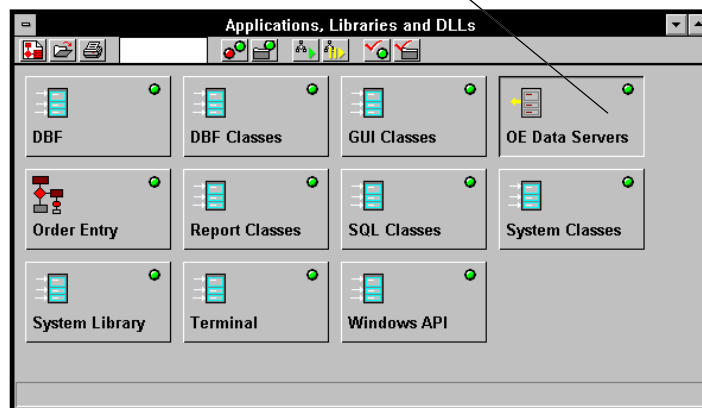
**Note:** The applications and libraries you create in CA-Visual Objects can be stored external to the repository as .AEF files using the File Export command to *export* them. When desired, you can *import* them back into the repository with the File Import command.

To access the predefined data server, you need to import the OE Data Servers library as follows:

1. Switch to the Application Browser (click it with the mouse or choose it from the list of names displayed on the Window menu).
2. Choose the File Import command.
3. From the resulting dialog box, select the OESRVR.AEF file from the CA-Visual Objects \SAMPLES\GSTUTOR directory and choose OK.

CA-Visual Objects creates the library and adds a button for it (labeled “OE Data Servers”) to the Application Browser:

New button added for the imported file



**Note:** You may receive a dialog box warning you that there is a conflict between the STD.UDC file associated with this .AEF and the STD.UDC on your hard drive. If so, click on the Keep Existing button to use the installed version of STD.UDC.

## A Quick Tour of the Customer Data Server

Before you move on to create a data server for the ORDERS.DBF file, it will help to first take a look at the data server you just imported. It was created for the CUSTOMER.DBF file using the DB Server Editor, the tool in the CA-Visual Objects IDE for creating DBF-style data servers.

### Loading the Customer Data Server

To load the Customer data server in the DB Server Editor:

1. Double-click on the button representing the OE Data Servers library.

CA-Visual Objects opens the library's Module Browser.

2. Double-click on the Customer module button to display a list of entities in that module:

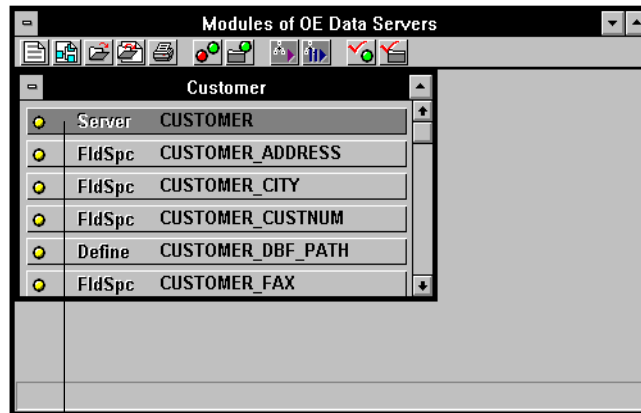
There are lots of entities in this module, but we didn't write a single line of code to create this data server. All the code you see was generated by the DB Server Editor, which you are about to launch.

**Note:** To display the *prototype* of any entity in the status bar of the Module Browser, scroll down the list of entities and click on one so that it is highlighted. This feature of CA-Visual Objects allows you to view information about a *source code* entity without having to open it up in the Source Code Editor.

3. Scroll down in the Entity Browser until you see the Customer *class* entity, and double-click on it.

This will load the source code for the Customer class declaration in the Source Code Editor. The Customer class has a single instance variable, CUSTOMER\_DBF\_PATH, that defines the drive and path name used to locate all files associated with this data server (i.e., CUSTOMER.DBF and all of its index files).

4. If needed, change the drive/path name to match your own, double-click on the system menu to close the editor, and click the Yes button when prompted to store the changed entity.
5. Double-click on the Customer server entity to load it in the DB Server Editor:



Double-click here

6. The DB Server Editor window appears.