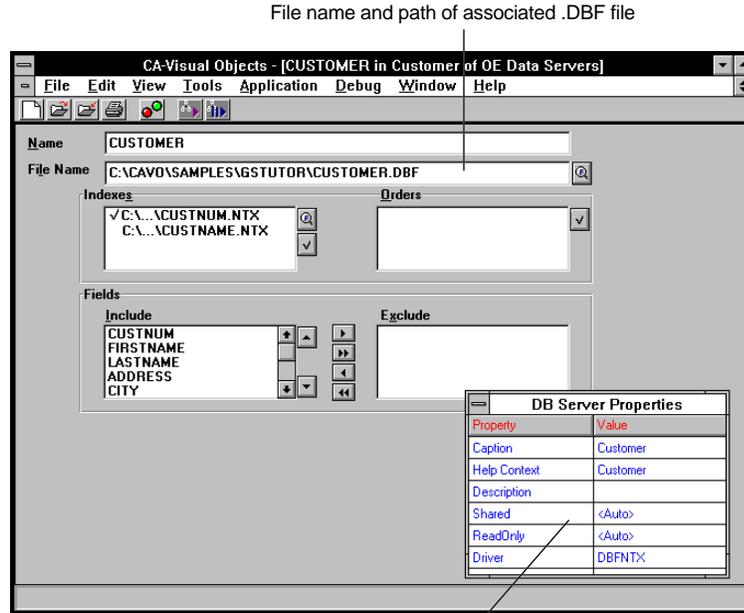


Your desktop should now look as follows after maximizing the DB Server Editor:



Properties window (currently set for the data server)

The first thing you will notice about the DB Server Editor is how simple its desktop is—just a few edit controls and list boxes, and the floating Properties window. (You’re going to see a lot of this Properties window—it is featured in almost every visual editor.)

The simplicity of this interface makes it easy to design and create data servers (as you will see later). For example, in the File Name edit control, you can see that this data server is associated with CUSTOMER.DBF (it also knows the location of the file, C:\CAVO\SAMPLES\GSTUTOR, which was defined earlier when you created the Order Entry application).

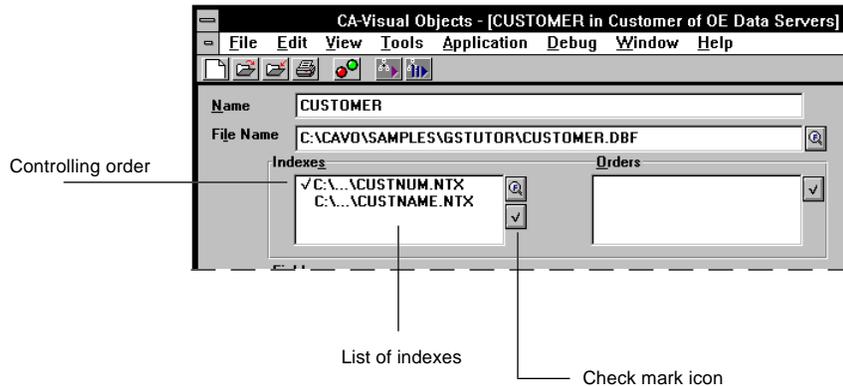
**Note:** If you installed CA-Visual Objects to any drive or directory other than C:\CAVO, then you must edit the text in the File Name edit control to show the proper path.

In addition, if you look at the Properties window, you can see the various properties that can be defined for a data server, including file open modes (Shared and ReadOnly) and the name of the associated RDD (Driver).

### The Indexes List Box

In the Standard Program, you could open database files only—you were warned against editing data in indexed database files because the corresponding index files were not opened. By defining your own customized data servers, you can associate as many index files as you like. You can also select the controlling order.

For the Customer data server, there are two index files in use, CUSTNUM.NTX and CUSTNAME.NTX. These files are listed in the Indexes list box—any time you use this data server, both index files will be updated with any changes made to the database file.

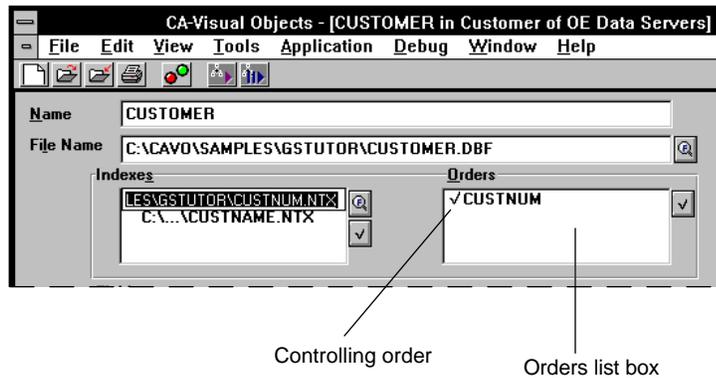


As you can see, CUSTNUM.NTX has a check mark next to it. The check mark indicates that this index file contains the controlling order. Now click on CUSTNUM.NTX.

**Note:** If you did not install CA-Visual Objects to the C:\CAVO directory, you will have to edit the Filename property for both indexes, which is contained in the Index Properties window.

## The Orders List Box

The Properties window changes so that it displays properties related to the index file, and CUSTNUM appears in the Orders list box with a check mark. The Orders list box shows all orders defined for CUSTNUM.NTX and further defines which of them is the controlling order:



Click on CUSTNUM in the Orders list box. Again, the Properties window changes, this time displaying properties related to the individual order.

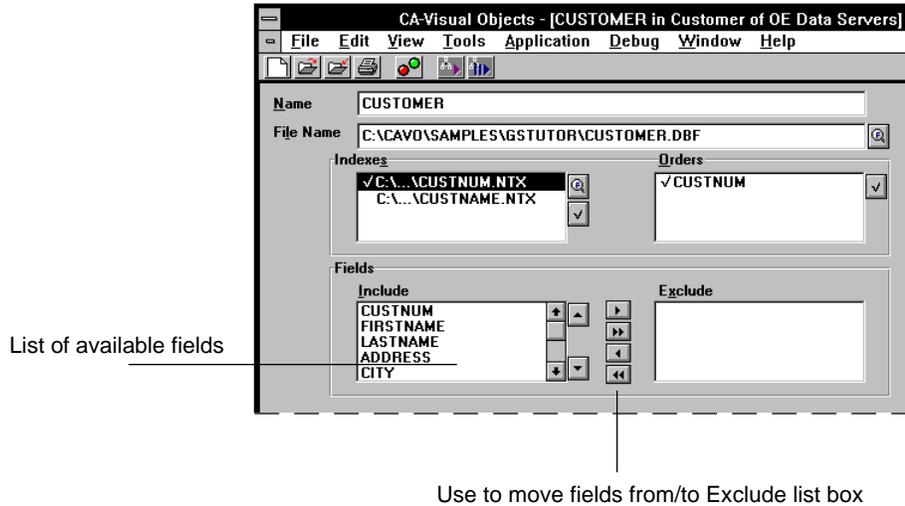
Because we are using .NTX index files, which support only one order per file, you can quickly choose a new controlling order by highlighting its associated index file in the Indexes list box and clicking on the button with the check mark icon.

**Note:** If you were working with an RDD that supported multiple orders per index file, the Orders list box would show all the orders defined in the highlighted index file. Thus, to define the controlling order, you would check the index file that contained the controlling order in the Indexes list box, then check the appropriate controlling order in the Orders list box.

However, you'll add menu commands and methods to the Order Entry application later in Lesson 5 of this tutorial, which will allow the user to switch from one sort order to the other in the resulting application.

## The Fields Group Box

The DB Server Editor lists all fields in the data server in the Include list box (which is in the Fields group box):



**Include versus Exclude** This list box is used by CA-Visual Objects to determine which fields in the database file should be accessible when this data server is used.

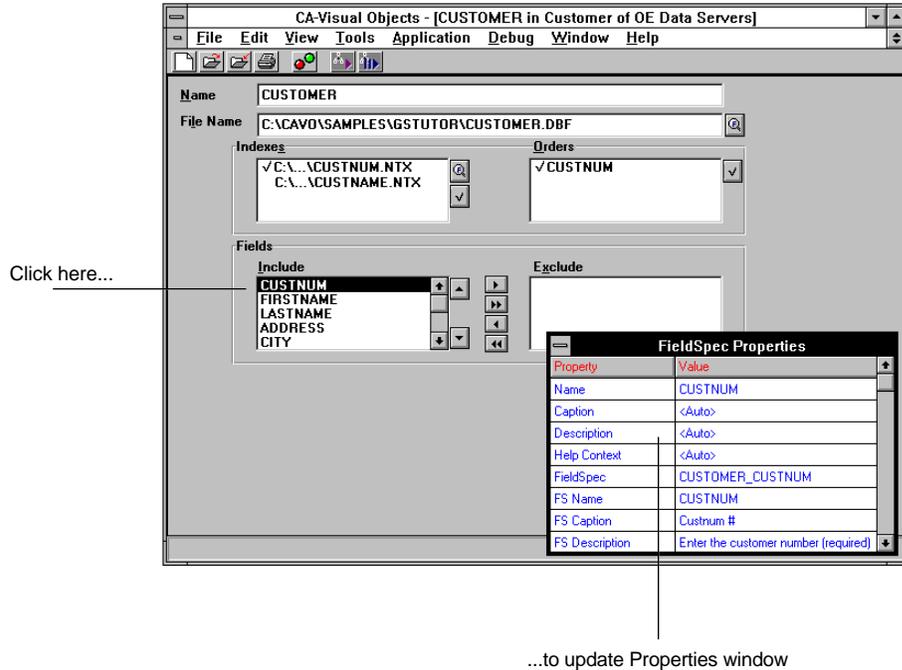
If desired, you can use the Left and Right arrow buttons to the right of the Include list box to move one or more fields to (or from) the Exclude list box. Fields that are moved to the Exclude list box will be inaccessible when using this data server, and a reference to an excluded field in your application will result in a runtime error.

**Field Order** If desired, you can use the Up and Down arrows to the right of the Include list box to reorder the fields (for example, move one in the middle to the top or bottom of the list).

The order in which fields appear in the Include list box determines the order in which they will appear when you use the Auto Layout feature of the Window Editor to define a data window for this data server (more on this later). (It also determines the order in which the fields are created if you generate a .DBF file using the File Export command.)

Field Properties

Finally, take a look at the properties available to a field. Click on the CustNum field, and watch as the Properties window changes to reflect field-related properties (and those defined for the CustNum field in particular):



Fields in a data server have many properties (such as picture and validation), all of which are shown in the Properties window when a field in the Include list box has focus. Note that these properties are not part of the physical structure of the underlying database file, although some of them (such as type and length) have counterparts in the file structure.

The data server uses something called a field specification (derived from the FieldSpec class) to group *all* the properties of a field into a single, logical entity, so that the information pertinent to a database field can be conveniently located. This arrangement has several advantages, but the most important ones are listed below:

- You can reuse a field specification's information in other data servers. For example, in many cases, the different data servers your application uses contain similar, if not identical, fields (for example, all zip code fields are typically the same, regardless of where they are used).

Using a field specification, you can create just one field specification and reuse it in each data server that needs it. Thus, multiple data servers can access the same property values for common fields.

- A field specification's properties are automatically inherited by data windows. Many of the properties that you define for a field specification in a data server are designed to be used by data windows that you create using the Window Editor. Thus, you need only define the attributes once, and they will be automatically inherited and used by any data window that is linked to that data server.
- Even though it may be used in many different places (for example, multiple data servers and data windows), you can make changes to a field specification quickly and easily, and in just one place.

Changing a field's properties is easy, even if the field specification is used by numerous data windows and data servers. If you change a field specification (whether in the DB Server Editor, the FieldSpec Editor, or the Window Editor), the CA-Visual Objects repository ensures that the change is automatically propagated to all other entities that use it. (You'll see this later on in this lesson.)

With that explanation, the tour of the Customer data server is concluded. Let's move on to create another data server, this time for the ORDERS.DBF file. By doing so, you'll get a better flavor of field specifications by reusing some of the ones that were already defined for the Customer data server. You will also see how changes are automatically propagated when field specifications are modified.

Before moving on, close the DB Server Editor by double-clicking on its system menu and, if you have made any changes (other than to the drive and directory in the File Name edit control), please do not save them at this time. You can also close the Customer Entity Browser.

## Creating the Orders Data Server

The process that you are about to go through to create the Orders data server is remarkably similar to the process we went through to create the Customer data server that you imported in the previous step.

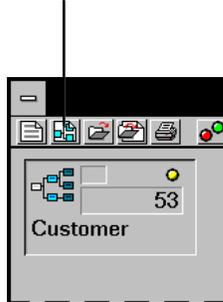
In this section, you will explore some of the features of the DB Server Editor hinted at in the previous section and will get a better understanding of how this editor works.

### Starting the DB Server Editor

Assuming you are in the Module Browser for the OE Data Servers library, perform the following steps to create the Orders data server:

1. Click on the New Module toolbar button:

New Module button



The Create Module dialog box appears:

