Lesson 3: Creating a Data Window

In this lesson, you will use the Window Editor to create a data window. As explained earlier, a *data window* is a type of window with which you can associate one or more data servers. Data windows are preconfigured so that they know how to display and operate on the information extracted from their underlying data servers.

Master-Detail

For the Order Entry application, we'll create a data window that "links" the Customer and Orders data servers in a *master-detail* fashion. Recall that CustNum is a field that is common to both the Customer and Orders data servers. In addition, the Orders data server uses the CustNum field as its controlling order key.

Thus, the two data servers can be linked together, based on the contents of this common field, in master-detail fashion. When data servers are linked in a master-detail fashion, the detail data server is synchronized with the master data server: the detail data server is automatically repositioned whenever the master data server moves to a new record. Thus, the data windows containing the two data servers will display only orders whose customer number matches that of the current customer record.

Note: The link between the two data servers is established using the DataWindow:SetSelectiveRelation() method.

Auto Layout

To create the data window, you'll use the Window Editor's Auto Layout feature. Using this feature will give you a productive start on your data window design and will also demonstrate how a great deal of the information you defined for the two data servers is automatically picked up by the data window.

Starting the Window Editor

To create a data window for the Order Entry application:

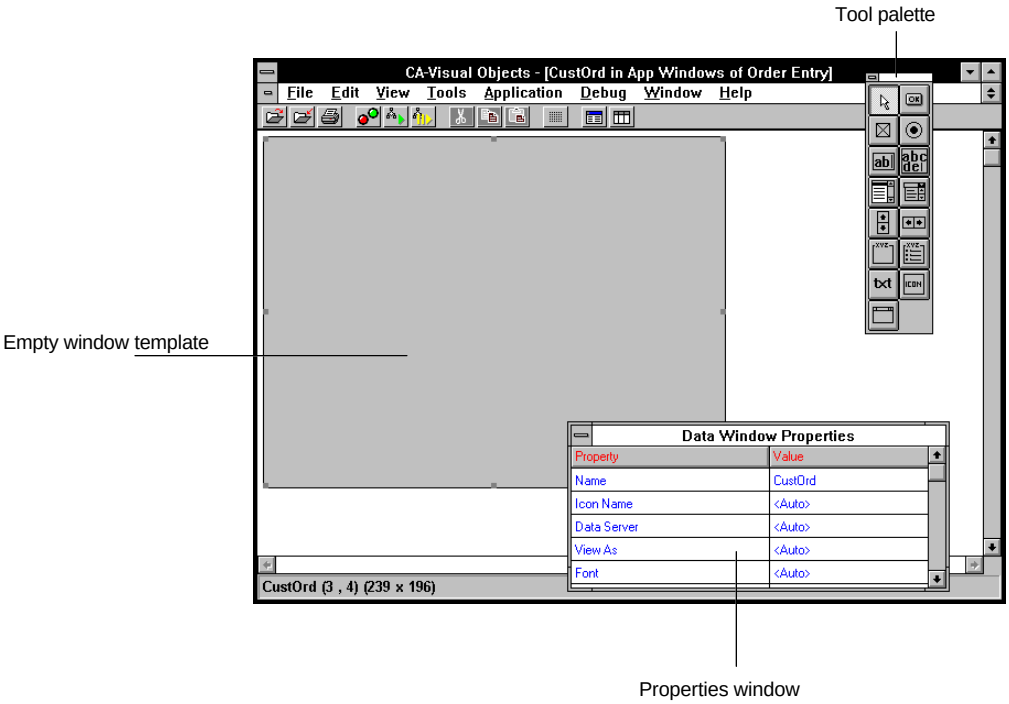
1. Open up Order Entry's Module Browser by double-clicking on the Order Entry button in the Application Browser.
2. Click on the New Module button, and in the Create Module dialog box, type **App Windows**, and choose OK.

   Note: If you don't remember where the New Module button is, simply move the mouse over the toolbar to display a quick reminder in the status bar.
3. Click on the New Entity toolbar button, and from the local pop-up menu, choose Window Editor.

   The following dialog box is displayed; it allows you to choose what type of window to create, as well as enter a name for it:

4. Since you are creating a DataWindow, which is the default selection, all you need to do is type **CustOrd** in the Name edit control.
5. Choose OK to launch the Window Editor.
6. If desired, reposition the Properties window and/or tool palette.

Your desktop should now look something like the following:



When you first access the Window Editor for a new window, you see an empty window template on which you can place controls, a floating tool palette from which to select the type of controls to place, and a Properties window to define properties for the data window and the various controls.

Window Properties

The Properties window probably looks familiar to you, because it is very similar to the one you used when working with the DB Server Editor. Like the DB Server Editor's Properties window, this one also changes depending on what currently has focus in the editor (for example, a window or a control).

Initially, the data window's blank template is selected, and, therefore, the Properties window displays properties for the data window. Let's take a look at some of the properties available to a data window.

Name

The first thing you see in the Properties window is the name of the data window, CustOrd. This is the name that you entered earlier when you launched the Window Editor.

View As

The View As property controls how the window displays data when it is initially opened. You can choose either form view or browse view, which you are familiar with from working with the data windows in the Standard Program back in the first lesson of this tutorial. Later in this lesson, this property will be filled in by the Auto Layout feature.

Data Server

Data Server is also a property of a data window. Like View As, this property will be filled in by the Auto Layout feature later in the lesson.

Menu

Now let's take a look at the Menu property. If you remember from working with the Standard Program, each child data window had its own menu (StandardShellMenu) that replaced the shell window's menu (EmptyShellMenu) when the child window was open and had focus.