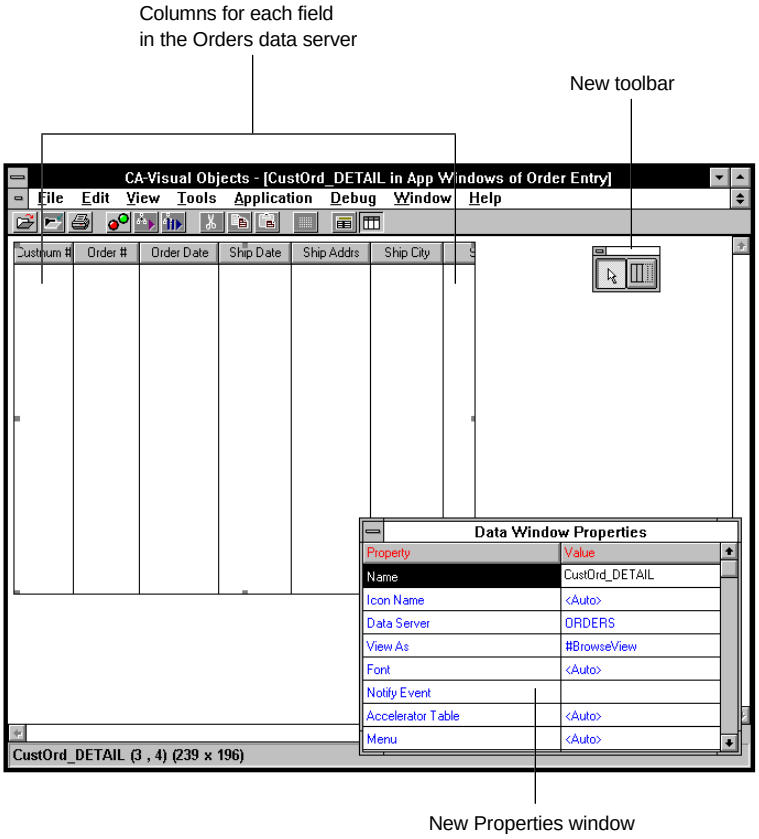
Additionally, the Style property allows you to select the Tab Stop option from the Sub-Form Styles dialog box in order to enable the tab key for the sub-data window. Lastly, the Generate Code option indicates whether or not source code will be generated for the sub-data window control.

Opening the Sub-Data Window

There are, in fact, additional properties that are available to the sub-data window. To view them, you must double-click on the sub-data window. Doing so actually launches a new copy of the Window Editor, unique to the sub-data window.

As you can see, you get both a new Properties window and a new tool palette, which allows you to add columns. There are many more properties available now—you may notice that this sub-data window is associated with the Orders data server and its default view is browse.

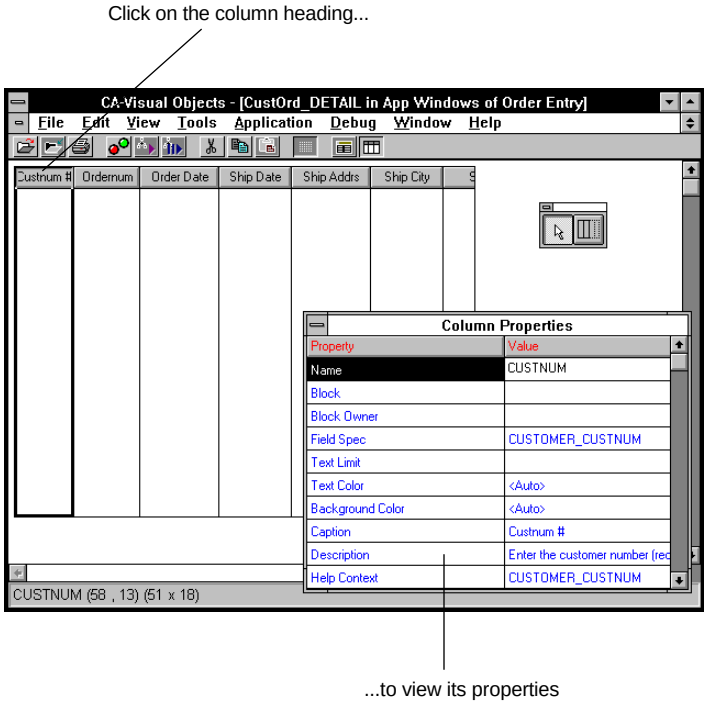


Browse View

Recall from Lesson 1 that when a data window is displayed in browse view, it displays each field in the underlying data server as a column. This is in contrast to form view (the view used for the main data window) in which each field in the underlying data server gets a fixed text control and an edit control.

Note that the column headings in this browse view are extracted from the captions defined for each field in the underlying data server (reusing your work in the DB Server Editor).

This is because, like the single-line edit controls in the main data window, these columns represent the fields themselves. If you click on a column heading, you can see that the properties defined for its associated data server field while in the DB Server Editor are reused here:



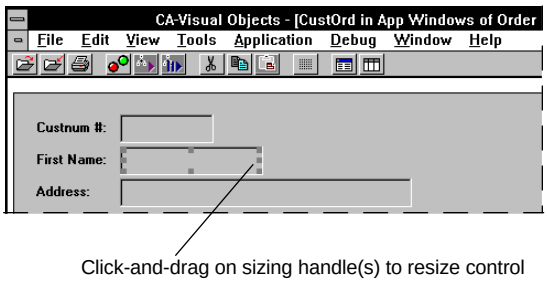
Finally, note that viewing the sub-data window in this workspace gives you an idea of how that rectangular control in the main data window will look at runtime. (The rectangular control is used for the size and placement; this view is used to illustrate the column heading text, spacing, etc.) Close the sub-data window by double-clicking on its system menu and choose No if you are prompted to save changes.

Customizing Windows

The master-detail window that you just created simply by using the Auto Layout feature is fully operational and ready to use. You may, however, choose to modify it by resizing it or moving or resizing any of its controls, or by changing its properties or the properties of any control associated with it.

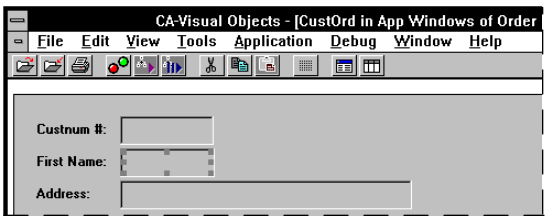
Sizing Windows/Controls

For example, you've already resized the main data window by clicking and dragging on one of its sizing handles so that the sub-data window would be fully visible. Similarly, you can resize its controls using the same technique:



Note: Some controls, such as radio buttons, have a fixed size.

In our CustOrd example, resize of the First Name edit control so that it is the same size as the Custnum # edit control, as follows:



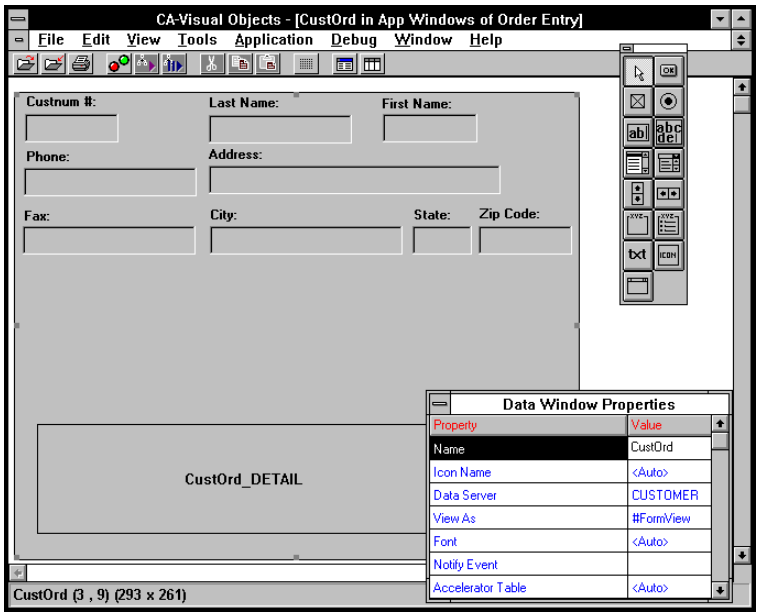
Moving Controls

Perhaps you would like the controls to be arranged across the top of the main data window, with Last Name placed before First Name, since most customer records are maintained in alphabetical order by last name. Also, you might like Phone and Fax placed under Custnum #.

First, if necessary, resize the window to allow the controls to fit in the window horizontally. Then, to arrange your controls:

1. Place the mouse pointer anywhere on a control.
2. Press the left mouse button and hold it down.
3. Drag the mouse to move the control to the desired location, and release the mouse button.
4. Repeat steps 1 to 3 for all fixed text and edit controls.

The CustOrd data window should now look something like this:



Note that now there is more room for the sub-data window so you can even resize that, if you like.

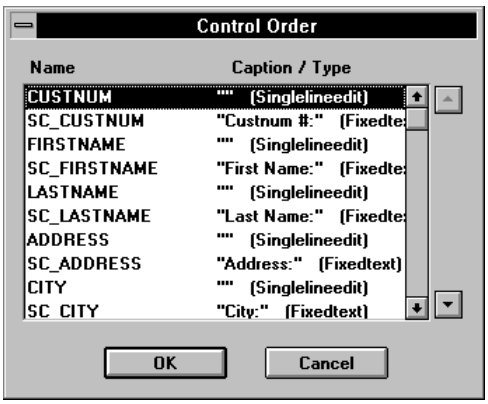
Positioning Controls

There are a number of useful commands on the Edit Arrange menu that will allow you to arrange these controls so that they have the same alignment, size, or spacing. These include such commands as Align Left, Align Right, Center Vertically, Center Horizontally, Even Vertical Spacing, etc. See the Modifying a Window in the "Using the Window Editor" chapter of the *IDE User Guide* for more information about moving, resizing, and positioning controls.

Viewing Tab Order

When you add controls to a window, the system automatically creates a default tab order for cursor movement. The initial order is based on the vertical and horizontal position of the controls. The control in the upper left-hand corner comes first, and subsequent controls are ordered based on a left-to-right, top-to-bottom progression.

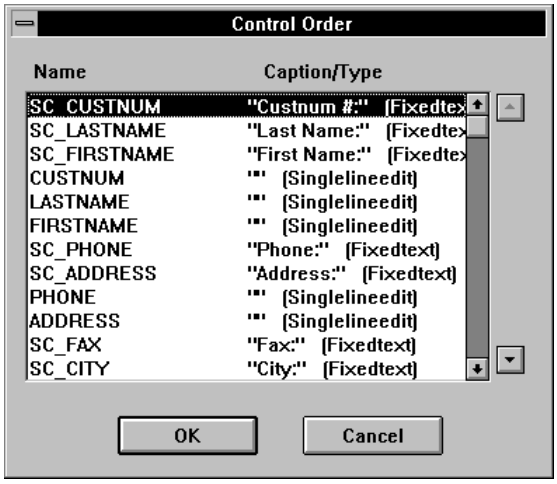
To view the tab order for a data window, choose the Edit Control Order menu command. For example, below is the default tab order of the CustOrd data window before we made any changes:



The Control Order dialog box displays the names of all the controls, as well as their captions and control types, in *tab stop* order as they appear in the source code. It also allows you to change the cursor tabbing order for a window's controls by reordering the controls in the source code.

When you customize a data window by rearranging the controls, the system automatically updates the default tab order. For example, let's view the tab order of the CustOrd data window after our changes.

Choose the Edit Control Order menu command, displaying the following:



Notice that the default tab order has been changed to reflect the new positioning of the controls.

Note: This command affects only the order in which the cursor moves from control to control within a window; it does *not* alter the controls' actual positions on the window.

The Control Order dialog box allows you to change the default tab order that is set automatically by the system when you place your controls. For more detailed information, see Changing Tab Order by Reordering Controls in the "Using the Window Editor" chapter of the *IDE User Guide*.

Moving On

You could make other modifications, such as changing the color or font used for any control in either the main data window or its nested sub-data window using the appropriate Properties window. However, our main data window is fine as it is, so let's close the Window Editor—saving the changes, of course—and look at the resultant source code.

The Source Code

Similar to the DB Server Editor, saving your changes in the Window Editor generates source code. Although you probably won't be directly modifying the generated code, it may help to take a closer look at exactly what was just generated for you, so that you get a better understanding of how all of the generated source fits together.

Window Entities

First, there are *window* entities for both the CustOrd and the CustOrd_Detail windows. The purpose of these entities is so that you can easily edit the data windows with the Window Editor.

Class Entities

There are also class entities for both windows that inherit from the DataWindow class defined in the GUI Classes library. These windows can, therefore, be edited and used individually.

Resource and Define Entities

There are resource entities for both windows, along with several define entities to number the individual edit controls.

Access/Assign and Init() Method Entities

Finally, there are access/assign methods for the field edit controls and Init() methods for both of the DataWindow subclasses.

These Init() methods are complex, defining all edit controls in the window's form view and instantiating a HyperLabel object for each one. The field edit controls also have an appropriate FieldSpec object attached.

Then, the data window is attached to the appropriate data server with the Use() method, and the browse view for the window is defined with the DataBrowser and DataColumn classes (also defined in the GUI Classes library).

Finally, the window establishes its default view with the ViewAs() method and, in the case of the CustOrd window, attaches the detail window and establishes the relationship between the windows using the SetSelectiveRelation() method mentioned earlier.

Summary

That concludes the lesson in setting up the data window. You will now move on to modify the EmptyShellMenu to add a menu command that will open this data window.