- *Tabular reports* provide a column for each field of the table or query and print the value of each field of the records in rows under the column header. If you have more columns than can fit on one page, additional pages print in sequence until all of the columns are printed; then the next group of records is printed. Figure 14.1 shows in Report Preview mode a tabular report that is based on Northwind.mdb's Customers table.



FIG. 14.1 A preview of a tabular report created from Northwind.mdb's Customers table.

- *Multicolumn reports* are created from single-column reports by using the "newspaper" or "snaking" column approach of desktop publishing and word processing applications. Information that doesn't fit in the first column flows to the top of the second column, and so on. The format of multicolumn tables wastes less paper but the uses are limited because the column alignment is unlikely to correspond with what you want.
- *Groups/totals reports* are the most common kind of report. Access groups/totals reports summarize data for groups of records and then adds grand totals at the end of the report.
- *Mailing labels* are a special kind of multicolumn report designed to print names and addresses (or other multifield data) in groups. Each group of fields constitutes a cell in a grid. The design of the stock adhesive label on which you are printing determines how many rows and columns are on a page.
- *Unbound reports* contain subreports based on unrelated data sources, such as tables or queries.

The first four types of reports use a table or query as the data source, as do forms. These kinds of reports are said to be *bound* to the data source. The main report of an unbound report is not linked to a table or query as a data source. The subreports contained by an unbound report, however, must be bound to a data source. *Unbound reports* allow you to incorporate subreports that are bound to independent tables or queries.

Oracle Ex. 1009, p. 534
Oracle v. DeBellis
IPR2024-00221

Creating a Grouping Report with the Report Wizard

This section shows you how to use the Report Wizard to create a grouping report based on data in the Products and Suppliers tables of the Northwind Traders sample database. (Like the Form Wizard, the Report Wizard allows you to create reports that contain data from more than one table without first creating a query.) This report displays the quantity of each specialty food product in inventory, grouped by product category.

To create an inventory report, you modify the basic report created by the Report Wizard. The process of creating a basic report with the Report Wizard is similar to the process that you used to create a form in Chapter 12, "Creating and Using Forms." An advantage of using the Report Wizard to introduce the topic of designing Access reports is that the steps for this process are parallel to the steps you take when you start with a default blank report. Chapter 15, "Preparing Advanced Reports," explains how to start with a blank report and create more complex reports.

To create a Products on Hand by Category report, follow these steps:



1. Click the Reports tab in the Database window, and then click the New button. Access displays the New Report dialog (see Figure 14.2).

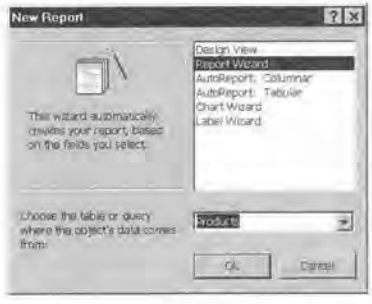


FIG. 14.2 The New Report dialog, in which you select the report's type and data source.

2. Like forms, reports require a data source, which can be a table or query. Select the Products table from the choices offered in the New Report dialog's drop-down list (refer to Figure 14.2). Select Report Wizard in the list in the dialog's upper-right corner and click OK. The Report Wizard displays its opening dialog.

3. The fields that you choose to display represent rows of the report. You want the report to print the product name and supplier so that users do not have to refer to another report to associate codes with names. The fields from the Products table that you need for this report are CategoryID, ProductID, ProductName, SupplierID, and UnitsInStock. With the > button, select these fields in sequence from the Available Fields list (see Figure 14.3). As you add fields to the Selected Fields list, Access removes the field names from the Available Fields list. Alternatively, you can double-click the field name in the Available Fields list to move the field name to

Oracle Ex. 1009, p. 535
Oracle v. DeBellis
IPR2024-00221

the Selected Fields list. The fields appear from left to right in the report based on the top-to-bottom sequence in which the fields appear in the Selected Fields list.

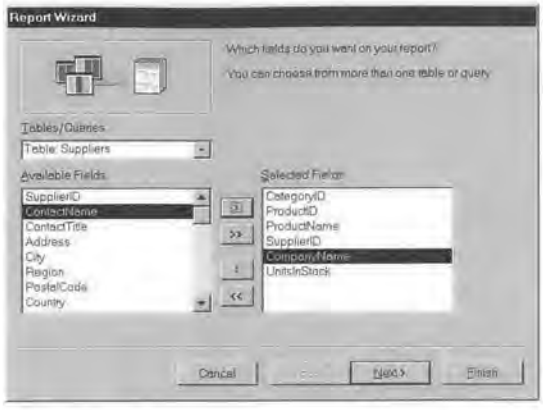


FIG. 14.3 Selecting the fields of a report from one or more tables or queries in the Report Wizard's opening dialog.

4. To avoid having to look up Supplier ID numbers in a separate report, you need to add to this report the CompanyName field from the Suppliers table. Open the Tables/Queries drop-down list and select Table: Suppliers (refer to Figure 14.3).

5. Instead of presenting the supplier name as the report's last field, you want the report's CompanyName column to follow the SupplierID report column. Select the SupplierID field in the Selected Fields list. Now select the CompanyName field from the Available Fields list and click the > button. Access moves the CompanyName field from the Available Fields list and inserts the field into the Selected Fields list, after the SupplierID field and before the UnitsInStock field (refer to Figure 14.3). Choose Next to continue with the second Wizard dialog, shown in Figure 14.4.

> **Note**
>
> If you want to change the field order shown in the right pane of Figure 14.4, use the < button to move the field back to the Available Fields list. You can retrace your steps to correct an error by clicking the Back button whenever it is activated. The Finish button accepts all defaults and jumps to the end of the wizard, so you shouldn't use this button until you're familiar with the Report Wizard's default selections.

6. The Report Wizard asks you to choose how you want to view the data in the report. Notice the Show Me More Information button near the left center of the wizard dialog. Click this button to display the first of a series of hint dialogs for the Report Wizard (see Figure 14.5). If you click the Show Me Examples option, Access displays additional hint screens. These screens use examples from the Sales Reps, Customers, and Orders tables to show you the different groupings that the Report Wizard can automatically add to the report. Click the Close button repeatedly until you return to the Report Wizard dialog shown in Figure 14.4.

Oracle Ex. 1009, p. 536
Oracle v. DeBellis
IPR2024-00221

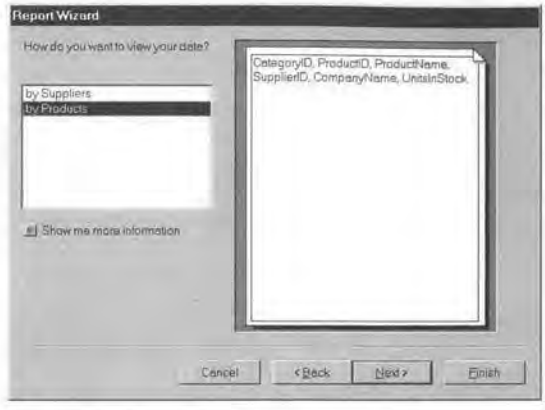


FIG. 14.4 Choosing how you want to view your data in the second Report Wizard dialog.



FIG. 14.5 The first of the Report Wizard's hint screens.

7. For this report, you select your own groupings. Select By Products in the list, and choose Next to continue with the third Report Wizard dialog.

8. The Report Wizard asks whether you want to add any grouping levels to the report. Select the CategoryID field in the list, and click the > button to establish the grouping By Products category. The Report Wizard dialog now appears as shown in Figure 14.6.

9. Click the Grouping Options button. The Report Wizard displays the Grouping Intervals dialog shown in Figure 14.7. By changing the grouping interval, you can affect how Access groups data in the report. For numeric fields, you can group items by 10s, 50s, 100s, and so on. For text fields, you can group items based on the first letter, the first three letters, and so on.

Oracle Ex. 1009, p. 537
Oracle v. DeBellis
IPR2024-00221

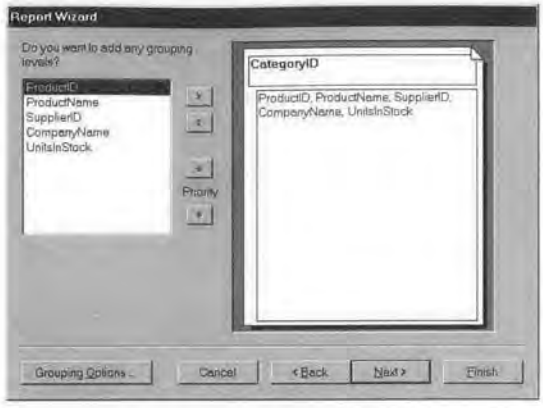


FIG. 14.6 Selecting grouping levels for your report.

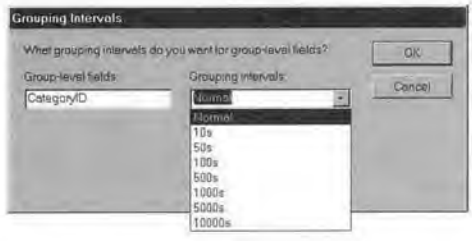


FIG. 14.7 Selecting grouping intervals for your report.

> **Note**
>
> If your application uses a text-coding scheme, such as BEVA for alcoholic beverages and BEVN for nonalcoholic beverages, you can combine all beverages in a single group by selecting 1st 3 Characters from the Grouping Intervals list. Access 97 provides this option for numeric fields and for fields of the Text data type.

10. This report doesn't require any special grouping interval, so select Normal in the Grouping Intervals list, and click OK to return to the Report Wizard's third dialog (refer to Figure 14.6). Choose Next to continue with the fourth wizard dialog.

11. You can sort the records within groups by any field that you choose (see Figure 14.8), with up to four different sorted fields. The dialog does not offer CategoryID as a choice because the records are grouped in this field, and the field on which the grouping is based is therefore automatically sorted. Select ProductID in the first drop-down list. By default, the sort order is ascending; if you want a descending sort order, click the button to the right of the drop-down list. (This button is a toggle control; click it again to return to an ascending sort.)

Oracle Ex. 1009, p. 538
Oracle v. DeBellis
IPR2024-00221

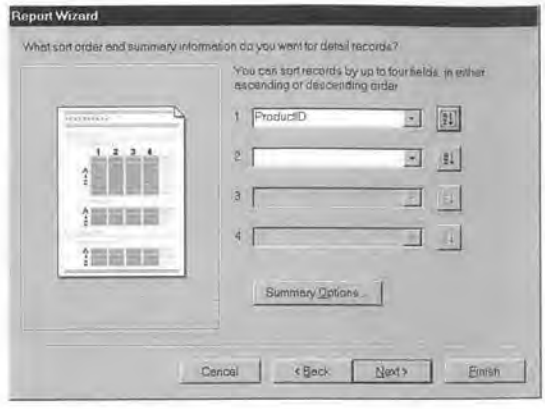


FIG. 14.8 Selecting a sort order for fields within groups.

12. Choose the Summary Options button to display the Summary Options dialog (see Figure 14.9). If you want to add summary information to a report column, set the options for that column in this dialog. The Report Wizard lists all of the numeric fields on the report that aren't AutoNumber fields and offers you check boxes to select a Sum, Average, Minimum, and Maximum for that report column. Depending on the check boxes that you select, the Report Wizard adds those summary fields to the end of the report.

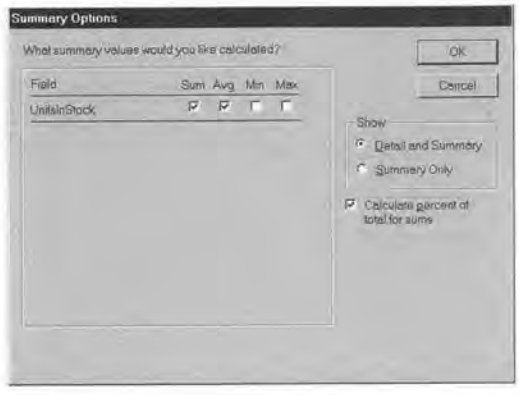


FIG. 14.9 Choosing the summary data you want included in your report in the Summary Options dialog of the Report Wizard.

The Show option group allows you to select whether the report shows the summary fields only or the full report with the summary fields added at the end of each group and at the end of the report. For this report, select the Sum and Avg check boxes, the Detail and Summary option, and the Calculate Percent of Total for Sums check box. (The Calculate Percent of Total for Sums check box displays the group's total as a percentage of the grand total for all groups.) Click OK to return to the Report Wizard dialog.

Oracle Ex. 1009, p. 539
Oracle v. DeBellis
IPR2024-00221

13. Choose Next to continue with the fifth Wizard dialog, shown in Figure 14.10. The Report Wizard asks you to select one of six layout styles for your report. The window in the left side of the wizard dialog shows a preview of the layout style that you select. For this report, choose Stepped in the Layout option group.

14. By default, the Report Wizard selects the Adjust the Field Width So All Fields Fit on a Page check box. As a rule, you should select this option to save paper and make your report more legible. In the Orientation option group, you select the report's printing orientation. Make sure that you select the Portrait option. Choose Next to continue with the sixth Report Wizard dialog.

15. Select one of the predefined report styles for your report. The window on the left shows a preview of the selected style (see Figure 14.11). (You can customize or create your own styles for the Report Wizard to use. This activity is described in the "Using AutoFormat and Customizing Report Styles" section later in this chapter.) Select the Compact style, and then choose Next to continue with the seventh and final Report Wizard dialog.

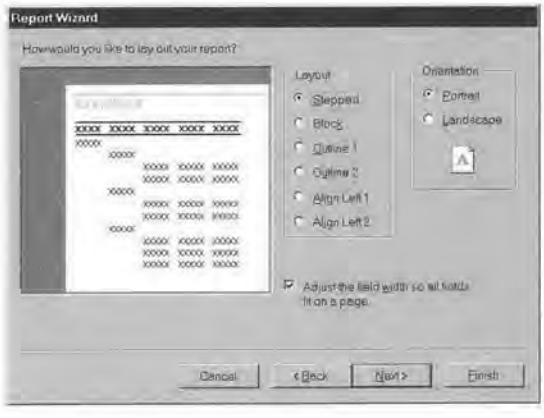


FIG. 14.10 Choosing a layout format for a report.

16. Type **Products On Hand by Category** as the title for the new report; the Report Wizard also uses this title as the name of the saved report it creates (see Figure 14.12). Select the Preview the Report option and click Finish to complete your report specification. The Report Wizard creates the report and displays it in print preview mode. (To get Help with the report, click the Display Help on Working with the Report? check box.)

    Figure 14.13 shows the basic report that the Report Wizard creates. Use the vertical and horizontal scroll bars to position the preview as shown. (Leave the report in Print Preview mode for now because you use this report as the basis for examples in subsequent sections of this chapter.)

Oracle Ex. 1009, p. 540
Oracle v. DeBellis
IPR2024-00221



FIG. 14.11 Selecting a style for your report.



FIG. 14.12 Giving your report a name and title, and choosing how you want to view the completed report.

◀◀ See "Using Lookup Fields in Tables," p. 332

In Figure 14.13, notice that the report appears to have duplicate columns—a Supplier column, which lists the name of the product's supplier, and the Company Name column, which also lists the name of the product's supplier. The reason for this duplication is twofold:

- A Caption property is set for the SupplierID field of the Products table. This property establishes an alias for the field. The Report Wizard substitutes the text in the Caption property for the field name in field labels on the report.
- The SupplierID field of the Products table is defined as a lookup field. Therefore, Access automatically looks up the value that corresponds to the SupplierID code in the Suppliers table and displays that value, rather than the actual code number stored in the field.

Oracle Ex. 1009, p. 541
Oracle v. DeBellis
IPR2024-00221

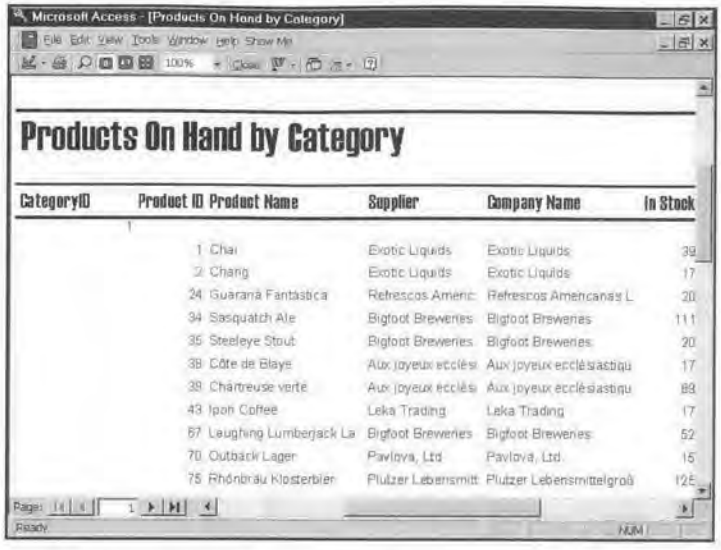


FIG. 14.13 The basic report created by the Report Wizard, shown in a maximized window.

When you view the lookup field in Design view, a nonfunctional drop-down list button (see Figure 14.14 in the next section, "Using Access's Report Windows") designates lookup fields. This report doesn't need to include the CompanyName field of the Suppliers table because the SupplierID field is a lookup field.

With a few simple modifications, you can obtain a finished report with the information necessary to analyze Northwind's current inventory. (See "Modifying a Basic Report Wizard Report" later in this chapter.)

Using Access's Report Windows

The windows that you use to design and run Access reports are easier to use than those windows that you use for other basic Access functions. To open an existing Access report, click the Report tab in the Database window and then select a report name from the Database window. If you click the Design button or the New button to create a new report, the Design Mode toolbar appears with the buttons listed in Table 14.1.

Table 14.1 Standard Toolbar Buttons in Report Design Mode

Button	Function	Menu Choice
	Selects Print Preview to display how your report appears when printed. You can print the form from the Print Preview window. (Same as the Print Preview button.)	File, Print Preview
	Saves the current report.	File, Save

(continues)

Oracle Ex. 1009, p. 542
Oracle v. DeBellis
IPR2024-00221

Table 14.1 Continued

Button	Function	Menu Choice
	Prints the report without displaying the Print dialog. Access prints the report using the current printer settings.	Not applicable
	Selects Print Preview to display how your report appears when printed. You can print the form from the Print Preview window.	File, Print Preview
	Starts the spelling checker to check the selected label control's spelling.	Tools, Spelling
	Cuts the selected object(s) from the report and puts it on the Clipboard.	Edit, Cut
	Copies the selected object(s) from the report onto the Clipboard.	Edit, Copy
	Pastes the Clipboard's contents onto the report.	Edit, Paste
	Copies formatting from the selected object to another object of a similar type.	Not applicable
	Undoes the last change that you made to the report.	Edit, Undo
	Inserts a new Hyperlink control or allows you to edit an existing Hyperlink control.	Insert, Hyperlink
	Toggles the display of the Web toolbar.	not applicable
	Displays a list of fields in the query or table that is the main report's data source.	View, Field List
	Displays or closes the toolbox.	View, Toolbox
	Displays the Sorting and Grouping dialog in which you can establish the structure of reports and the order in which the report presents the data.	View, Sorting and Grouping
	Applies your choice of several predefined report formats, including formatting for the text fonts and color settings.	Format, AutoFormat
	Opens the window in which you can edit event-handling code.	View, Code
	Displays the Properties dialog for the entire report, the sections of the report when you click the section divider bars, or the properties of a control when a control is selected.	View, Properties
	Displays a Build Wizard for the selected object or property in the report. This button is enabled only if Access has a builder for the selected item.	Not applicable
	Displays the Database window.	Window, 1 Database

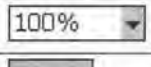
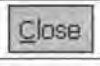
Oracle Ex. 1009, p. 543
Oracle v. DeBellis
IPR2024-00221

Button	Function	Menu Choice
	Creates a new object. Click the arrow to the right of this button to see a drop-down list of objects that you can create.	Not applicable
	Displays the Microsoft Office Assistant that, in turn, displays help text related to the actions you are performing.	Help, Microsoft Access Help

Many of the buttons listed in Table 14.1 serve the same purposes for both forms and reports. As is the case for Form Design mode, the Formatting toolbar's buttons for formatting text are enabled only when a control object that can contain text is selected. The Formatting toolbar for reports is identical to the Formatting toolbar for forms. The Formatting toolbar was described in Chapter 12, "Creating and Using Forms."

If you double-click the name of an existing report, or click the Preview button in the Database window, the report displays in Print Preview mode, which is the Run mode for reports. Table 14.2 lists the toolbar's buttons in Print Preview mode.

Table 14.2 Standard Toolbar Buttons in Report Print Preview Mode

Button	Function	Menu Choice
	Prints the report without displaying the Print dialog. The report is printed using the current printer settings.	Not applicable
	Toggles between full-page and full-size views of the report. Clicking the mouse when its pointer appears as the magnifying glass symbol produces the same effect.	View, Zoom
	Displays one full page.	View, Pages, 1
	Displays two full pages.	View, Pages, 2
	Displays a palette from which you can select several multiple-page views of the report.	
	Changes the size of the view from 200 percent to 10 percent or fits the report to the window.	View, Zoom
	Closes Print Preview and returns to Report Design view or to the Database window.	File, Print Preview
	Displays a drop-down list of shortcut commands for Microsoft Office Links: Merge It, Publish It with MS Word, and Analyze It with MS Excel.	Tools, OfficeLinks
	Opens the Database window.	Window, 1 Database
	Creates a new object. Click the arrow at the right of this button to display a drop-down list of objects that you can create.	Not applicable
	Displays the Microsoft Office Assistant that, in turn, displays help text related to the actions you are performing.	Help, Microsoft Access Help

Oracle Ex. 1009, p. 544
Oracle v. DeBellis
IPR2024-00221

Chapter 22, "Using Access with Microsoft Word and Mail Merge," discusses using the Office Links button to print reports as files in rich-text format. Chapter 21, "Using Access with Microsoft Excel," discusses printing files in Excel BIFF format. Chapter 15, "Preparing Advanced Reports," describes linking files created from reports to Microsoft Mail messages.

Using AutoFormat and Customizing Report Styles

The AutoFormat toolbar button works the same way for reports as for forms. Chapter 12, "Creating and Using Forms," contains a detailed, step-by-step explanation of how to use Access 97's AutoFormat button and customize the predefined AutoFormat styles or create your own AutoFormat styles. Follow the instructions in the "Using AutoFormat" section of Chapter 12 to apply an AutoFormat style to a report or define or customize a report AutoFormat style. Access stores styles for reports and forms separately, so you'll need to create separate AutoFormat styles for your reports.



◀◀ See "Using AutoFormat," p. 411

As with forms, to create an AutoFormat style for customized reports, you must first create a report that contains controls that are formatted the way you want for your new style. You click the AutoFormat button on the toolbar and then click the Customize button in the AutoFormat dialog to customize the format.

Modifying a Basic Report Wizard Report

The Report Wizard tries to create the optimum final report in the first pass. Usually, the wizard comes close enough to a finished product that you spend far less time modifying a wizard-created basic report than creating a report from the default blank template.

In the following sections, you use Access's report design features to make the report more attractive and easier to read.

Deleting, Relocating, and Editing Existing Controls

The first step in modifying the wizard's report is to modify the existing controls on the report. You don't need to align the labels and text boxes precisely during the initial modification; the section "Aligning Controls Horizontally and Vertically" later in this chapter covers control alignment. To modify the wizard's report in order to create space for adding additional controls, follow these steps:



1. Click the Close button of the Print Preview toolbar to enter Report Design mode; click the Maximize Window button to maximize the design window, if necessary. The Products on Hand by Category report, as created by the Report Wizard, appears as shown in Figure 14.14.

Oracle Ex. 1009, p. 545
Oracle v. DeBellis
IPR2024-00221

2. The SupplierID and CompanyName fields are redundant in this report because the SupplierID field is a lookup field. Select the Company Name label in the Page Header section, then hold down the Shift key, and click the CompanyName field in the Detail section. Press Delete to remove the field and label from the report. (Don't worry about aligning the fields and labels yet.)

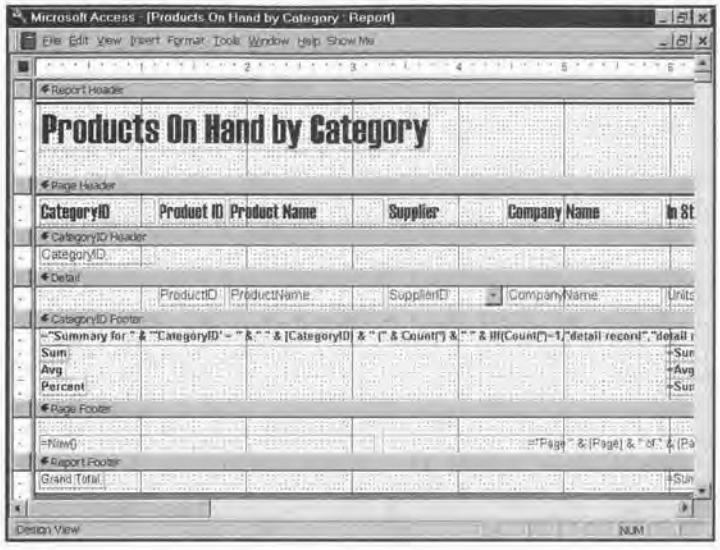


FIG. 14.14 The basic report in Design mode (all toolbars have been hidden to make the entire report visible).

3. This report is more useful if you include the dollar value of both the inventory and number of units on hand. To accommodate one or two additional columns, you must compress the fields' widths. CategoryID occupies a column, but you can display this column's content in the CategoryID footer (or header) without using the extra column space. Select and delete the CategoryID label from the Page Header section, and select and delete the CategoryID text box from the CategoryID Header section. For this report, you'll put the CategoryID name in the footer section of the group, so drag the Detail section bar upward to eliminate the space occupied by the CategoryID Header. Your report appears as shown in Figure 14.15.

4. All of the Page Header labels, Detail text boxes, and Totals text boxes in the CategoryID Footer and Report Footer sections must move to the left as a group. Click the Product ID label to select it, and then press and hold down Shift. Click the remaining Page Header labels, each of the Detail text boxes, the three summary field text boxes in the CategoryID Footer section, and the Grand Total text box in the Report Footer section. Then release Shift. (To select all of the labels and text boxes, you'll need to scroll the report to the right and left and up and down.)

Oracle Ex. 1009, p. 546
Oracle v. DeBellis
IPR2024-00221

5. Position the mouse pointer over the Product ID label at a location where the pointer turns into the graphic of a palm of a hand. Hold down the left mouse button and drag the selected fields to the left margin. Your report appears as shown in Figure 14.16.

6. You can more easily edit and position the labels if you left-justify them. Click a blank area of the report to deselect the group, select all of the Page Header labels, and click the Align Left button on the toolbar. Do the same for the Grand Total label in the Report Footer section.

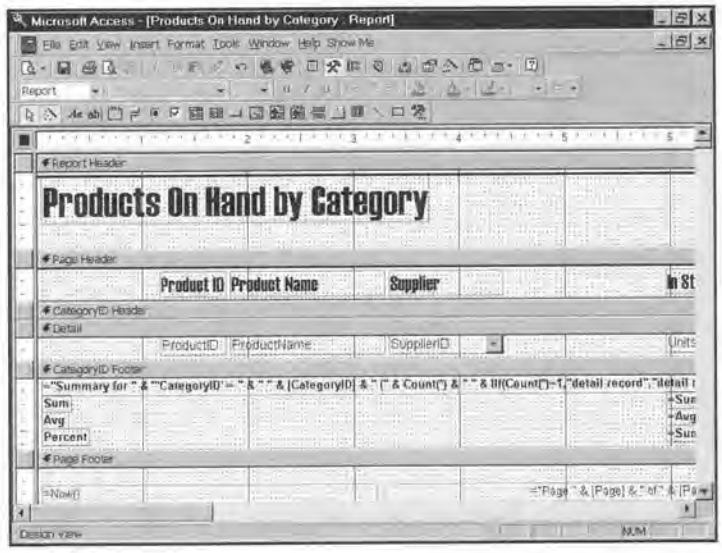


FIG. 14.15 The basic report after deleting the CategoryID label and text box and closing the space for the CategoryID Header.

7. Edit the Product ID label to read **Product**, and edit the Units In Stock label to read only **Units**. Select all of the labels in the Page Header and choose Format, Size, To Fit. Resize the widths of the ProductID, SupplierID, and UnitsInStock text boxes in the Detail section to match the width of the labels in the Page Header. Relocate the labels to provide more space on the right side of the report, as shown in Figure 14.17.

8. By default, the Report Wizard adds to the CategoryID Footer a calculated field (visible in Figure 14.17) that displays the group's field name (CategoryID) and value to help identify the group footer's summary fields. For example, for CategoryID 1, the calculated field displays the following:

```
"Summary for 'CategoryID' = 1 (12 detail records)."
```

For this report, you want a more explicit description of the product category—more than just the CategoryID number. Delete this calculated field; you'll replace it in the next step.

Oracle Ex. 1009, p. 547
Oracle v. DeBellis
IPR2024-00221

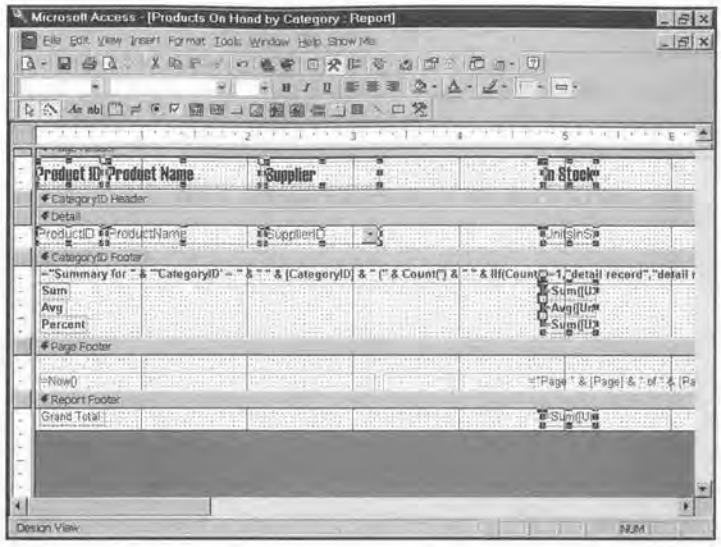


FIG. 14.16 Moving selected labels and text boxes to the report's left margin.

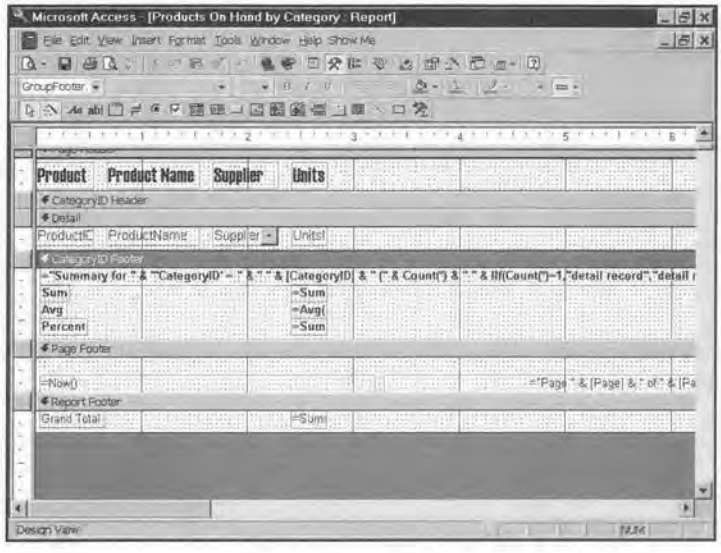
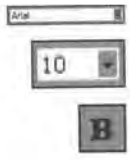


FIG. 14.17 The Products on Hand by Category report after you edit, resize, and relocate existing controls.

9. You now need to add a bound text box to identify the subtotal in the CategoryID Footer section. Click the Field List button on the toolbar. Select CategoryID from the list in the Field List window.

Oracle Ex. 1009, p. 548
Oracle v. DeBellis
IPR2024-00221

10. Click and drag the field symbol mouse pointer to the left margin of the CategoryID Footer. Because the CategoryID field is a lookup field, it displays with a drop-down list button for the field box. When printed or displayed in Print Preview, this field shows the CategoryID name rather than the numeric code. Click the Field List button on the toolbar to close the Field List window.



11. Select the label of the CategoryID field that you just placed, and use the Font and Size drop-down lists on the Formatting toolbar to set the label's font to Arial and the label's size to 8 points. Next, select the CategoryID text box, click the Bold button on the toolbar to add the bold attribute to the CategoryID text box, and also select the Arial font at a size of 8 points. Figure 14.18 shows the new bound CategoryID field in place of the calculated field that you deleted in step 8. (In the figure, the report has been scrolled upward in the window to make the Report Footer and Page Footer sections visible.)

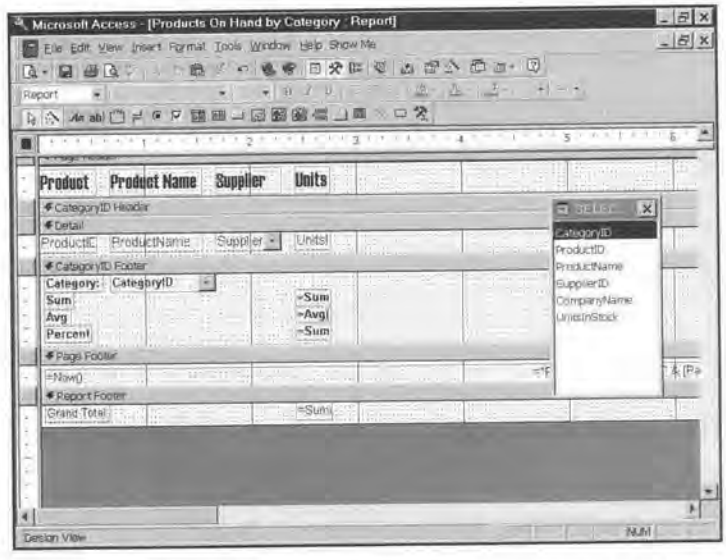


FIG. 14.18 The Products On Hand by Category report, after adding the CategoryID field to the CategoryID Footer section.

12. Drag the two calculated fields in the Page Footer section until they are one grid mark away from the top of the Page Footer section. Drag the Report Footer bar upward to reduce the Page Footer's height, as shown in Figure 14.18.

13. For this report, the Average field is unnecessary; delete it and its label, and then rearrange the remaining fields and labels. Click and drag the `=Sum([UnitsInStock])/[UnitsInStock Grand Total Sum]` text box from its present location below the `=Sum([UnitsInStock])` text box to a position at the top of the CategoryID Footer, near the page's right edge. Drag the `=Sum([UnitsInStock])` field to a position at the

Oracle Ex. 1009, p. 549
Oracle v. DeBellis
IPR2024-00221

top of the CategoryID text box and near the center of the page. Finally, move up the Page Footer divider bar to reduce the footer's depth (see Figure 14.19).

14. Click the toolbar's Save button to save your report.

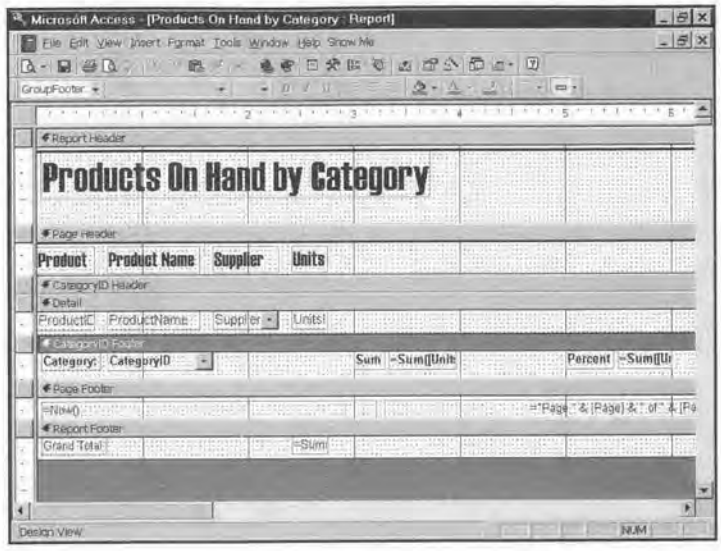


FIG. 14.19 The completely modified Products On Hand by Category report in Design mode.

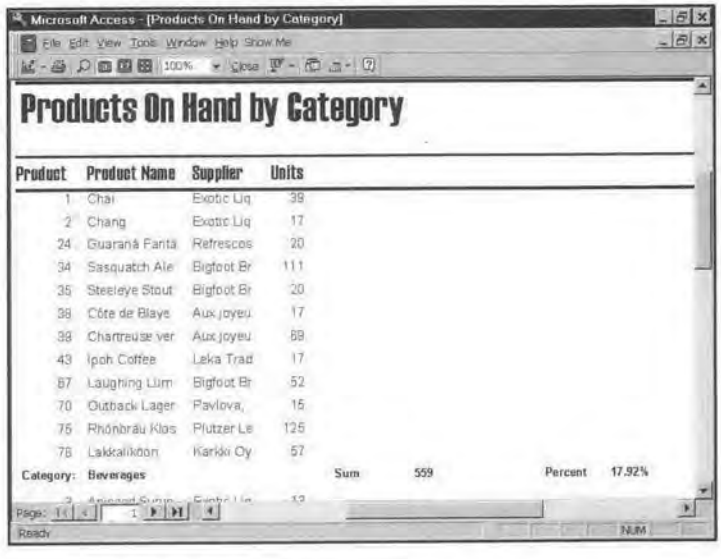


FIG. 14.20 Previewing the Products on Hand by Category report.

Oracle Ex. 1009, p. 550
Oracle v. DeBellis
IPR2024-00221

Troubleshooting

When I preview or print my report, Access displays or prints a blank page after each page with data.

If a report's width becomes greater than the net printable width (the paper width minus the sum of the left and right margins), the number of report pages doubles. Columns of fields that do not fit a page's width print on a second page, similar to the printing method used by spreadsheet applications. If you set your right margin beyond the right printing margin, or if the right edge of any control on the report extends past the right printing margin, the added pages often are blank. Change the printing margins or reduce the width of your report so that it conforms to the printable page width. (See the section "Adjusting Margins and Printing Conventional Reports" later in this chapter.)



To check the progress of your work, periodically click the toolbar's Print Preview button to display the report prior to printing. At this point, your Products On Hand by Category report appears in Print Preview mode, as shown in Figure 14.20.

Using the `DLookUp()` Domain Aggregate Function to Print Product Category Names

As you saw in the preceding section of this chapter, lookup fields in a table are placed on a report as a limited-function combo box. You can't use the combo box to select values, but it causes a value looked up from another table to be displayed instead of the actual field value. For example, if you place the CategoryID field on the report (as you did in the preceding section), Access displays the category name in the report rather than the actual CategoryID number, because the CategoryID field in the Products table has lookup field properties assigned to it. When you place the CategoryID field on the report, Access automatically creates a combo box control with the properties needed to look up the CategoryName field from the Categories table.

Not every table that you use in your reports will have lookup fields, however, nor is it necessarily desirable to create lookup fields for all numeric code fields (such as CategoryID and SupplierID). If you want to display a looked-up value for a field that isn't defined as a lookup field, you use Access's domain aggregate function, `DLookUp()`, to find values from another table that correspond to a value in one of the report's fields. For example, if you want to display both the actual CategoryID number and the CategoryName in the Group Footer of the Products On Hand by Category report, you can use the `DLookUp()` function to display the text of the CategoryName field from the Categories table, and a bound text field to display the CategoryID number from the Products table.

To change the CategoryID combo box control to a standard text box and add a new Category Name field that uses the `DLookUp()` function to the CategoryID Footer section of the Products On Hand by Category report, follow these steps:



1. In Report Design mode, select the CategoryID combo box, and then choose Format, Change To, Text Box to convert the combo box to a regular text box field. This field will no longer display the looked-up CategoryName field but will display the actual number stored in the CategoryID field.

Oracle Ex. 1009, p. 551
Oracle v. DeBellis
IPR2024-00221

2. Edit the CategoryID field label to read `ID:`, then resize the CategoryID text box so that it is approximately 3/8 of an inch wide. Move both the text and field boxes closer together, near the left edge of the report's CategoryID Footer section (see Figure 14.21).
3. Click the Toolbox button to display the toolbox if it is not already displayed.
4. Click the Text Box tool and add a new unbound text box to the right of the CategoryID text box with the approximate dimensions shown in Figure 14.21.
5. Delete the field label for the new text box control, and give the text box itself the bold text attribute.
6. Click the new text box, and type the following as the text box's value:

   =DLookUp("[CategoryName]","Categories","[CategoryID] = Report!CategoryID") & " Category"

   `[CategoryName]` is the value that you want to return to the text box. `Categories` is the table that contains the CategoryName field. `[CategoryID] = Report!CategoryID` is the criterion that selects the record in the Categories table with a CategoryID value that is equal to the value in your report's CategoryID text box. The `Report` prefix is necessary to distinguish between the CategoryID field of the Categories table and a control object of the same name. (`Report` is necessary in this example because Access has automatically named the report's CategoryID text box control as CategoryID.)

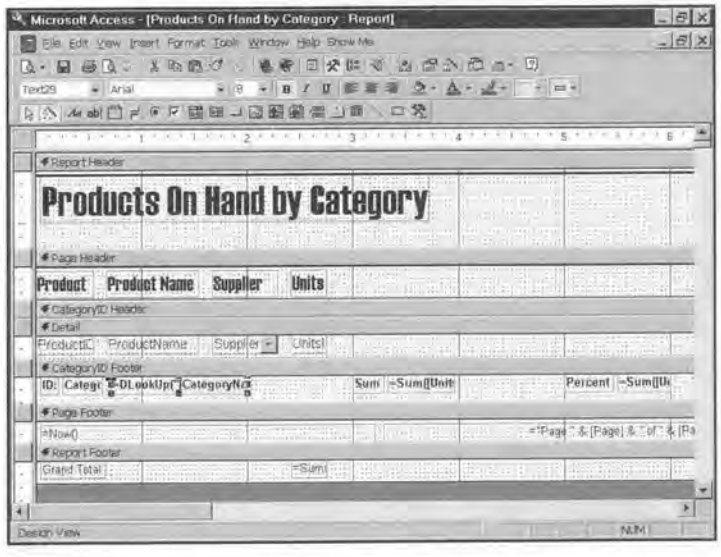


FIG. 14.21 The transformed and resized CategoryID field with the new, unbound text box containing the `DLookUp()` function.

Oracle Ex. 1009, p. 552
Oracle v. DeBellis
IPR2024-00221

7. Select the CategoryID text box and verify that the name CategoryID appears in the Object Name text box (at the left side of the Formatting toolbar). If not, click the Properties button of the toolbar, select the Other tab in the Properties window, and type **CategoryID** as the value of the control's Name property (see Figure 14.22).



8. Click the Report View button of the toolbar. Your Products On Hand by Category report appears as shown in Figure 14.23.

Troubleshooting

When I preview or print my report, the text box that contains the `DLookUp()` *function displays* `#error` *or just the word* `Category`.

Your DLookUp expression contains a typographical error, or one of the objects that you specified does not exist. Make sure that you have typed the entry in the CategoryName text box exactly as shown in the preceding step 6. If the field name in the table or query for which you are searching is the same as the control name, make sure that you add the `Report!` prefix to the control name. For example, you must add **Report!** if you assign Category Name as the name of the new control that you added in the preceding example.



FIG. 14.22 Verifying the name of the CategoryID text box.

Adding Other Calculated Controls to the Report

Calculated controls, such as the `DLookUp()` control that you added in the preceding section, are quite useful in reports. You use calculated controls to determine extended values, such as quantity times unit price or quantity times cost. Now you have enough space at the right of the report to add a column for the UnitPrice field and a column for the extended inventory value, which is UnitPrice multiplied by UnitsInStock. To add these controls, follow the steps described in the following subsections.

Changing the Report's Record Source. You created the Products On Hand by Category report by selecting fields directly from the Products and Suppliers table in the Report Wizard. As a result, the Record Source property for the report, as a whole, is a SQL statement that selects only those fields that you chose initially in the Report Wizard. Although it's possible to add fields to the report by creating unbound text box controls

Oracle Ex. 1009, p. 553
Oracle v. DeBellis
IPR2024-00221

and using the Expression Builder to create an expression to retrieve the desired value, it's much easier to create a query to select the fields desired for the report and substitute the new query as the report's data source.

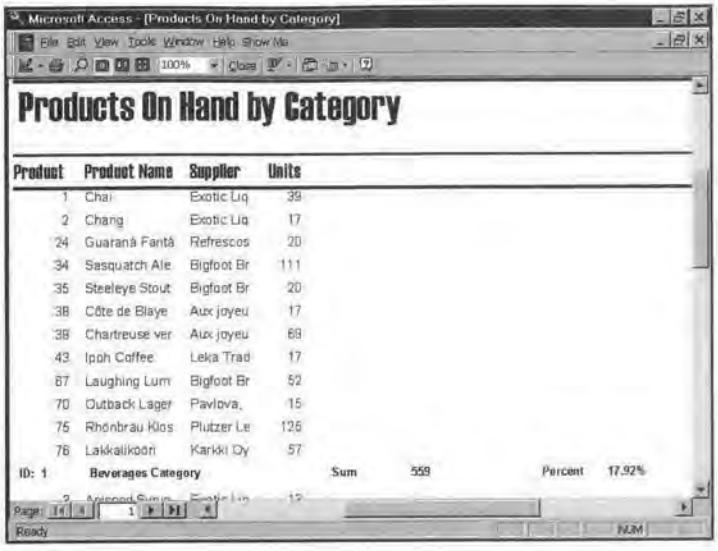
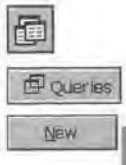


FIG. 14.23 The Products On Hand by Category report with the `DLookUp()` field added.

To create a query for use with the Products on Hand by Category report, follow these steps:



1. Click the toolbar's Database Window button to display the Database window. Click the Queries tab and then click New to create a new query.
2. In the New Query dialog, select Design View and then click OK.
3. Double-click the Products table in the Show Table dialog to add the table to the query. Then click Close to dismiss the Show Table dialog.
4. Drag the * field to the first column of the query grid to add to the query all of the Products table's fields.
5. Drag the Discontinued field to the query grid's second column.
6. Clear the Show check box for the Discontinued field, and then type **=False** in the Discontinued field's first Criteria row (see Figure 14.24).
7. Click the Close Window button to close the Query window. A prompt asks whether you want to save changes to the query design. Click Yes.
8. In the Save As dialog's Query Name text box, type **qryProductOnHand**. Then click OK to save the query.

Oracle Ex. 1009, p. 554
Oracle v. DeBellis
IPR2024-00221

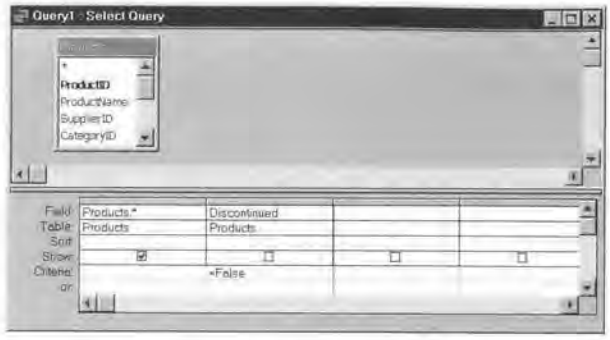


FIG. 14.24 The qryProductOnHand query that contains all fields from the Products table and excludes discontinued products.

The query that you have just created contains all of the fields from the Products table and excludes discontinued products from the record set. (In other words, the query includes only those records whose Discontinued field contains the `False` or `No` value.)

To change the report's Record Source property, follow these steps:



1. Open the Products on Hand by Category report in Design mode.
2. Choose Edit, Select Report.



3. Click the toolbar's Properties button to open the report's Properties window. Then click the Data tab to display the report's various data properties.
4. Click the Record Source text box, and then use the drop-down list to select the `qryProductOnHand` query as the report's new Record Source property.



5. Click the toolbar's Save button to save the changes to the report.

Adding the Calculated Fields. Now that you've changed the report's record source, you have easy access to the UnitPrice field that you need for adding the additional calculated fields to the report. To add the UnitPrice field and the Value calculated field to the report, follow these steps:



1. Display the Products on Hand by Category report in Design mode, if necessary. Then click the toolbar's Toolbox button to display the Access toolbox if it isn't already displayed.



2. Click the Label tool in the toolbox and place the label to the right of the Units label in the Page Header section. Type **Price** as the label.
3. Add another label to the right of Price and type **Value**.



4. Click the toolbar's Field List button to display the Field List window. Select UnitPrice and drag the field symbol to a position under the Price label in the Detail section. Drop the text box, and then delete the UnitPrice field's label in the report's Detail section.

Oracle Ex. 1009, p. 555
Oracle v. DeBellis
IPR2024-00221

5. To create the calculated Value text box, click the Text Box button in the toolbox and add the text box to the right of the Unit Price text box.

Tip

Entering expressions is easier if you display the Properties window and enter the expression as the Control Source property. Press Shift+F2 to open the Zoom box so you can see the entire expression as you enter it.

6. Type **=[UnitsInStock]*[UnitPrice]** as the expression for the Value text box. Delete the field label for this text box in the report's Detail section.

7. Drag the Page Footer section bar downward to increase the height of the report's CategoryID Footer section, and rearrange the text labels and fields as shown in Figure 14.25. Edit the Sum label to read **Total:** and the Percentage label to read **Percentage:**.

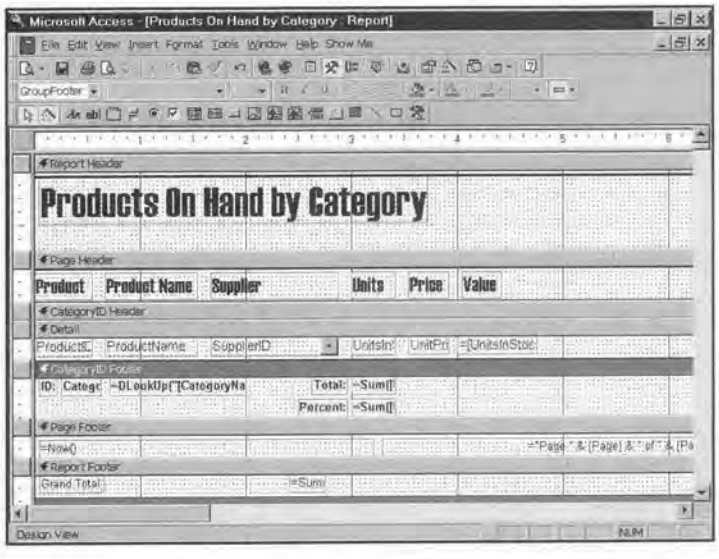


FIG. 14.25 Rearranging the CategoryID Footer section to make room for additional fields.

8. Repeat steps 5 and 6 to create the Value Subtotal text box in the CategoryID Footer section, but type **=Sum([UnitsInStock]*[UnitPrice])** as the subtotal expression. Click the toolbar's Bold button to set the Font Weight property to Bold. In the Properties window, click the Other tab and then set this text box's Name property as **txtTotalValue**.

Oracle Ex. 1009, p. 556
Oracle v. DeBellis
IPR2024-00221

9. Repeat step 8 to create the Value grand total box in the Report Footer section. In the Other page of the Properties window, set this text box's Name property as **txtGrandTotalValue**.

10. Add another unbound text box underneath the `txtTotalValue` text box. Type **=[txtTotalValue]/[txtGrandTotalValue]** as the value of the Control Source property and set the Format property's value to Percent. The report design appears as shown in Figure 14.26.

Tip

If the Parameter dialog appears, you misspelled one or more field names in the expressions. Click Cancel and check the properties that you added in steps 6 through 10.

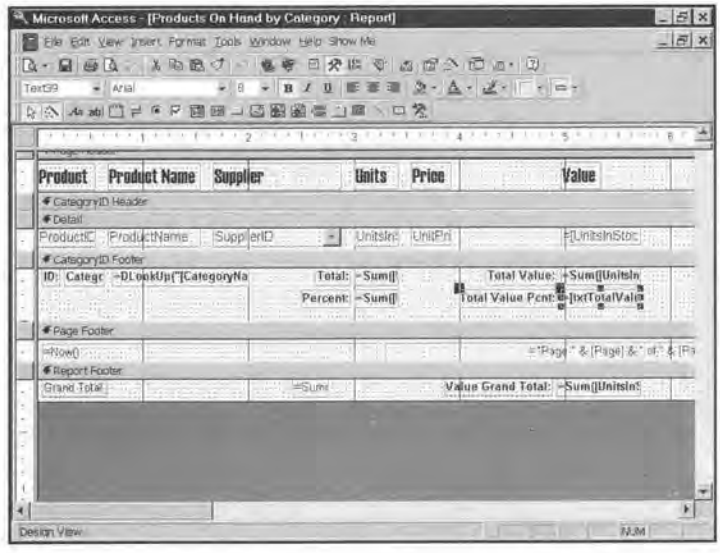


FIG. 14.26 Adding the Price, Value, Total Value, Total Value Pcnt, and Grand Total Value fields to the report.



11. Click the toolbar's Report View button to check the result of your additions. The report appears as in Figure 14.27. Use the vertical scroll bar, if necessary, to display the category subtotal. The next section describes how you can correct any misaligned values and the spacing of the Detail section's rows.

12. Click the Bottom of Report page selector button to display the grand totals for the report (see Figure 14.28). The record selector buttons become page selector buttons when you display reports in Run mode.

Oracle Ex. 1009, p. 557
Oracle v. DeBellis
IPR2024-00221

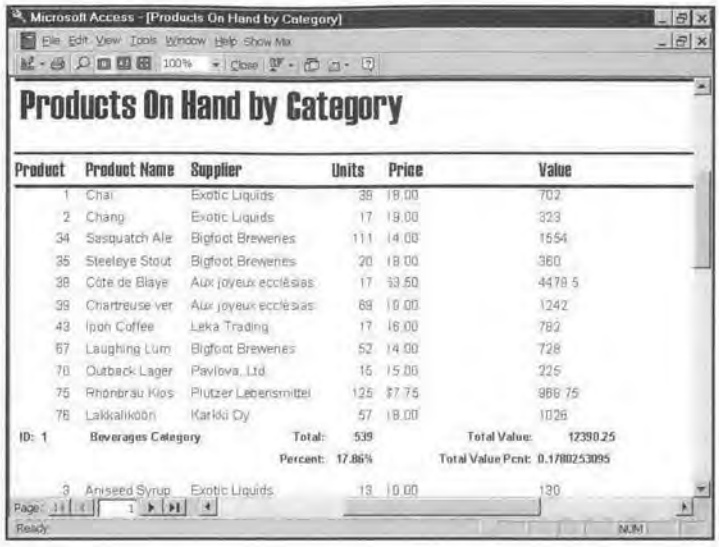


FIG. 14.27 Page 1 of the report, with calculated product values and value subtotals.

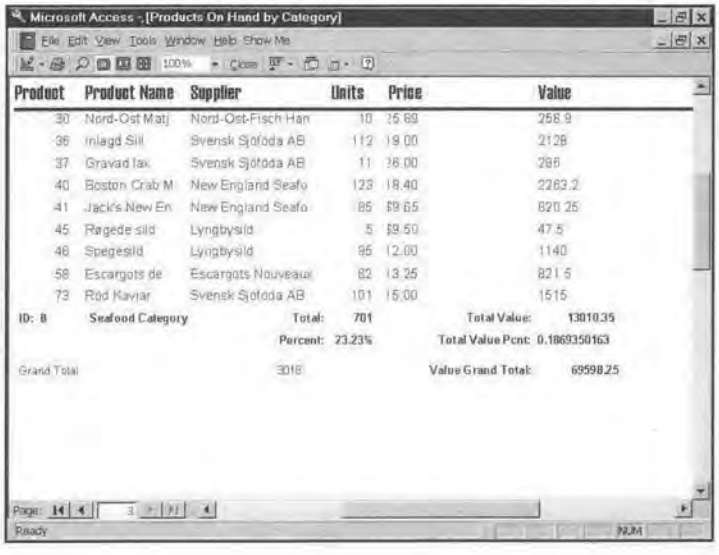


FIG. 14.28 The last page of the report, with grand totals for Units and Value.

Aligning and Formatting Controls, and Adjusting Line Spacing

On reports, the exact alignment of label and text box controls is more important than alignment on forms because in the printed report, any misalignment is obvious. Formatting the controls further improves the report's appearance and readability.

Oracle Ex. 1009, p. 558
Oracle v. DeBellis
IPR2024-00221

The spacing of the report's rows in the Detail section is controlled by the section's depth. Likewise, you can control the white space above and below the headers and footers by adjusting the depth of their sections and the vertical position of the controls within the sections. To create a professional-looking report, you must adjust the controls' alignment and formatting as well as the sections' line spacing.

Aligning Controls Horizontally and Vertically. You align controls by first selecting the rows to align and then aligning the columns. Access provides several control-sizing and alignment options to make the process easier. To size and align the controls that you created, follow these steps:



1. Click the Report View toolbar's Close button to return to Design mode.
2. You can simultaneously adjust the height of all text boxes to fit the font used for their contents. Choose Edit, Select All to select all of the controls in the report.
3. Choose Format, Size, To Fit to adjust the height of the selected controls. Access adjusts all of the controls to the proper height. To deselect all of the controls, click a blank area of the report.
4. Select all labels in the Page Header sections. Choose Format, Align, Top. This process aligns the tops of each selected label with the uppermost selected label. Click a blank area of the report to deselect the labels.
5. Select all text boxes in the Detail section, and repeat step 4 for the text boxes.
6. Select the labels and text boxes in the CategoryID Footer and Report Footer sections and repeat step 4.



7. Select all controls in the Units column. Choose Format, Align, Right so that Access aligns the column to the right edge of the text farthest to the right of the column. Next, click the toolbar's Align Right button to right-align the contents of the labels and text boxes. (The first part of this step aligns the controls themselves to the right-most control, and the second part right-aligns the text or data displayed by the selected controls.)
8. Select all controls in the Price column and repeat step 7.
9. Select all controls in the Values column (except the Page Footer text box) and repeat step 7.



10. Click the toolbar's Report View button to display the report with the improved alignment of rows and columns.

Formatting Controls. As you can see in Figure 14.27, you must revise the formatting of several controls. Although ProductID values are right-aligned, centering or left-justification is more appropriate for values used as codes rather than numbers to total. The repeated dollar signs in the Unit Price field detract from the report's readability, and the Value column's left-justification is inappropriate.

Oracle Ex. 1009, p. 559
Oracle v. DeBellis
IPR2024-00221

To change the Format property of these fields, follow these steps:



1. Click the toolbar's Close button to return to Design mode.



2. Select the ProductID text box in the Detail section and click the toolbar's Center button.
3. Select and then center the CategoryID text box in the CategoryID Footer section.
4. Double-click the Unit Price text box to open its Properties w indow, then click the Format tab of the Properties window.
5. In the Format text box, type **#,#00.00**. This procedure eliminates the dollar sign but preserves the monetary formatting.
6. Repeat steps 4 and 5 for the Values text box. The Detail section doesn't require dollar signs.

Tip

If you select Currency formatting instead of typing **$#,#00.00** to add a dollar sign to the value, your totals do not align. Currency formatting offsets the number to the left to provide space for the parentheses that accountants use to specify negative monetary values.

7. Select the Values subtotal in the CategoryID Footer. Click the Properties window's Format tab and type **$#,#00.00** in the Format field. Accountants use dollar signs to identify subtotals and totals in ledgers.
8. Select the Values grand total in the Report Footer and type **$#,#00.00** as the Format property of the Values grand total.
9. Select the Total Value Percent text box in the CategoryID Footer. Choose Percent from the Format property's drop-down list to display the text box's contents as a percentage.





10. The Values grand total in the Report Footer is the report's most important element, so click the toolbar's Line/Border Color button and then click the black box to give this field a black border. Next, click the Line/Border Width button and then the 2-point border button. This procedure increases the thickness of the border around the grand total.



11. Click the toolbar's Report View button to check your formatting modifications. Click the Bottom of Report page selector button to display the last page of the report (see Figure 14.29).

Adjusting Line Spacing. In the Page Header section, shown in Figure 14.27, the controls are placed further apart than is necessary, and the depth of the controls in the Report Header section is out of proportion with the size of the text. The line spacing of the remainder of the report's sections is satisfactory, but you can also change this spacing. Minimizing line spacing allows you to print a report on fewer sheets of paper.

Oracle Ex. 1009, p. 560
Oracle v. DeBellis
IPR2024-00221

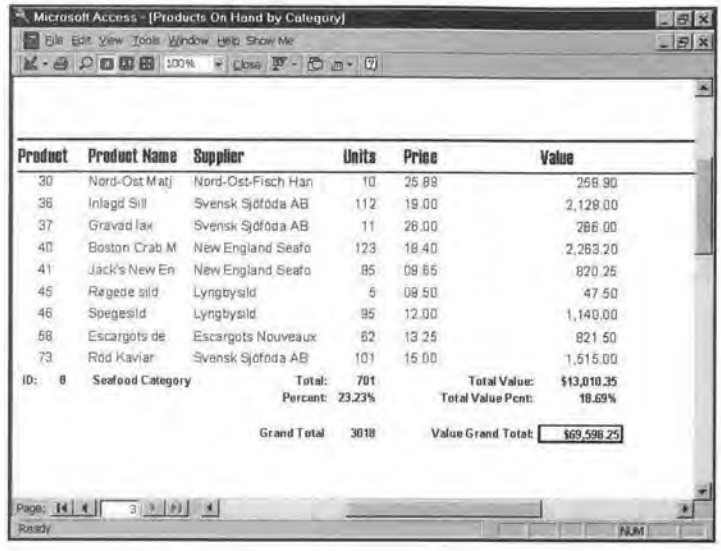


FIG. 14.29 The last page of the report, with the correct Format property assigned to the values.

Tip

You might have to return to Design mode and adjust the width or position of the Subtotals and Grand Totals text boxes to align these values with those for the individual products.

To change the spacing of the report's Page Header and Detail sections, follow these steps:

Tip

You can adjust the size of controls and the line spacing more precisely if you choose Format, Snap to Grid. This command toggles the Snap to Grid feature on and off.

Close

1. Click the toolbar's Close button to return to Design mode.
2. Select all of the labels in the Page Header and move the group as close to the top of the section as possible.
3. Click the bottom line of the Page Header and move the line as close to the bottom of the text boxes as possible. (To select the line, you may have to move the CategoryID Header section downward temporarily.)
4. Click a blank area of the report and then move the CategoryID Header section to the bottom of the labels. You cannot reduce a section's depth to less than the Height property of the label that has the maximum height in the section.
5. Select all of the text boxes in the Detail section, and move those boxes as a group to the top of the section. Move the CategoryID footer up to the bottom of the text boxes.

Oracle Ex. 1009, p. 561
Oracle v. DeBellis
IPR2024-00221

6. Move the line and label in the Report Header section upward to minimize the amount of white space in the Report Header.
7. Click the toolbar's Report View button to check the Page Header depth and line spacing of the Detail section. The spacing shown in Figure 14.30 is close to the minimum that you can achieve. You cannot reduce a section's line spacing to less than that required by the tallest text box or label by reducing the section's Height property in the Properties box because Access rejects the entry and substitutes the prior value.



8. Click the toolbar's Zoom button to display the report in full-page view. Clicking the mouse when the pointer is the magnifying glass symbol has the same effect as clicking the Zoom button. Alternate clicks toggle between full-size and full-page views.



9. Choose File, Save to save your changes.

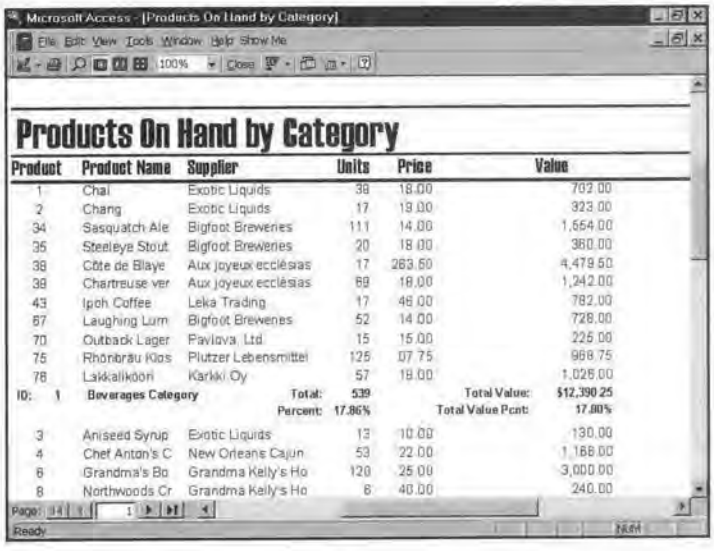


FIG. 14.30 The report in Report view after you adjust the depth of the Report Header, Page Header, and Detail sections.

Adjusting Margins and Printing Conventional Reports

The full-page Report View of the report shows the report as it would print using Access's default printing margins of one inch on the top, bottom, and sides of the report (see Figure 14.31). In the Print Setup dialog, you can adjust the printed version of the report. The procedure for printing a report also applies to printing the data contained in tables and queries as well as single-record or continuous forms.

Oracle Ex. 1009, p. 562
Oracle v. DeBellis
IPR2024-00221

To change the printing margins for a report, follow these steps:

1. Choose File, Page Setup to open the Page Setup dialog (see Figure 14.32).
2. The Page Setup dialog is similar to the Print and Page Setup dialogs of other Windows applications, with a section for printing margins included. To increase the amount of information on a page, decrease the top and bottom margins. By selecting the Print Data Only check box, you can print only the data in the report; the Report and Page Headers and Footers do not print.
3. In the Left text box, type **2.0** to specify a two-inch left margin. In the Right, Top, and Bottom text boxes, type **0.75** inches. Click OK. The full-page view of the report with the revised margins appears (see Figure 14.33).

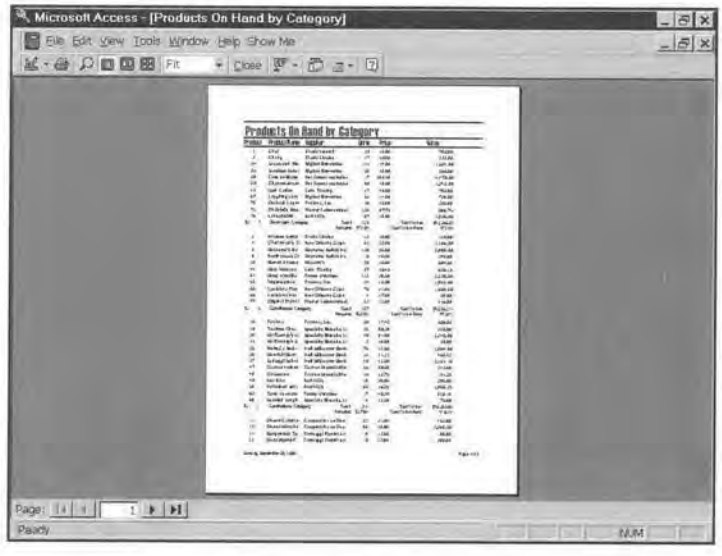


FIG. 14.31 The completed report in full-page view.

Tip

The printing margins that you establish for a report in the Page Setup dialog apply to the active report only; each report has a unique set of margins. When you save the report, Access saves the margin settings.



4. To print the report, click the toolbar's Print button. Access immediately prints the report using the current printer options. If you want to change the selected printer, page orientation, graphics quality, or other printer options, choose File, Print. The standard Print dialog appears for the printer specified in Windows as the default printer. Figure 14.34 shows, for example, the Print dialog for an IBM Laser Printer shared on a network.

Oracle Ex. 1009, p. 563
Oracle v. DeBellis
IPR2024-00221

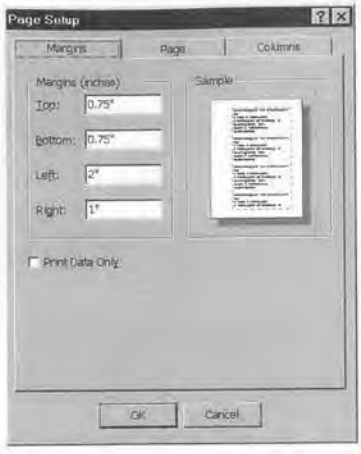


FIG. 14.32 The Page Setup dialog for printing data sheets, forms, and reports.

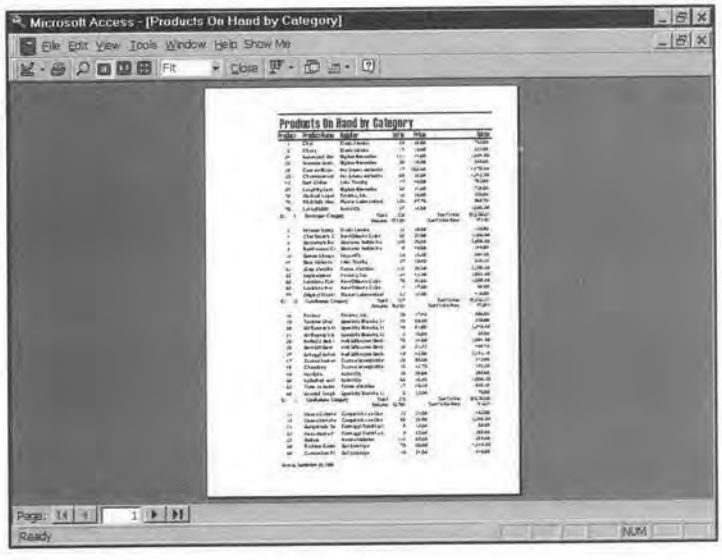


FIG. 14.33 The full-page preview of the report with new printing margins applied.

5. You can choose to print all or part of a report, to print the report to a file for later printing, and select the number of copies to print. By choosing the Properties button, you can change the parameters that apply to the printer you are using. Click OK to print the report.

The Page Setup dialog includes a Page tab that allows you to select the paper size and orientation. The dialog also includes a Columns tab that allows you to establish specifications for printing mailing labels and other multiple-column reports. The next section describes these specifications and how you set them.

Oracle Ex. 1009, p. 564
Oracle v. DeBellis
IPR2024-00221

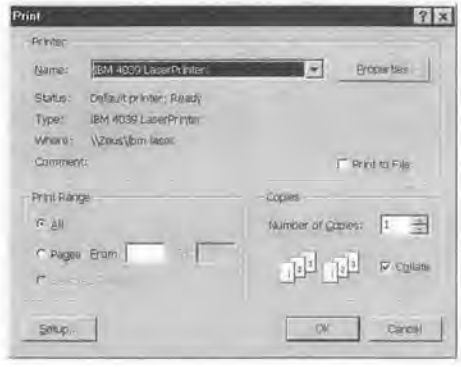


FIG. 14.34 The Print dialog controls printing of datasheets, forms, and reports.

Preventing Widowed Records with the Group Keep Together Property

Access includes a Keep Together property for groups that prevents widowed records from appearing at the bottom of the page. Depending on your report section depths, you might find that only a few records of the next group (called *widowed records*) appear at the bottom of the page. You can force a page break when an entire group does not fit on one page by following these steps:



1. With the report in Design view, click the toolbar's Sorting and Grouping button to open the Sorting and Grouping dialog.
2. Select the field with the group symbol in the selection button that corresponds to the group that you want to keep together. In this example, select CategoryID.
3. Open the Keep Together drop-down list and select Whole Group, as shown in Figure 14.35.

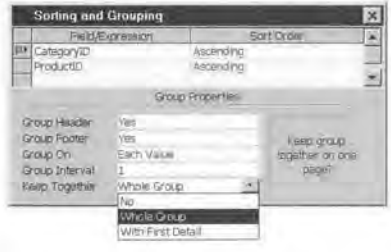


FIG. 14.35 Setting the report group's Keep Together property.



4. Close the Sorting and Grouping dialog, and click the Report View button to see the result of applying the group Keep Together property.

Oracle Ex. 1009, p. 565
Oracle v. DeBellis
IPR2024-00221

Tip

If you want to delete or add a Group Header or Footer singly (rather than in pairs), select Yes or No in the appropriate property field of the Sorting and Grouping dialog.

The Report Wizard makes the other entries in the Sorting and Grouping dialog for you. The next chapter describes how to use the Sorting and Grouping dialog to design reports without the aid of the wizard.

Printing Multicolumn Reports as Mailing Labels

Access allows you to print multicolumn reports. You can create a single-column report with the Report Wizard, for example, and then arrange the report to print values from the Detail section in a specified number of columns across the page. The most common application of multicolumn reports is the creation of mailing labels.

You can create mailing lists with the Report Wizard, or you can start with a blank form. The Report Wizard's advantage is that it includes the dimensions of virtually every kind of adhesive label for dot-matrix or laser printers made by the Avery Commercial Products division. You select the product number of the label that you plan to use, and Access determines the number of columns, rows per page, and margins for the report's Detail section. You can also customize the Mailing Label Wizard for labels with unusual sizes or that other manufacturers produce.

The Northwind Traders database includes a Customer Labels report that you can modify to suit the design of any mailing label. Figure 14.36 shows the Detail section of the Customer Labels report with the font changed to Courier New in a 10-point font and the size of the label adjusted to 2.5×0.833 inches.

You specify the number of columns in a row and the number of rows on a page by selecting settings in the Columns tab in the Page Setup dialog, as shown in Figure 14.37. This dialog appears when you choose File, Page Setup in either Print Preview or Report Design mode.

The dialog's text boxes, check boxes, and option buttons allow you to perform the following procedures:

- The Items Across property sets the number of labels across the page. In this example, this property is set to 3, so the labels will print three across.

Note

The Left and Top margin settings (which you set on the Margins tab of the Page Setup dialog) specify the position at which Access prints the upper-left corner of the first label on the page. For most laser and inkjet printers, these values cannot be less than about 0.25 inches. Labels designed for laser and inkjet printers are die-cut so that the marginal areas remain on the backing sheet when you remove the individual labels.

Oracle Ex. 1009, p. 566
Oracle v. DeBellis
IPR2024-00221

- The Width property in the Item Size group overrides the left margin, and the Height property overrides the bottom margin that you establish in Report Design view only if you don't select Same as Detail to use the margins you set in the Detail section.
- Column Spacing specifies the position of the left edge of columns to the right of the first column.
- Row Spacing and the Height property determine the number of labels that fit vertically on a page and the vertical distance between successive labels. If you set Row Spacing to zero, the depth of your Detail section determines the vertical spacing of the labels.
- The Across, Then Down option causes the labels to print in columns from left to right and then in rows from the top to the bottom of the page. This setting is preferred for mailing labels because it wastes less label stock for continuous-feed printers.
- The Down, Then Across option causes the labels to print in *snaking* column style. The first column is filled from top to bottom, then the second column is filled from top to bottom, and so on.

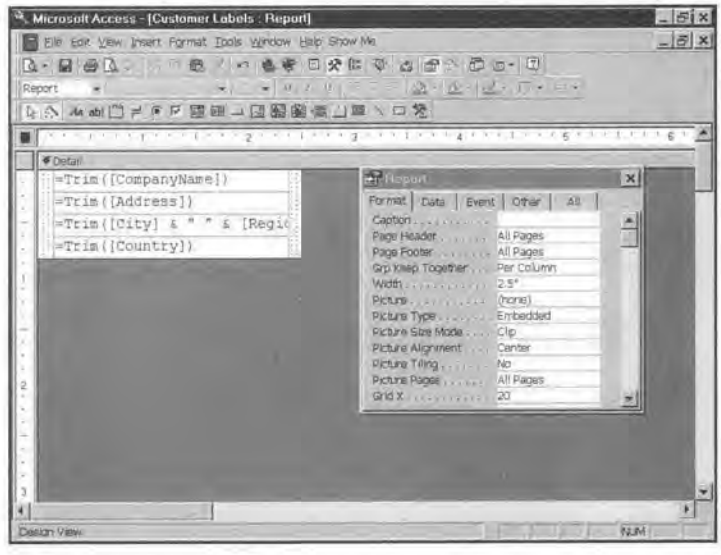


FIG. 14.36 The modified Customer Labels report in Design mode.

You set these properties' values for three columns of 12 labels per page. You can access the settings for all of these columns and pages in the Page Setup dialog's Layout page.

After you set the dimensions of the mailing labels and click OK, the full-size view of the labels appears in Print Preview mode, as shown in Figure 14.38.

Oracle Ex. 1009, p. 567
Oracle v. DeBellis
IPR2024-00221

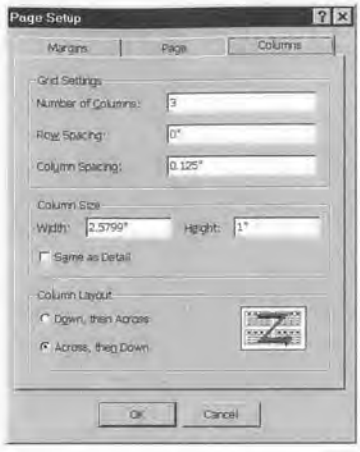


FIG. 14.37 The Columns tab of the Page Setup dialog.



FIG. 14.38 Three-across mailing labels shown in Report View's full-size view.

Click the toolbar's Zoom button to display the full-page layout. To test the label layout properties that you set, print only the label's first page on standard paper or a xerographic duplicate of the label template supplied with labels designed for laser printing.

You might have to make minor alignment adjustments because the upper-left corner of the printer's image and the upper-left corner of the paper might not correspond exactly.

If you select the Down, then Across option, the technique for printing successive Detail rows is identical to that which word processing and page layout applications such as

Oracle Ex. 1009, p. 568
Oracle v. DeBellis
IPR2024-00221

Aldus PageMaker use to create newspaper (snaking) columns. When the first column fills to the page's height, Detail rows fill the next column to the right.

Note

Newspaper columns are suitable for mailing labels but are difficult to format correctly when you convert other kinds of single-column reports to multiple columns. For newspaper columns to operate at all, you must set the Keep Together property of the report's Detail section to No, and then set the Detail section's height so that the field data for a single record appears in a single set of rows. If the Detail data includes Memo fields with variable amounts of text in a text box with the Can Grow property set to Yes, formatting newspaper columns properly becomes almost impossible. Instead of having Access attempt to create newspaper columns, an easier approach is to lay out the Detail section with multiple columns in Design mode.

Oracle Ex. 1009, p. 569
Oracle v. DeBellis
IPR2024-00221

Chapter 15

Preparing Advanced Reports

Access 97's Report Wizard can create reports that you can use "as is" or modify to suit most of your database reporting requirements. In some cases, however, you might have to create reports that are more complex than or differ from those offered by the Report Wizard. For example, you might have to apply special grouping and sorting methods to your reports. Including subreports within your reports requires that you start from a blank report form instead of using the Report Wizard.

To understand fully the process of designing advanced Access reports, you must be familiar with Access functions, which is one of the subjects of Chapter 9, "Understanding Operators and Expressions in Access." You also must understand the methods that you use to create and design forms, which are covered in Chapters 12 and 13. Reports extensively use Access functions such as `Sum()` and expressions like `="Subtotal of" & [Field Name] & ":"`. The toolbox that you use to add controls to forms also adds controls when you create or modify reports. You assign properties of controls, such as labels and text boxes, with the methods that you use with forms. If you skipped Chapters 9, 12, or 13, you might want to refer to the appropriate sections of those chapters whenever you encounter unfamiliar subjects or terminology in this chapter.

Grouping and Sorting Report Data

Most reports you create require that you organize their data into groups and subgroups in a style similar to the outline of a book. The Report Wizard lets you establish the initial grouping and sorting properties for your data, but you might want to rearrange your report's data after reviewing the Report Wizard's first draft.

The Sorting and Grouping dialog (see Figure 15.1) allows you to modify these report properties in Design mode. This section uses the Products On Hand by Category report that you created in the preceding chapter. The sorting and



Oracle Ex. 1009, p. 570
Oracle v. DeBellis
IPR2024-00221

grouping methods described here, however, apply to any report that you create. To display the dialog, open the report in Design view and click the toolbar's Sorting and Grouping button.

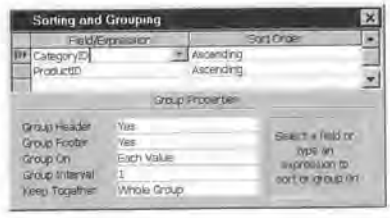


FIG. 15.1 The Sorting and Grouping dialog lets you create or alter report groups and the sort order within groups.

The Sorting and Grouping dialog allows you to determine the fields or expressions on which Access is to group the products, up to a maximum of three levels. You can sort the grouped data in ascending or descending order, but you must select one or the other; "unsorted" is not an option. The Sorting and Grouping symbol in the selection button at the left of the window indicates that Access uses the field or expression in the adjacent column to group the records.

Grouping Data

The method that you use to group data depends on the data in the field by which you group. You can group by categories, in which case a unique value must represent each category. You can group data by a range of values, which usually are numeric but also can be alphabetic. You can use the data in a field to group the data, or you can substitute an expression as the basis for the grouping.

Grouping by Category. When you told the Report Wizard to use CategoryID as the field by which to group, you elected to group by category. You can alter the grouping sequence easily by using the Sorting and Grouping dialog. To group by SupplierID, for example, select SupplierID as the first group field. (When you change the group field, Access automatically renames the Group Header and Footer sections.) In the SupplierID Footer section, change the ID text box's Control Source property to the SupplierID field, and rename the text box to **txtSupplierID**. Next, change the title label in the Report Header to Products on Hand by Supplier, and use the `DLookUp` function to change the text box's expression to the following:

```
="Supplier: " & DLookUp("[CompanyName]","Suppliers",
"[txtSupplierID]=Report!SupplierID")
```

Increase the `DLookUp` text box's width so that more of the supplier's name is visible in the report. The report appears as shown in Figure 15.2. Choose File, Save As, and save the report under the new name **Products On Hand by Supplier**.

Oracle Ex. 1009, p. 571
Oracle v. DeBellis
IPR2024-00221

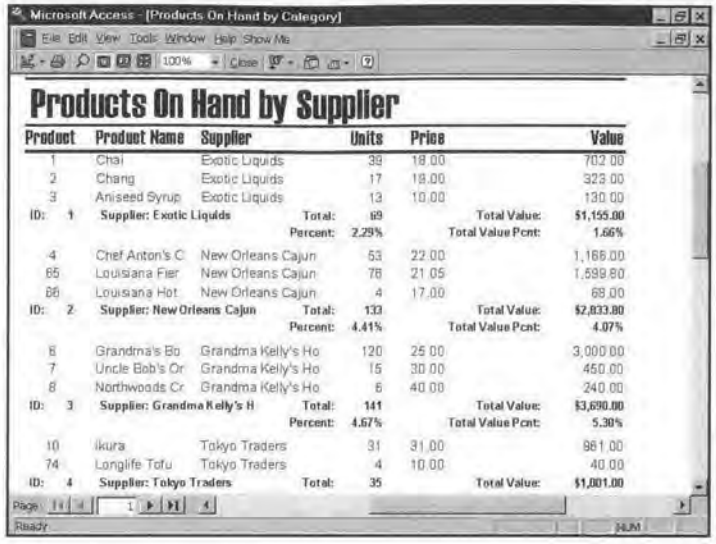


FIG. 15.2 The effect of changing the report grouping so that it displays records by SupplierID.

Note

You cannot let a report's properties or controls limit the number of rows of detail data that a report presents, unless you write a Top*N* or Top*N*Percent query using Access SQL. (Search the online Help for the `TopValues` property to learn more about Top*N* and Top*N*Percent queries.) All rows of a table or query appear somewhere in the report's Detail section if the report includes a Detail section with at least one control. To include only a selected range of dates in a report, for example, you must base the report on a query with the criteria necessary to select the Detail records. If the user is to choose the range of records to include in the report, use a parameter query as the report's data source.

◀◀ See "Text-Manipulation Functions," p. 302



If you use a systematic code for grouping, you can group by the first five or fewer characters of the code. With an expression, you can group by any set of characters within a field. To group by the second and third digits of a code, for example, use the following expression:

```
=Mid([FieldName],2,2).
```

If your table or query contains appropriate data, you can group reports by more than one level by creating subgroups. The Employee Sales by Country report (one of the Northwind Traders sample reports), for example, uses groups (Country) and subgroups (the employee's name—the actual group is an Access expression that combines the FirstName and LastName fields) to organize orders received within a range of dates. Open the Employee Sales by Country report in Design mode to view the additional section created by a subgroup.

Oracle Ex. 1009, p. 572
Oracle v. DeBellis
IPR2024-00221

Grouping by Range. You often must sort reports by ranges of values. (If you opened the Employee Sales by Country report, close this report and reopen the Products On Hand by Category report in Design mode.) If you want to divide the Products on Hand by Category report into a maximum of six sections—each beginning with a five-letter group of the alphabet (A through E, F through J, and so on) based on the ProductName field—the entries in the Sorting and Grouping dialog should look like the entries in Figure 15.3.

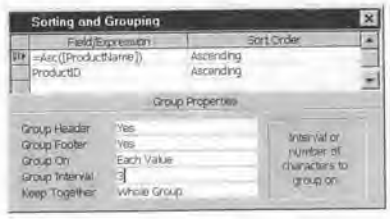


FIG. 15.3 Sorting and Grouping criteria to group records in alphabetical intervals.

Access VBA's `=Asc([ProductName])` function returns the ASCII (numeric) value of the first character of its string argument, the ProductName field. You set the Group On specification to Interval and then set the Group Interval to 3. This setup groups the data into names beginning with A through C, D through F, and so on (see Figure 15.4). You delete all text boxes in the Group Footer because subtotals by alphabetic groups are not significant. Although of limited value in this report, an alphabetic grouping often is useful for grouping long, alphabetized lists to assist readers in finding a particular record.

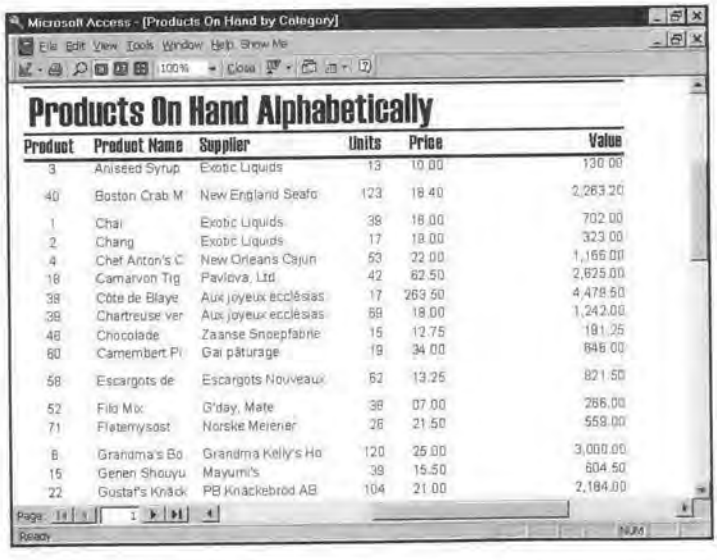


FIG. 15.4 A report that categorizes products by three-letter alphabetic intervals.

Oracle Ex. 1009, p. 573
Oracle v. DeBellis
IPR2024-00221

◀◀ See "Functions," p. 294



If you group data on a field with a Date/Time data type, Access allows you to set the Sorting and Grouping dialog's Group On property to Year, Qtr (quarter), Month, Week, Day, Hour, or Minute. To group records so that values of the same quarter for several years print in sequence, type the following in the Field/Expression column of the Sorting and Grouping dialog:

```
=DatePart("q",[FieldName])
```

Sorting Data Groups

Although most data sorting within groups is based on the values contained in a field, you also can sort by expressions. When compiling an inventory valuation list, the products with the highest inventory value are the most important. The report's users might want these products listed first in a group. This decision requires a record sorting within groups on the expression `=[UnitsInStock]*[UnitPrice]`, the same expression that Access uses to calculate the report's Value column. Figure 15.5 shows the required entries in the Sorting and Grouping dialog.

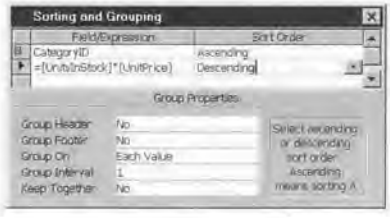


FIG. 15.5 Sorting data by each record on an expression.

The descending sort on the inventory value expression results in the report shown in Figure 15.6. As expected, the products with the highest inventory value appear first in each category.

Working from a Blank Report

Usually, the fastest way to set up a report is to use the Report Wizard to create a basic report and then modify the basic report as described in Chapter 14, "Printing Basic Reports and Mailing Labels," and previous sections of this chapter. If you are creating a simple report, however, it could take you longer to modify a standard report style created by the Report Wizard than it would take to create a report by using the default blank report that Access provides.

The Basis for a Subreport

To create a report to use as the Monthly Sales by Category subreport (rptMonthlyCategorySales) in the following section of this chapter, "Adding and Deleting Sections of Your Report," follow these steps:

Oracle Ex. 1009, p. 574
Oracle v. DeBellis
IPR2024-00221

◀◀ See "Creating a Monthly Product Sales Crosstab Query," p. 362

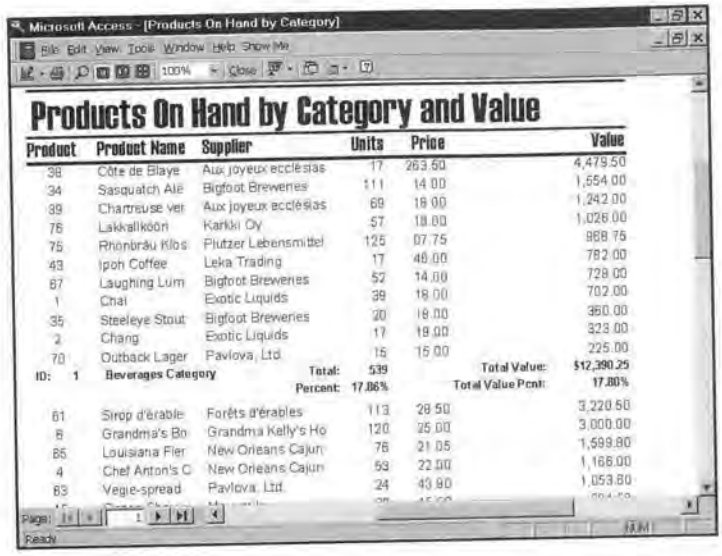


FIG. 15.6 The Products On Hand by Category and Value report, showing records sorted by inventory value within groups.

Queries

1. Close the Products On Hand by Category report, and click the Query tab in the Database window.

Design

2. Select the qry1995MonthlyProductSales query that you created in Chapter 10, "Creating Multitable and Crosstab Queries." Click the Design button. A slightly modified version of this crosstab query serves as the data source for the report of the same name.

3. Change the first column's field name from ProductID to CategoryID by opening the Field list and clicking the CategoryID field name. You need the CategoryID field to link with the CategoryID field in the qryProductOnHand query that the Products On Hand by Category report uses as its data source.

4. Delete the ProductName column. The modified query appears as shown in Figure 15.7.

5. Choose File, Save As and name the modified query **qry1995MonthlyCategorySales**. Your query result set appears as shown in Figure 15.8.

New Object: Report

6. Open the New Object drop-down list on the toolbar, and select New Report from the list. The New Report dialog appears.

Oracle Ex. 1009, p. 575
Oracle v. DeBellis
IPR2024-00221

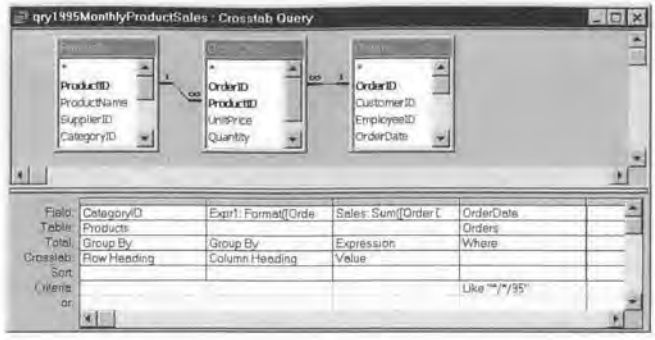


FIG. 15.7 The modified crosstab query for the Monthly Sales by Category subreport.

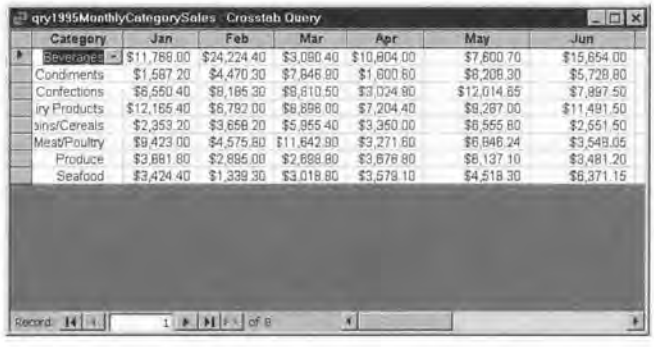


FIG. 15.8 The result set returned by the crosstab query of Figure 15.7.

7. Access automatically selects qry1995MonthlyCategorySales as the query on which to base the report. Select Design view from the list and click OK. Access creates the default blank report shown in Figure 15.9.

Adding and Deleting Sections of Your Report

When you create a report from a blank template or modify a report created by the Report Wizard, you might want to add a new section to the report by using the following guidelines:

- To add Report Headers and Footers as a pair, choose View, Report Header/Footer.
- To add Page Headers and Footers as a pair, choose View, Page Header/Footer.
- To add a Group Header or Footer, click the toolbar's Sorting and Grouping button and set the Group Header or Group Footer property value to Yes.

Figure 15.10 shows a blank report, with the headers and footers for each section that you can include in a report. (Although Figure 15.10 shows only one group, you may add as many group levels as you want to your report.)

Oracle Ex. 1009, p. 576
Oracle v. DeBellis
IPR2024-00221

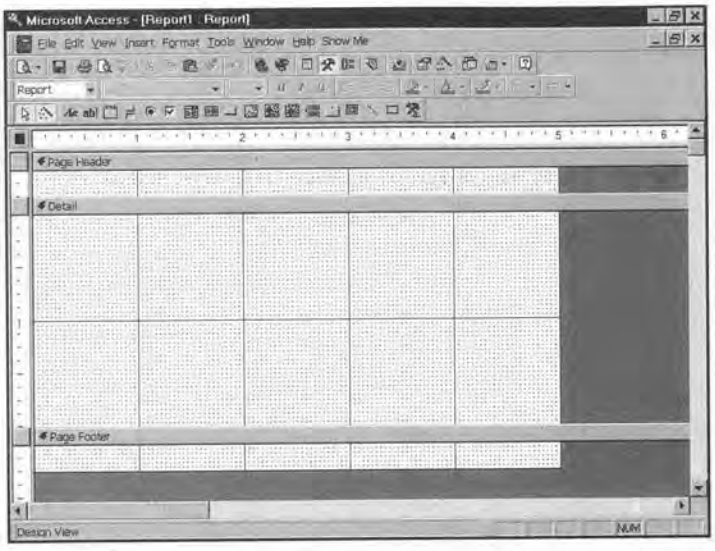


FIG. 15.9 The default report presented by Access after you select Design view in the New dialog.

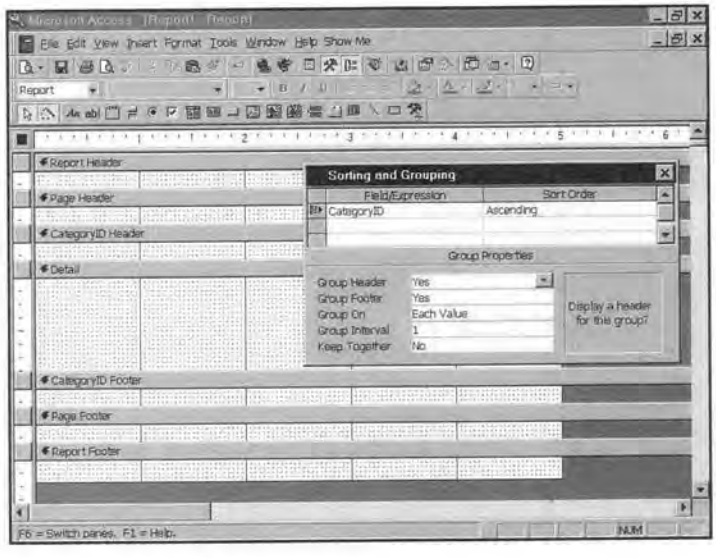


FIG. 15.10 A blank report with all sections added.

If you group the data in more than one level (group, subgroup, sub-subgroup), you can add a Group Header and Footer for each level of grouping. This action adds to your report another pair of sections for each subgroup level.

You delete sections from reports by using methods similar to those that you use to create the sections. To delete unwanted sections, use the following guidelines:

Oracle Ex. 1009, p. 577
Oracle v. DeBellis
IPR2024-00221

- To delete the Detail section or an individual Report Header, Report Footer, Page Header, or Page Footer section, delete all controls from the section, and then drag up the divider bar below so that the section has no depth. To delete a report footer, drag the report's bottom margin to the Report Footer border. These actions do not actually delete the sections, but sections with no depth do not print or affect the report's layout.
- To delete Report Headers and Footers as a pair, choose View, Report Header/Footer. If the Report Header or Footer includes a control, a message box warns you that you will lose the controls in the deleted sections.
- To delete Page Headers and Footers as a pair, choose View, Page Header/Footer. A warning message box appears if either section contains controls.
- To delete a Group Header or Footer, click the toolbar's Sorting and Grouping button and set the Group Header or Group Footer property's value to No.



Note

Page and Report Headers and Footers that incorporate thin lines at the header or footer's upper border can be difficult to delete individually. To make these lines visible, choose Edit, Select All to add sizing anchors to the lines. Hold down the Shift key and click the controls that you want to save in order to deselect these controls. Then press the Delete key to delete the remaining selected lines.

Controlling Page Breaks and the Printing of Page Headers and Footers

The Force New Page and Keep Together properties of the report's Group Header, Detail, and Group Footer sections control manual page breaks. To set these properties, double-click the group's section border to display the section's Properties window. Force New Page causes an unconditional page break immediately before printing the section. If you set the Keep Together property to Yes, and insufficient room is available on the current page to print the entire section, a page break occurs and the section prints on the next page.

To control whether Page Headers or Footers print on the first or last page of a report, choose Edit, Select Report and then click the toolbar's Properties button. You then select the Page Headers and Page Footers printing option in the Format page of the Properties window (see Figure 15.11).

Creating the Monthly Sales by Category Report

Earlier in this chapter, you started creating the Monthly Sales by Category report, which is a good example of a report that you might want to create by starting with a blank report. The report includes information about total monthly sales of products by category. Comparing the monthly sales to the inventory level of a category allows the report's user to estimate inventory turnover rates. This report serves two purposes: as a report and as a subreport within another report. You add the Monthly Sales by Category report as a subreport in the "Incorporating Subreports" section later in this chapter.

Oracle Ex. 1009, p. 578
Oracle v. DeBellis
IPR2024-00221

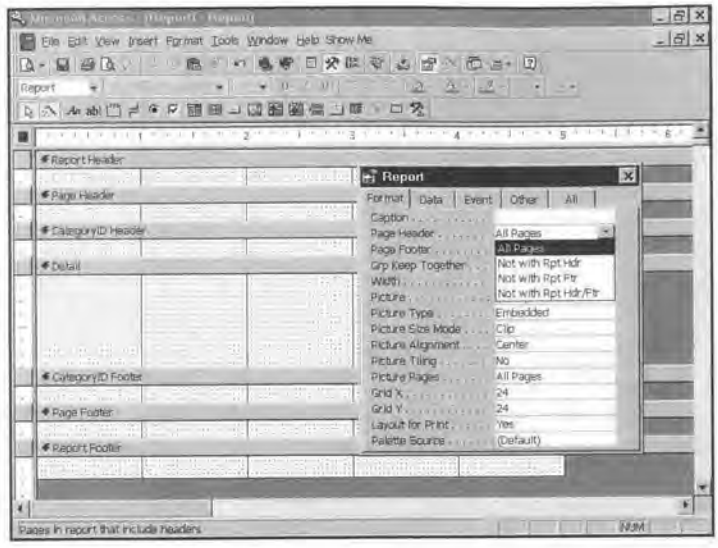


FIG. 15.11 The Properties window for a report, displaying page-section printing options.

The crosstab query that acts as the report's data source is closely related to a report, but the crosstab query doesn't include detail records (see Chapter 10, "Creating Multitable and Crosstab Queries"). Each row of the query consists of subtotals of sales for a category for each month of the year. One row appears below the inventory value subtotal when you link the subreport to the main report, so the report needs only a Detail section. Each detail row, however, requires a header label to print the month. The CategoryID field is included so that you can verify that the data is linked correctly.

To create the Monthly Sales by Category report (and subreport), follow these steps:

1. Delete all sections of your blank report except the Detail section. By default, blank reports have 24×24 grid dots, and Snap to Grid is selected.
2. Drag down the bottom margin of the Detail section so that the section has an inch or two of depth. You need maneuvering room to relocate the text boxes and associated labels that you add in the following steps. Drag the right margin of the Detail section to the right so that the report is 6.5 inches wide.



3. Click the toolbar's Sorting and Grouping button to display the dialog, and select CategoryID as the field to use to sort the data with a standard ascending sort. Close the Sorting and Grouping dialog.



4. Click the toolbar's Field List button, select CategoryID, and drag the field symbol to the Detail section.
5. Click the CategoryID label, and relocate the label to the Detail section's upper left so that the CategoryID combo box is directly underneath it. (CategoryID appears as a dummy combo box, not a text box, in Report Design view because CategoryID is a lookup field.) Adjust the depth of the label and text box to 0.2 inches (four grid dots) and the width to 1 inch. Edit the label's text to **ID**.

Oracle Ex. 1009, p. 579
Oracle v. DeBellis
IPR2024-00221

6. Click and drag the field list's Jan field to the right of the CategoryID field. Move the label to the top of the section, adjacent to the right border of the field to its left. Move the text box under the label. Adjust the label and text box depth to four dots and the width to 16 dots. Edit the label's text to delete the colon.

7. Repeat step 6 for the month fields of Feb through Jun. The report design now appears as shown in Figure 15.12.

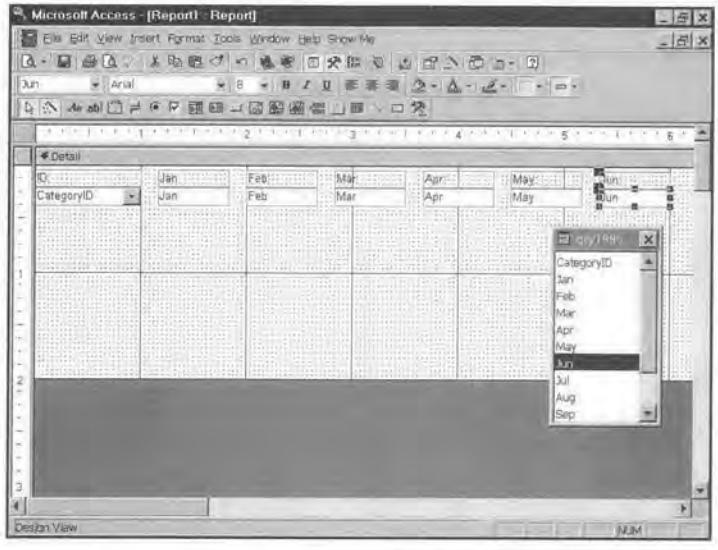


FIG. 15.12 The report with labels and text boxes added for the first six months of 1995.

8. Click each label while holding down the Shift key so that you select all labels (but only the labels).

9. Click the toolbar's Bold button to add the bold attribute to the labels. Then click the Center button to center the labels above the text boxes.




10. Select the CategoryID text box, and click the toolbar's Bold button.

11. Choose Edit, Select All, and drag the labels and text boxes so that the tops of the labels are two dots down from the top of the Detail section. Click a blank area of the report to deselect the controls.

12. If the toolbox is invisible, click the Toolbox button of the toolbar. Click the Line tool and add a line at the top edge of the labels. Drag the line's right-end handle to the right edge of the Jun text box.




13. Click the drop-down list of the Line/Border Width button on the toolbar, and click the 2-point line-thickness button.



14. Repeat steps 12 and 13 for another identical line, but add the new line under the labels (and above the text boxes).

Oracle Ex. 1009, p. 580
Oracle v. DeBellis
IPR2024-00221

15. Drag the Detail section's margins to within two dots of the bottom and right edge of the controls. The report's design appears as shown in Figure 15.13.

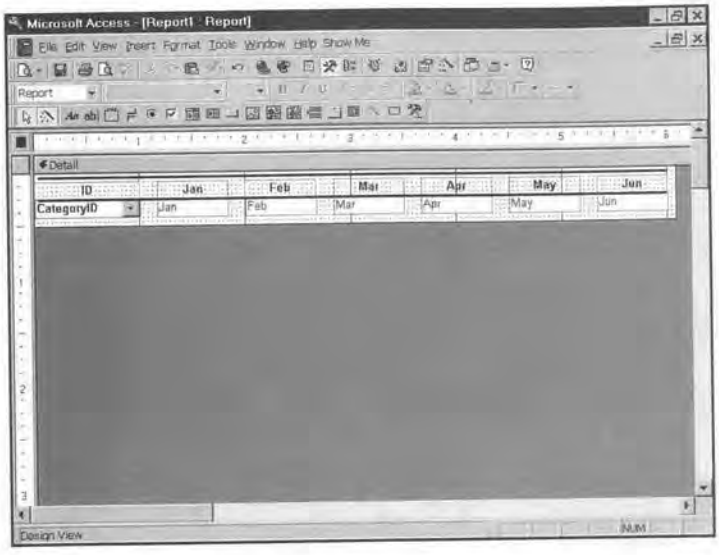


FIG. 15.13 The layout of the new report's Detail section.

16. Click the toolbar's Report View button to verify the design. The full-size view of the report appears (see Figure 15.14).

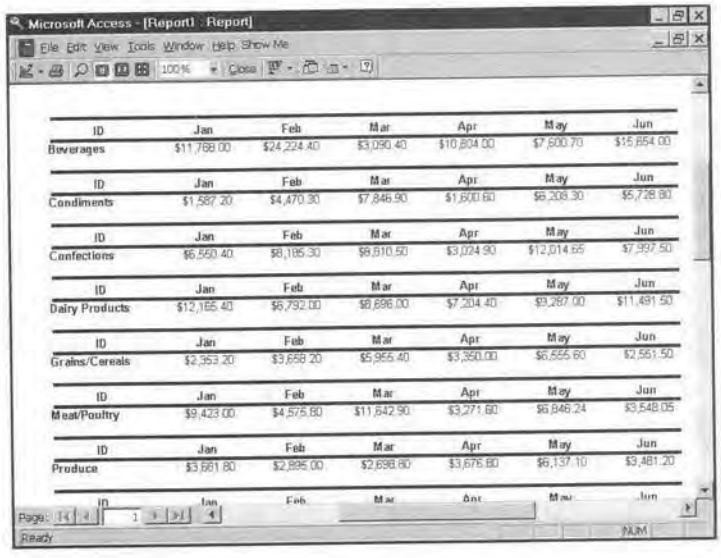


FIG. 15.14 The monthly category sales report in Print Preview mode.

17. Choose File, Save As and type **rptMonthlyCategorySales** as the report's name.

Oracle Ex. 1009, p. 581
Oracle v. DeBellis
IPR2024-00221

To add to your report the remaining months of the year, follow these steps:

1. To accommodate another row of labels and text boxes, increase the depth of the Detail section by dragging the bottom margin down (about 1 inch).
2. Choose Edit, Select All.
3. Click the Copy button on the toolbar, or press Ctrl+C to copy the labels and text boxes to the Clipboard.



4. Click the Paste button on the toolbar, or press Ctrl+V to paste a copy of the labels and text boxes to the Detail section.



5. Move this copy under the original labels and text boxes. Space the copy two grid dots below the originals, as shown in Figure 15.15.

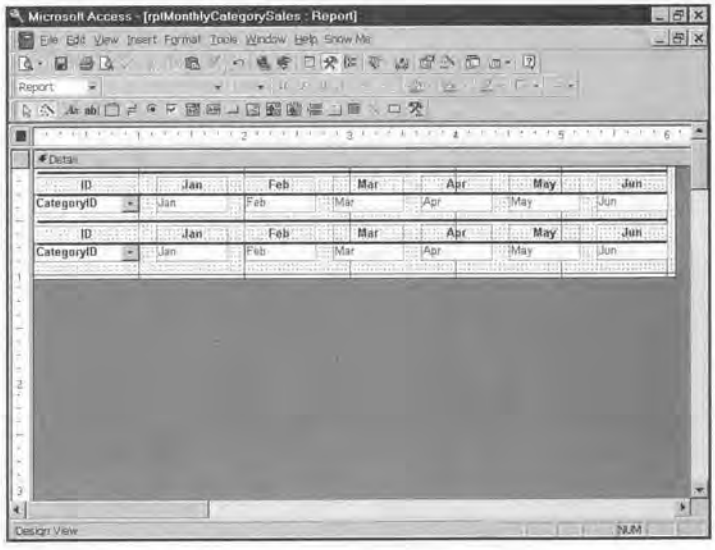


FIG. 15.15 The rptMonthlyCategorySales report after adding a copy of all text boxes and labels.

6. Click a blank area of the report to deselect the controls, and then select the new CategoryID text box. Delete the CategoryID text box. When you delete this text box, you also delete the associated label.
7. Edit both the labels and text boxes to display Jul through Dec. (Access automatically resizes the labels to fit the new text value and automatically sets the text boxes' Control Source property to match the field name you type into the text box.)
8. Delete the ID label in the first row, and drag the CategoryID text box to the position that its label formerly occupied.

Oracle Ex. 1009, p. 582
Oracle v. DeBellis
IPR2024-00221

9. Drag up the bottom margin to within two dots of the bottom of the text boxes in the second row. The final design appears in Figure 15.16.



10. Click the toolbar's Print Preview button to display the double-row report (see Figure 15.17).

11. Close the rptMonthlyCategorySales report and save the changes.

The technique of copying controls to the Clipboard, pasting copies to reports, and then editing the copies is often faster than creating duplicate controls that differ from one another only in the text of labels and text boxes.

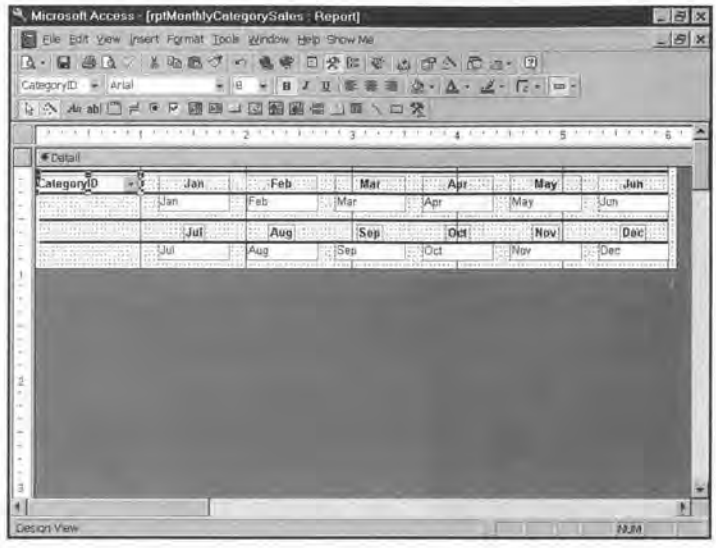


FIG. 15.16 The final design of the rptMonthlyCategorySales report.

Incorporating Subreports

Reports, like forms, can include subreports. Unlike the Form Wizard, however, the Report Wizard offers no option of automatically creating reports that include subreports. You can add subreports to reports that you create with the Report Wizard, or you can create subreports from blank reports, as shown in the preceding section, "Working from a Blank Report."

Adding a Linked Subreport to a Bound Report

If a main report is bound to a table or query as a data source, and the subreport's data source can be related to the main report's data source, you can link the subreport's data to the main report's data.

Oracle Ex. 1009, p. 583
Oracle v. DeBellis
IPR2024-00221

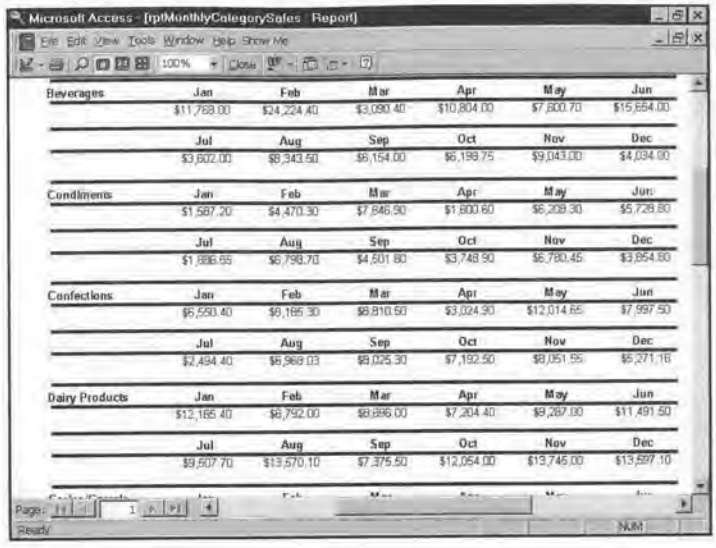


FIG. 15.17 The rptMonthlyCategorySales report in Print Preview.

To add and link the rptMonthlyCategorySales report as a subreport to the Products On Hand by Category report, for example, follow these steps:

1. Open the Products On Hand by Category report in Design mode.



2. Drag down the top of the Page Footer border to make room for the subreport in the CategoryID Footer section (about 0.5 inches).



3. Click the toolbar's Database Window button. If the Database window is maximized, click the Restore button.



4. Click and drag the small Report icon from the left of the rptMonthlyCategorySales report to a location inside the CategoryID Footer section. Drop the icon below the CategoryID text box. You must be able to see the location where you want to drop the new subreport in the Report Design window, as shown in Figure 15.18.
5. At the point where you drop the icon, Access creates a subreport box that is similar to a text box. This subreport box has an associated label (see Figure 15.18). Delete the label.
6. Click the Maximize button to restore the Report Design window.
7. Adjust the CategoryID Footer's depth to provide about 0.1-inch margins above and below the section's controls. The report appears in Design view as shown in Figure 15.19.
8. You need to link the data in the subreport to the data of the main report so that only the sales data corresponding to a specific group's CategoryID value appears on screen. Select the subreport box and click the Properties button to display the

Oracle Ex. 1009, p. 584
Oracle v. DeBellis
IPR2024-00221

subreport's Properties window. Click the Data tab, type **CategoryID** as the value of the Link Master Fields property, and type **CategoryID** as the value of the Link Child Fields property.

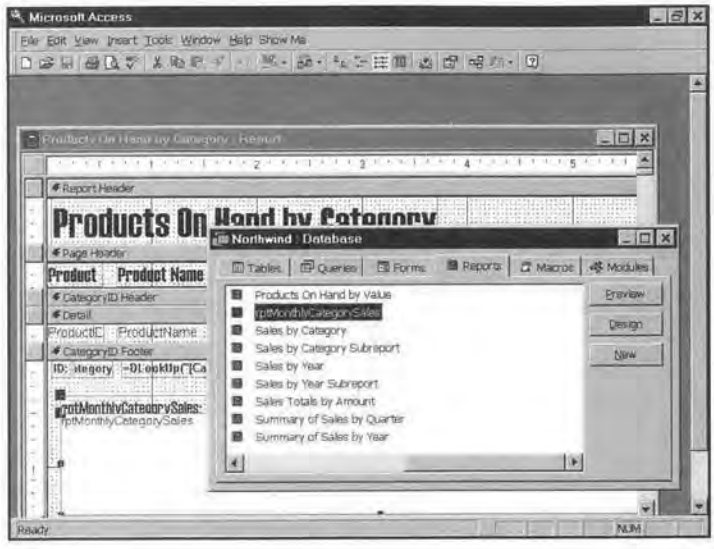


FIG. 15.18 Dragging and dropping a report as a subreport within another report.

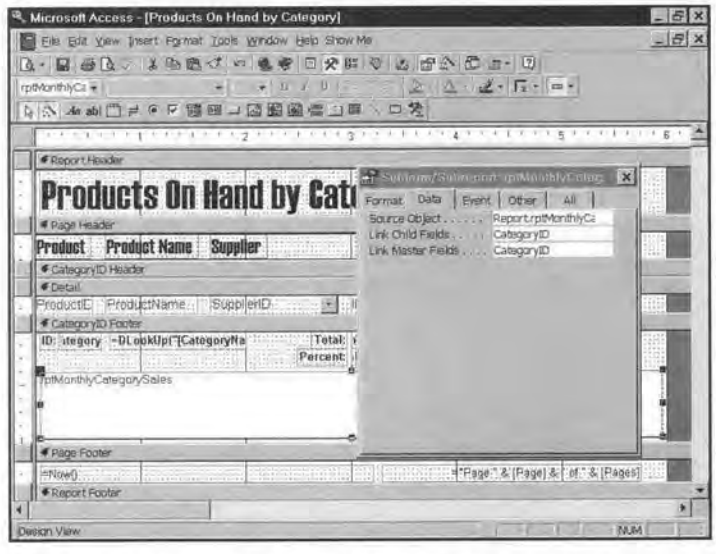


FIG. 15.19 Setting the properties to link the subreport to the main report.

Oracle Ex. 1009, p. 585
Oracle v. DeBellis
IPR2024-00221

Access attempts to create the link. If the main report and subreports are based on tables and a relationship is set between the tables, Access creates the link to the related fields. If the main report is grouped to a key field and the subreport's table or query contains a field of the same name and data type, Access creates the link.

9. Click the toolbar's Print Preview button to display the report in the full-size view. The subreport appears as shown at the bottom of Figure 15.20. Click the page selector buttons to view other parts of the subreport to confirm that the linkage is correct.



10. Choose File, Save to save the changes.

You can add and link several subreports to the main report if each subreport has a field in common with the main report's data source.

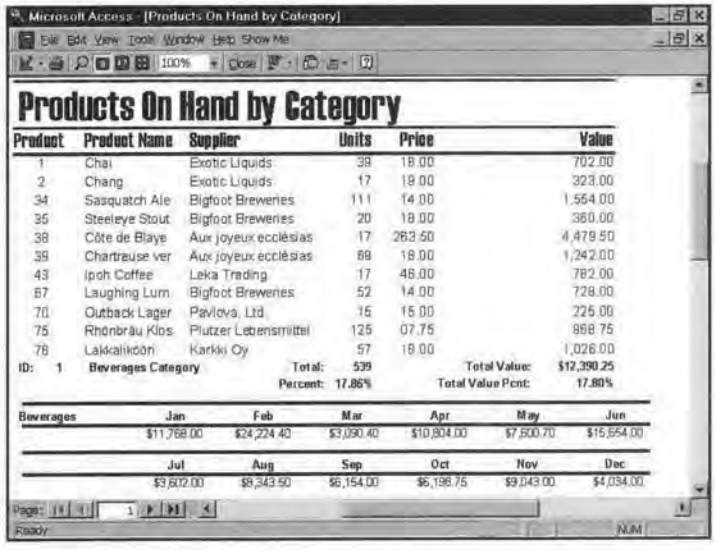


FIG. 15.20 The rptMonthlyCategorySales subreport linked to the Products On Hand by Category report.

Note

You can use calculated values to link main reports and subreports. Calculated values often are based on time—months, quarters, or years. To link main reports and subreports by calculated values, you must create queries for both the main report and subreport that include the calculated value in a field, such as Month or Year. You create the calculated field in each query by using the corresponding Access date function, `Month` or `Year`. To group by quarters, select `Interval` for the Group On property and set the value of the Group Interval property to 3. You cannot use `Qtr` as the Group On property because the calculated value lacks the Date/Time field data type.

Oracle Ex. 1009, p. 586
Oracle v. DeBellis
IPR2024-00221

Troubleshooting

When I try to create a link between the main report and subreport, I get a `Can't evaluate expression` *error message.*

The most likely cause is that you are trying to create a master-child (or, more properly, parent-child) link with an incompatible data type. Parent-child linkages are similar to joins of queries that use the WHERE *SubreportName*.*FieldName* = *ReportName*.*FieldName* criterion. As with joins, the data types of the linked fields of tables or columns of queries must be identical. You cannot, for example, link a field of the Text data type with a field of the Integer data type, even if your text field contains only numbers. If you use an expression to create the link, the data type that the expression returns must match the field value. You can use the data type conversion functions described in Chapter 9, "Understanding Operators and Expressions in Access," to change the data type that the expression returns to that of the linked field. For example, you can link a text field that contains numbers to a field of the Long Integer data type by entering **=CLng**(*TextField*) as the linking value.

Using Unlinked Subreports and Unbound Reports

Most reports that you create use subreports that are linked to the main report's data source. You can, however, insert independent subreports within main reports. In this case, you don't enter values for the Link Child Fields and Link Master Fields properties—in fact, if Access adds values, you delete them. The subreport's data source can be related to or completely independent of the main report's data source. Figure 15.21 shows the affect of including an unlinked subreport. Figure 15.21 shows a portion of page 2 of the rptMonthlyCategorySales subreport within the Products on Hand by Category report after deleting the CategoryID values of the Link Child Fields and Link Master Fields properties. Notice that, without the link, the subreport displays all records instead of just those records related to the particular category in which subreport appears.

You can add multiple subreports to an unbound report if all of the subreports fit on one page of the report or all subreports fit across the page. In the latter case, you can use landscape printing orientation to increase the available page width. To create an unbound report with multiple subreports, follow these steps:



1. Click the Report tab in the Database window, and then click the New button to display the New Report dialog.
2. Keep the text box for the New Report dialog's data source blank, select Design view in the list, and then click OK. This action creates an unbound report.



3. Click the toolbar's Database Window button to display the Database window, and drag the Report icon for the first subreport to the blank form's Detail section.
4. Drag the Report icon for the second subreport to the blank form's Detail section. If the two subreports fit vertically on one page, place the second subreport below the first subreport. If either of the two subreports requires more than a page, place the second subreport to the right of the first. In this case, you must add column labels for the subreports in the main report's Page Header section so that each page identifies the columns.

Oracle Ex. 1009, p. 587
Oracle v. DeBellis
IPR2024-00221

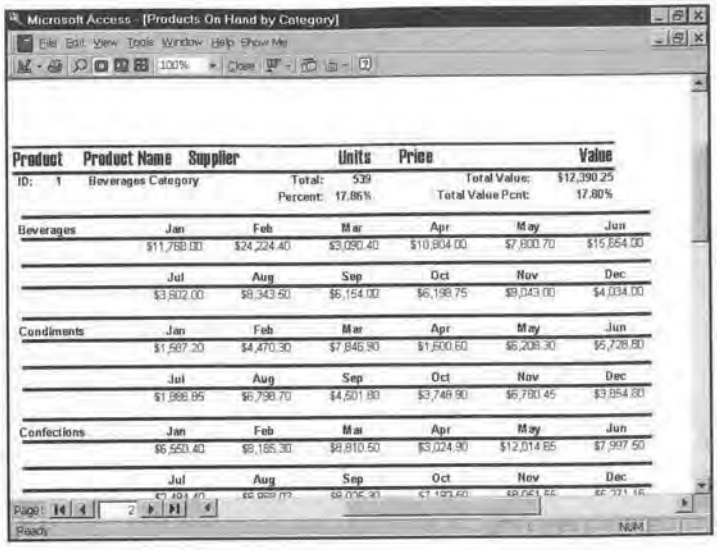


FIG. 15.21 The appearance of the rptMonthlyCategorySales subreport for the beverages category when the linking properties have been deleted.

Note

You can also add subreports to a report by using the Toolbox's subform/subreport tool. Use the same procedures for adding subforms that you learned in Chapter 13, "Designing Custom Multitable Forms."

Adding Other Controls to Reports

Access places no limit on the toolbox controls that you add to reports. So far, the controls that you have modified or added have been limited to labels, text boxes, lines, and the combo boxes that Access places automatically for fields configured as lookup fields. These four kinds of controls are likely to comprise more than 90 percent of the controls used in the reports you create. Controls that require user interaction, such as lists and combo boxes, can be used in a nonprinting section of the report, but practical use of these controls in reports is limited. The following list describes other controls that you might want to add to reports:

- *Bound object frames* print the contents of the OLE Object field data type. An OLE object can be a still or animated graphic, a video clip, a waveform or CD audio, or even MIDI music. Reports are designed only for printing, so animated graphics, video, and sound are inappropriate for reports.



- *Unbound object frames* display OLE objects created by server applications, such as Microsoft Graph 97 (included with Access), Windows Paint, Excel, or the Microsoft WordArt or Equation Editor OLE applets included with Microsoft Word. Usually,



Oracle Ex. 1009, p. 588
Oracle v. DeBellis
IPR2024-00221

you place unbound objects in the report's Form Header or Form Footer section, but you can add a logo to the top of each page by placing the image object in the Page Header section. A graph or chart created by the Chart Wizard is a special kind of unbound OLE object.



- *Lines* and *rectangles* (also called *shapes*) create decorative elements on reports. Lines of varying widths can separate the sections of the report or emphasize a particular section.



- *Check boxes* and *option buttons* can be used to indicate the values of Yes/No fields or within group frames to indicate multiple-choice selections. Group frames, option buttons, and check boxes used in reports indicate only the value of data cells and do not change the values. Reports seldom use toggle buttons.



- *Command buttons* execute Access VBA procedures.

Bound and unbound object frames are the subject of Chapter 20, "Adding Graphics to Forms and Reports." Access VBA programming is the topic of Part VII, "Programming, Distributing, and Converting Access Applications."

Sending Reports by Microsoft Exchange and Outlook

If you have installed the Microsoft Exchange client (Inbox) along with the Microsoft Outlook messaging application, you can send a report to others as an attachment to a Microsoft Mail or other Microsoft Exchange message. Exchange lets you send the report output as a fax, through CompuServe mail, Internet Mail, Microsoft Network mail, or through any other messaging system that you have configured to work with Microsoft Exchange. To send a report by Microsoft Exchange, follow these steps:

Note

A full discussion of Microsoft Exchange is beyond the scope of this book. Search the Windows online Help for more information about Microsoft Exchange.

Note



You can also publish your reports as HTML pages to make them available to Internet users or your company's intranet users. Part IV of this book describes using HTML in Access.





1. In Print Preview mode, open the report that you want to send. (You do not need to have Microsoft Mail or other Exchange client software running when you create the message, although it must be installed and correctly configured.) You can also select the report that you want to send in the Reports tab of the Database window.
2. Choose File, Send to display the Send dialog (see Figure 15.22).

Oracle Ex. 1009, p. 589
Oracle v. DeBellis
IPR2024-00221

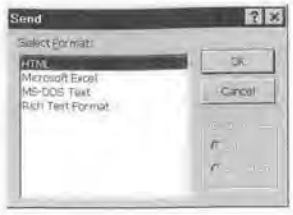


FIG. 15.22 Choosing the format for the Microsoft Exchange attachment.

3. In the Select Format list box (refer to Figure 15.22), select the format in which you want to send the report file. This example sends the Products on Hand by Category report in Microsoft Excel 97 (.xls) format.

4. Click OK to create the message. A progress-reporting dialog appears while Access creates the Excel worksheet (.xls) file. When the process is complete, Microsoft Exchange's Choose Profile dialog appears (see Figure 15.23).



FIG. 15.23 Choosing the Microsoft Outlook profile for sending a report as a mail attachment.

5. In the Profile Name drop-down list box, select the profile name that you want to use for this message, and then click OK. (Refer to Exchange's online Help for information about Exchange's profiles.) The Microsoft Exchange's New Message window now appears (see Figure 15.24).

6. Enter the recipients' names, a subject, and an optional transmittal message, as shown in Figure 15.24. Click the Send button (the left-most button of the upper toolbar) to send the message with the attached report file.

Note

Depending on your report's formatting, some of the field values might not appear in an Excel worksheet file. For example, the subtotals for the Values fields (as well as both percentage fields) are missing. However, you can easily reconstruct these fields in Excel. Access cannot incorporate subreports that are contained in reports printed to files of any of the three available formats. You must provide individual columns for subtotals and other section footers to ensure that the values appear in an .xls file.

Oracle Ex. 1009, p. 590
Oracle v. DeBellis
IPR2024-00221

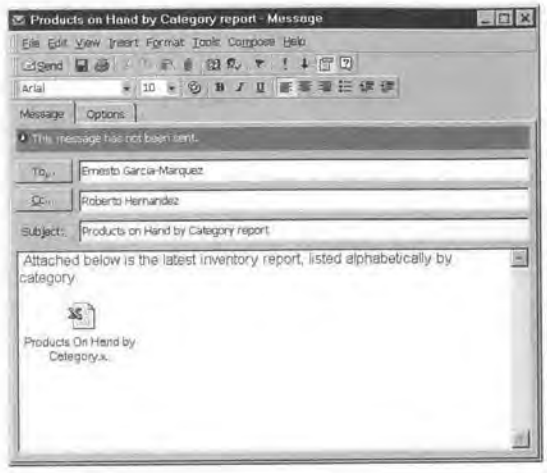


FIG. 15.24 Completing the Microsoft Outlook message.

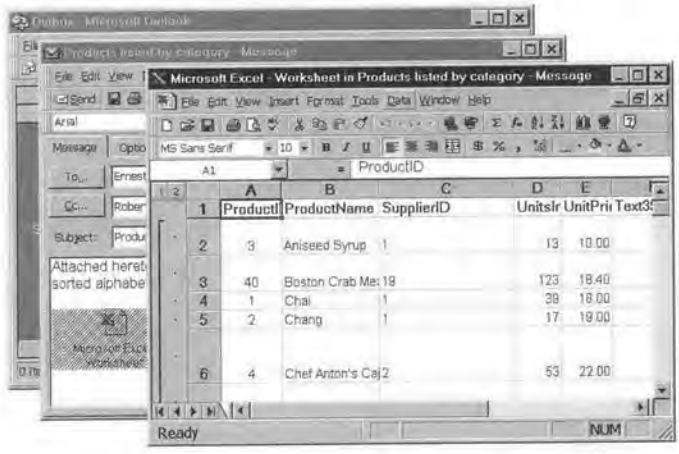
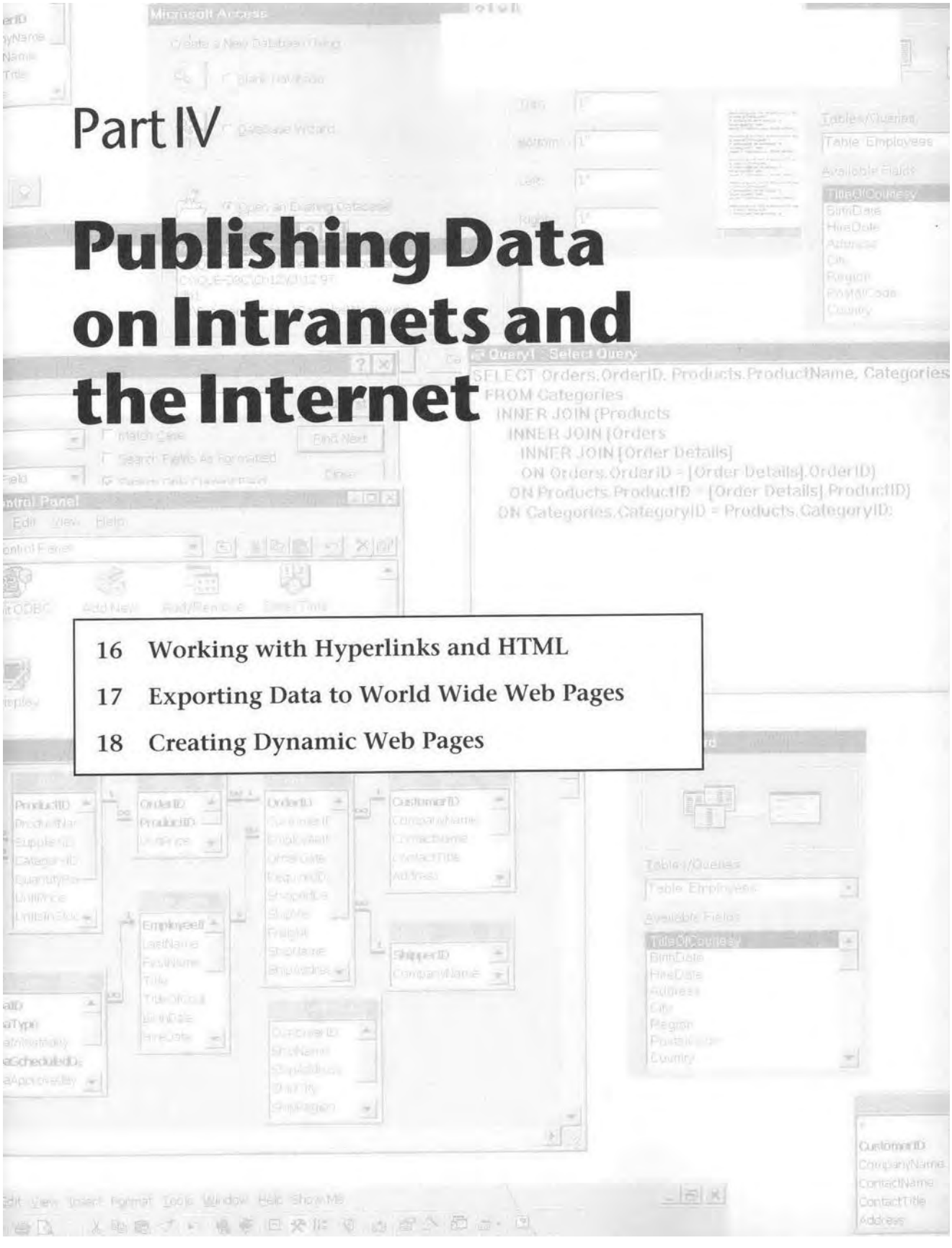


FIG. 15.25 The appearance of the report attachment in Microsoft Excel 97.

Oracle Ex. 1009, p. 591
Oracle v. DeBellis
IPR2024-00221



Part IV

Publishing Data on Intranets and the Internet

16 Working with Hyperlinks and HTML

17 Exporting Data to World Wide Web Pages

18 Creating Dynamic Web Pages

Oracle Ex. 1009, p. 592
Oracle v. DeBellis
IPR2024-00221

Chapter 16

Working with Hyperlinks and HTML

The Internet has made the most profound change in the direction of the computer software industry since the introduction of Windows 3.0. Virtually every new software product published in 1996 had some connection with the Internet, even if tenuous, and there currently is no indication of a slackening in this trend. All of the members of Microsoft Office 97 are "Internet-enabled" to a varying degree. Word 97 lets you convert conventional .doc files to .htm Web pages. PowerPoint 97 encourages you to create presentations for delivery via the World Wide Web. Excel 97 lets you export worksheets to HTML tables. Each of these enhancements rely on *hyperlinks* that lead from one document to another related document and *HyperText Markup Language* (*HTML*), a variant of the *Standard Generalized Markup Language* (*SGML*) designed for text document formatting.

Databases currently play an important role in Internet publishing and are destined to become the backbone of commercial transactions conducted on the Internet (called *electronic commerce* or *e-commerce*). Private intranets, which quickly deliver up-to-date information within an organization, have even greater potential for exploiting relational database technology. Microsoft has equipped Access 97 with a variety of enhancements for distributing data via private intranets and the public Internet. Access 97 also includes features that enable your Access applications to interact directly with documents located on the World Wide Web. This chapter, plus the two chapters that follow, shows you how to take best advantage of these new Access 97 features.

Putting Microsoft's Internet Program in Perspective

The astounding growth in the use of the Internet that occurred in 1994 and early 1995 appeared to have gone unnoticed by Microsoft. The World Wide Web, with its hyperlinked documents and easy-to-use browsers, had extended the reach of the Internet to ordinary computer users. Sales of Internet server software and hardware mushroomed. Commercial versions of Web browsers,

Oracle Ex. 1009, p. 593
Oracle v. DeBellis
IPR2024-00221

based on the original NCSA Mosaic browser design, gained a major market presence. Ultimately, Netscape Communications, Inc. gained the lion's share of the rapidly expanding Internet server and Web browser software market. The consensus of the computer trade press by late 1995 was that Microsoft had "missed the Internet boat."

Bill Gates announced on December 7, 1995, that Microsoft Corporation would, henceforth, "embrace and extend the Internet." To achieve his goal, Gates initiated a major reorganization that gave Internet-related development programs top priority at Microsoft. The result of diverting a large part of Microsoft's considerable technical and economic resources to the Internet was a 1996 flood of new products and technology incentives, plus Internet upgrades to existing products. Just keeping up with Microsoft's product announcements bordered on a full-time occupation. What's more, Microsoft distributed most of its new Internet-related products, often in the beta (or pre-beta) testing stage, to users at no charge.

An Overview of Microsoft's Internet Products

Virtually all of Microsoft's newly-released Internet-related products incorporate or employ database technology. Following is a list of the most important new Internet-related products introduced by Microsoft as of late 1996:

- *Internet Explorer 3.0 (IE)* is Microsoft's entry in the Web browser sweepstakes at the time this book was written. Unlike Netscape Navigator 3.0, which has a retail list price of $49.95, IE 3.0 is free. Office 97 includes IE 3.0, and IE 3.0 is required to take full advantage of the Office 97 Internet feature set, but Office 97 applications also support Netscape Navigator 3.0 as your preferred browser. Figure 16.1 shows Internet Explorer 3.0 displaying the home page of the Volcano Coffee Company example application included with Internet Information Server 2.0.

  Microsoft has succeeded in convincing all of the major online services and many large Internet Service Providers (ISPs) to make IE 3.0 their preferred browser. The next version of IE will be incorporated directly within the Windows 95 and Windows NT operating system. Ultimately, all PC operating systems will have built-in browser capability.


On the Web

Download the latest Internet Explorer from:

http://www.microsoft.com/ie/

- *NetMeeting 2.0* is an extension to IE 3.0 for collaborative computing. NetMeeting currently includes Internet telephony for voice communication, multiuser data conferencing for application sharing, an electronic whiteboard for diagramming, chat, and binary file transfer. Future versions of NetMeeting are expected to include videoconferencing capability over ISDN and faster connections. NetMeeting uses a database to maintain a list of locations of other NetMeeting users.


On the Web

You can download your own copy of NetMeeting from:

http://www.microsoft.com/netmeeting/

Oracle Ex. 1009, p. 594
Oracle v. DeBellis
IPR2024-00221

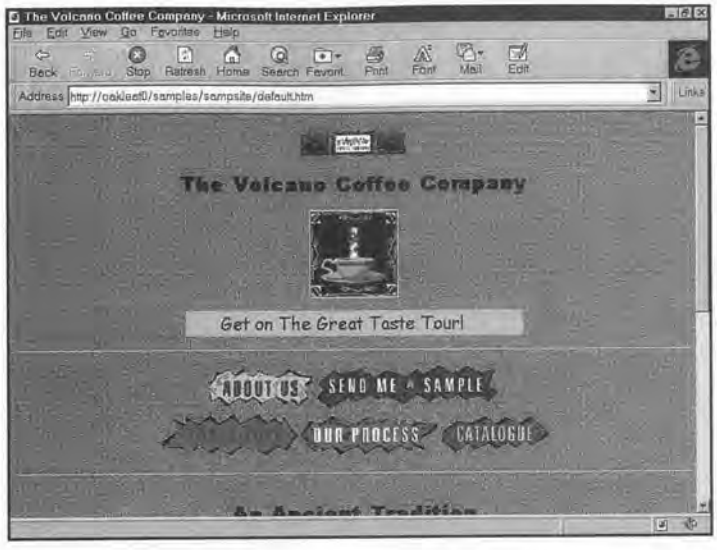


FIG. 16.1 Internet Explorer 3.0 displaying the home page for the Volcano Coffee Company example included with Internet Information Server 2.0.

- *NetShow* delivers live and on-demand multimedia content over the Internet and intranets with IE 3.0. NetShow currently enables multicasting of live audio and data, plus storing and streaming of graphics (*illustrated audio*). Future versions of NetShow will include streaming video content for high-bandwidth connections. Databases store locations of NetShow sites and program guides for upcoming live NetShow performances.

  More information on NetShow is available from:

  http://www.microsoft.com/netshow/

- *Internet Information Server 2.0 (IIS)* is included with Windows NT Server 4.0, a high-performance Internet/intranet delivery system that's closely integrated with Windows NT 4.0's security features. IIS 2.0 includes the Internet Database Connector (IDC), which is the subject of Chapter 18, "Creating Dynamic Web Pages." Figure 16.2 shows a Web page created from a simple IDC query against SQL Server 6.5's Pubs database.

  Get the latest information about Internet Information Server at:

  http://www.microsoft.com/ntserver/iis/

- *Proxy Server*, originally code-named *Catapult*, provides secure access to the Internet via internal local-area networks (LANs). Proxy Server, which now is a member of Microsoft BackOffice 2.5, improves performance for networked users by *caching* (holding local copies of) commonly accessed Internet content. You can download a free 60-day evaluation copy of Proxy Server from Microsoft's Web site. (Windows NT 4.0 Service Pack 1 is required to run Proxy Server.)

Oracle Ex. 1009, p. 595
Oracle v. DeBellis
IPR2024-00221


On the Web

The details on Proxy Server are found at:

http://www.microsoft.com/proxy/

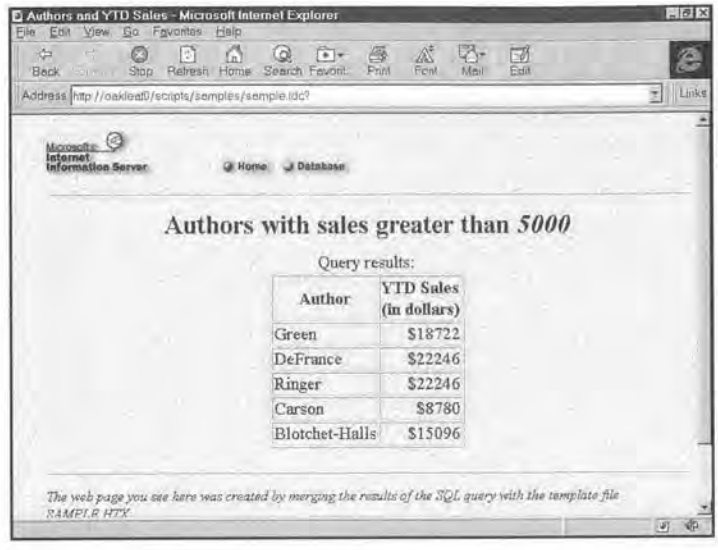


FIG. 16.2 Internet Explorer 3.0 displaying a Web page created from a query processed by the Internet Database Connector.

- *Index Server 1.1* performs full-text searches and retrieves information from any Web browser. Index Server is a free, downloadable service for Windows NT Server 4.0 and Internet Information Server 1+. Index Server 1.1 supports HTML, text, and Microsoft Office document formats, so you need not convert documents to HTML for indexing. Index Server 1.1 uses a special database structure for locating documents stored on one or more Windows NT servers. The Search feature of the Microsoft Web site uses Index Server 1.1.


On the Web

You can download your own copy of Index Server 1.1 from:

http://www.microsoft.com/ntserver/search/step111dl.htm

- *Merchant Server* contains a set of Windows NT services designed for conducting secure electronic commerce on the Internet. Merchant Server lets retailers first build "electronic storefronts" and accept credit-card payment via Secure Electronic Transaction (SET) technology. Microsoft acquired the storefront technology from eShop, an early Internet shopping mall. Merchant Server uses SQL Server 6.5 as its database back-end. Microsoft is expected to charge for licensing Merchant Server.


On the Web

A variety of information on Merchant Server is located at:

http://www.microsoft.com/merchant/

Oracle Ex. 1009, p. 596
Oracle v. DeBellis
IPR2024-00221

- *Media Server*, code-named *Cougar* when this book was written, is an Internet/intranet server designed expressly for delivering streaming MPEG-1 video content using the ActiveMovie Streaming Format (ASF). Media Server is a downsized version of Microsoft's Tiger server designed to support full-scale interactive video-on-demand (IVOD) using MPEG-2 compression, the standard compression system for DVD, plus digital direct broadcast satellite (DBS) and Advanced TV (ATV, also known as HDTV) transmission. Bandwidth limitations dictate that users connect to Media Server over a LAN or a high-speed WAN.

- *Commercial Internet System* (code-named *Normandy*) *for Windows NT 4.0* provides the following components: Internet Chat Server, Internet News Server, Internet White Pages and Locator Server, Information Retrieval Server, Content Replication System, and Personalization System. Normandy is intended primarily for use by commercial ISPs and large corporations that host their own Web sites. In mid-1996, CompuServe Information Services, Inc. announced that it would use the Normandy platform to implement its new Internet-based online service. Normandy uses SQL Server 6.5 for many of its services. Some of the components of Normandy, such as the Chat Server, are available for free downloading from the Microsoft Web site.

  You can download a few components of Normandy from:

  
  On the Web

  http://www.microsoft.com/internet/normandy/

- *dbWeb 1.1a* is a free, downloadable application that creates fully-formatted Web pages from data stored in databases that offer ODBC (Open Database Connectivity) drivers, such as Access and SQL Server. dbWeb consists of a ISAPI (Internet Server API) service, which runs on Internet Information Server 1+, and a stand-alone dbWeb Administrator component for creating schemas that define the data-related content of Web pages.

  You can download dbWeb 1.1a from:

  
  On the Web

  http://www.microsoft.com/intdev/dbweb/

▶▶ See "Defining Open Database Connectivity," p. 917



- *FrontPage 97* is an easy-to-use Web page authoring and Web site management system that supports ActiveX technologies and Java components. FrontPage 97 includes a Database Connectivity Wizard that simplifies the design of Web pages that offer user-defined queries. Figure 16.3 shows the FrontPage Explorer's map of the Volcano Coffee Company sample Web site. Although FrontPage 97 is a member of Office 97, Microsoft sells it as a separate retail product. When this book was written, the beta version of FrontPage 97 was available for downloading from the Microsoft Web site.

Oracle Ex. 1009, p. 597
Oracle v. DeBellis
IPR2024-00221

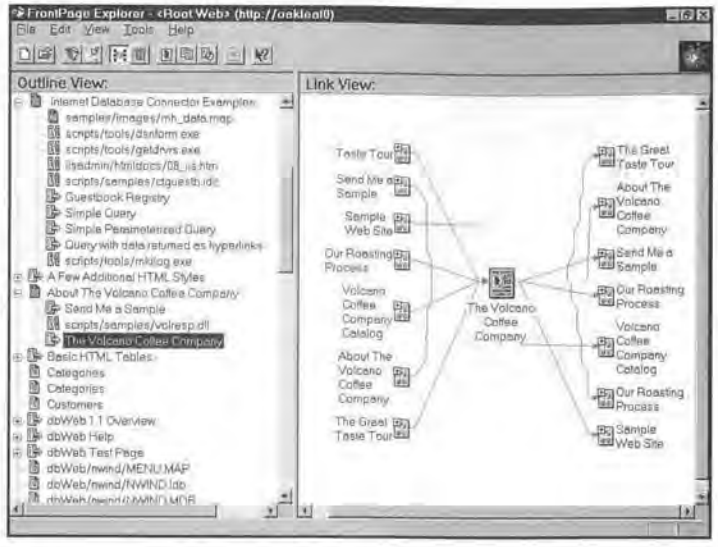


FIG. 16.3 The Volcano Coffee Company Web site displayed in the FrontPage Explorer application.


On the Web

Get the latest information on FrontPage 97 from:

http://www.microsoft.com/frontpage/

The preceding list of Internet-related products introduced in 1996 demonstrates that Microsoft is, to use Bill Gates' expression, "hard-core about the Internet." By mid-1996, the computer press simultaneously praised Microsoft for its "amazing Internet turnaround" and condemned the firm for its aggressive Internet-related marketing practices. Regardless of the outcome of complaints by its competitors to federal regulators, you can expect a continuing stream of new and upgraded Microsoft products for the Internet and intranets in 1997 and beyond.

Microsoft Technologies Supporting Internet-Related Products

Behind the new products listed in the preceding section are Microsoft technologies that make the products possible. The following list, current as of the fall of 1996, describes both the mature and the new specifications that provide the foundation for current and future Microsoft Internet products:

- *Common Object Model (COM)* is the foundation for OLE (Object Linking and Embedding) and OLE's lightweight successor for the Internet, ActiveX. COM is a specification for creating interactive software components, called *objects*, that become the building blocks of larger applications. In the fall of 1996, Microsoft appointed the Open Group, a computer software industry association, to manage the COM specification. Microsoft's objective in making COM an "open specification" is to gain support for COM as an industry-wide standard for defining programming objects.

Oracle Ex. 1009, p. 598
Oracle v. DeBellis
IPR2024-00221

▶▶ See "Introducing Object Building Blocks, p. 670"



You can read the 15 chapters of the COM specification at:



http://www.microsoft.com/oledev/olecom/title.htm

- *Distributed COM (DCOM)* implements COM over local- and wide-area networks, including the Internet. Microsoft first implemented the final version of DCOM (formerly *NetworkOLE*) in Windows NT 4.0. A beta version of DCOM for Windows 95 was available for downloading from Microsoft's Web site when this book was written. DCOM technology is critical to the implementation of Internet electronic commerce.

  The working draft of the DCOM specification is at:



  http://www.microsoft.com/oledev/olecom/dcomspec.txt

- *ActiveX* contains a collection of COM technologies intended by Microsoft to "activate the Internet" by enabling authors and graphics designers to create dynamic Web pages. The most important ActiveX components for Access 97 users are ActiveX Controls, a replacement for the OLE Controls of earlier versions of Access. IE 3.0 basically is a container for Microsoft's ActiveX Internet Control, which you can insert into an Access 97 form. OLE Documents, which first appeared in the Office 95 Binder, now are called *ActiveX Documents*.

  The following URL leads you to a wide range of information on ActiveX technologies:



  http://www.microsoft.com/ActiveX/

▶▶ See "Introducing Microsoft's ActiveX Technology," p. 1096
▶▶ See "Adding ActiveX Controls to Your Application," p. 1122



- *ActiveMovie* is the replacement for Microsoft's aging Video for Windows (VfW 1.1e for Windows 95 and Windows NT) that enables real-time playback of motion video (.avi files) under Windows. ActiveMovie 1.0 includes the ability to play MPEG-1 movies on a fast Pentium PC without the need for special hardware and eliminates the 2G .avi file size limit, but otherwise is uninspiring. Future versions of ActiveMovie are expected to support full-motion video capture and other features offered by VfW 1.1e.

  Current information on ActiveMovie is at:



  http://www.microsoft.com/imedia/activemovie/activem.htm

Oracle Ex. 1009, p. 599
Oracle v. DeBellis
IPR2024-00221

- *Visual Basic, Scripting Edition (VBScript, VBS)* is a simplified subset of Visual Basic, Applications Edition (VBA) 5.0, which is included as an application programming (macro) language with all of the members of Office 97, plus Visual Basic 5.0. You can incorporate VBS code in Web pages to program ActiveX controls. VBS also is used for server-side scripting in the ActiveX Server Framework. Once you learn VBA, it's relatively easy to write VBS code embedded in Web pages.


On the Web

The following URL provides links to pages that describe VBScript:

http://www.microsoft.com/vbscript/

- *Internet Services API (ISAPI)* contains a set of extensions to Internet Information Server 1+ that lets developers write Windows DLLs to replace slower-performing CGI (Common Gateway Interface) scripts. The most useful ISAPI extension for Access users is the Internet Database Connector (IDC), discussed in the preceding section.
- *Active Server Pages (ASP).* Formerly *ActiveX Server Framework* and code-named *Denali* when this book was written; an extension to IIS 2.0 that lets you include VBS on the server to automate the creation of custom Web pages based on input from Web browsers. ASP is included in the forthcoming Internet Information Server 3.0.
- *OLE DB* contains an SDK (Software Development Kit) and specification for simplifying access to tabular data stored in a variety of formats, such as ODBC-enabled databases (Access and SQL Server, for example), worksheets, e-mail, and text files. OLE DB enables the creation of *ActiveX Data Objects* (ADOs) which provide simplified data connectivity with VBS for Web pages distributed with the ASF. In its FAQs (Frequently Asked Questions) for OLE DB, Microsoft says, "We intend for ADO to be an evolution of our earlier object models. Over time, all our products will standardize on ADO and OLE DB."


On the Web

You can download version 1.0 of the OLE DB SDK and a preview edition of ADO from:

http://www.microsoft.com/oledb/

- *OLE Transactions* has a specification for handling updates to databases that involve more than one table. SQL Server 6.5's *Distributed Transaction Controller* (DTC) uses OLE Transactions to handle database updates across multiple databases, including databases residing on remote servers. Both OLE Transactions and DTC play an important role in Microsoft's newly-announced *Transaction Server* (MTx, code-named *Viper* when this book was written.)
- *DirectX* contains a series of hardware-related Windows 95 and Windows NT technologies for improving video (DirectDraw and Direct3D), audio (DirectSound, Direct3DSound, and DirectMusic), joystick (DirectInput) performance, and game playing over the Internet (DirectPlay).


On the Web

The details on DirectX are at:

http://www.microsoft.com/mediadev/

Oracle Ex. 1009, p. 600
Oracle v. DeBellis
IPR2024-00221

- *PC97*, a specification (*PC97 Hardware Design Guide 1.0*), defines three types of Windows-based personal computers: Basic, Workstation, and Entertainment PC. As of July 1, 1997, PCs must meet the requirements of one of the three PC97 types to qualify for use of Microsoft's "Designed for Windows 95" logo. The Entertainment PC includes basic specifications for large-screen PCs designed for family rooms, sometimes called *PCTVs* or *TVPCs*.

  The PC97 specification is available from:

  http://www.microsoft.com/hwdev/desguid/pc97des.htm



Note

The DirectX and PC97 items of the preceding list are included because they represent the hardware-enabling technology to make high-bandwidth multimedia Internet content, such as ActiveMovie/Media Server MPEG-1 video, practical for users with T-1 or faster connections to the Internet. Individual consumers of Internet multimedia must wait for cable, satellite, or xDSL (high-speed telephone line) access to view real-time, full-motion Internet video broadcasts.

Navigating with Hyperlinks

The key to the rapid adoption of the Internet's World Wide Web is the ability to jump quickly from one document to a related document or a particular page of a related document with *hyperlinks*. Apple Computer was one of the first firms to commercialize hyperlinks with its original HyperCard development environment, which used HyperTalk as a scripting language. Windows help files also use hyperlinks to display help topics, as do many CD-ROM titles created with CD-ROM authoring applications, such as Microsoft's MediaView and Macromedia's Authorware.

HyperCard, Windows help files, and CD-ROM authoring software allow the linked document files to be located on your PC's fixed disk drive or on a CD-ROM, or be accessible via a conventional LAN or WAN. The public Internet expanded the geographic scope of the hyperlinking process to the entire world. If you're reading a North American Web page on consumer electronics products, it's likely that many of the hyperlinks will point to the Web sites of Japanese manufacturers of TV sets, VCRs, and camcorders, such as **http://www.sony.co.jp**, which usually provide both English and Japanese versions of their text content.

Note

On the Web, *document* and *page* are synonyms; a Web page basically is a file, usually with an .htm or .html extension, stored in a folder on a Web server. *ActiveX documents* (formerly called *document objects* or *DocObjects*) created by Office 97 applications or Visual Basic 5.0 can be embedded within a Web page, which also may include other text and graphic elements.

Another feature of hyperlinking is the ability to return quickly to prior documents, then move on to another related document via a hyperlink. It's this forward-backward

Oracle Ex. 1009, p. 601
Oracle v. DeBellis
IPR2024-00221

navigation system that led to the terms Web *browsing* and *browser*. To provide improved performance, Web browsers store .htm or .html files, together with required graphics files (usually as .gif or .jpg files) on your local disk drive, a process called *caching*. The browser checks to see if a local copy of the file is present before requesting a download from the Web server.

Hyperlinks and File Paths

Hyperlinks can point to the default home page of a Web site, such as **http://www.microsoft.com**, or to the default page of an intranet server, as in **http://oakleaf0**, the IIS 2.0 intranet server used to write the Internet-related chapters of this book. (OAKLEAF0 is the name of the server on which IIS 2.0 is installed.) You don't need to specify the default page, which usually is Default.htm or Index.htm, as in **http://oakleaf0/default.htm**. You can jump directly to the default page for a specific topic by extending the address entry, for example **http://www.microsoft.com/msaccess** or **http://oakleaf0/samples/sampsite/default.htm** (refer to Figure 16.1).

Hyperlinks to Web pages use *relative paths* to specify the folders that store HTML pages, graphic images, or scripts that control the customization of pages. As an example, the default folder for the content OAKLEAF0 server is C:\InetPub\wwwroot; the Volcano Coffee Company home page is at C:\InetPub\wwwroot\samples\sampsite. The server name (**oakleaf0**) you include in the Uniform Resource Locator (URL) for the Volcano home page specifies the first part of the path (C:\InetPub\wwwroot) for an intranet server. Extended paths in the URL (**...\samples\sampsite**) are relative (added) to the default Web server path. Understanding the difference between fully-qualified paths (starting with a logical drive letter or *SERVERNAME*) and relative paths is very important to your use of hyperlinks with Access 97.

> **Note**
>
> The Internet was developed with and, until recently, dominated by UNIX hosts (servers). The UNIX operating system uses the virgule (forward slash, /) instead of the backslash as the separator in path statements. Another UNIX tradition is the use of lowercase letters for virtually everything, ranging from program names (grep and ftp) to Internet URLs (**http://www.microsoft.com**). UNIX ordinarily is case-sensitive, which would make InetPub a different folder from inetpub, but URLs and path extensions for Web sites are not case-sensitive.

Hyperlinks and Intranets

The combination of hyperlinks and browsing capability makes Web browsers effective competitors to custom database front-ends created with conventional database development platforms, such as Access and Visual Basic. All Web browsers share a relatively standard, simple interface with button-based navigation. Training costs are minimized because most users already are familiar with browsing the Web. Another advantage of browsers compared with custom database front-end applications is the promise of platform independence. Netscape Navigator runs under 16-bit and 32-bit Windows, the Macintosh operating system, and the more popular flavors of UNIX. Microsoft promises that updated versions of Internet Explorer for the Macintosh and UNIX will be available in 1997.

Oracle Ex. 1009, p. 602
Oracle v. DeBellis
IPR2024-00221

It remains to be seen if Web browsers will take over the role of forms-based database applications, such as Access 97 or Visual Basic 5.0, for delivering data via private networks. The primary incentive for establishing intranets is quick and easy deployment of new and updated Web pages that take the place of Access forms. As an example, you can duplicate the function of the Main Switchboard form of Northwind.mdb with a home page that replaces the form's command buttons with hyperlinks leading to Web pages that imitate the Categories, Products, Suppliers, and Orders forms. If you want to change the Categories page, you need only copy the page from the Web server, edit and test the page, then replace the page on the server. To add a new page, you need only place another hyperlink on the Main Switchboard page. The updating process is considerably easier than altering Northwind.mdb and delivering the new version to every user.

Understanding Access 97's Hyperlink Field Data Type

Access 97 adds a new member, *Hyperlink*, to its repertoire of field data types. Hyperlink is not a Jet 3.5 data type, because Jet 3.5 stores Access Hyperlink values in Memo fields. Access 97's Hyperlink data type simply interprets the value of the Memo field as a hyperlink and formats the display accordingly.

A Hyperlink field value may consist of the following three components:

- *Display text*, an optional descriptive name for the hyperlink that appears as emphasized text, usually underlined and having a distinctive color.
- *Address*, a required reference to the location of a related document. The reference can be an Internet URL for World Wide Web and FTP (File Transfer Protocol) sites, a relative or fully-qualified path and file name, or a file on a network server specified by UNC (Uniform Naming Convention).
- *Subaddress*, an optional reference to a named location in a related document, such as a bookmark in a Word or HTML document, or a named range in an Excel worksheet.

Individual components of Hyperlink values are separated by the pound sign, the same delimiter used for Jet date values. Following are examples of typical Hyperlink values:

- `Microsoft Web Site#http://www.microsoft.com` displays Microsoft Web site and jumps to the home page of **www.microsoft.com**.
- `Yen Conversion#c:\currency\1996dec.xls#Yen` displays Yen Conversion and points to the c:\currency\1996dec.xls file's Yen named range.
- `#\\Server1\Documents\Reports\1996sales.doc#Drugs` displays and points to the Drugs bookmark in the 1996sales.doc Word document in the shared \Documents \Reports folder of Server1. If you don't include a display text value, the address and subaddress values are displayed.
- `##frmProducts` points to the frmProducts form in the currently-open database.

Oracle Ex. 1009, p. 603
Oracle v. DeBellis
IPR2024-00221

Testing Hyperlinks in the Northwind Orders Table

Northwind.mdb's Suppliers table, shown in Figure 16.4, includes a Home Page Hyperlink field that includes links to Web pages located in your \Program Files\Microsoft Office\Office\Samples folder and to placeholder pages at **http://www.microsoft. com/accessdev/sampleapps**. Hyperlinks stored in tables are useful for creating searchable databases for storing links to your favorite Web sites. Hyperlinks in tables also offer automatic navigation assistance within large corporate intranets. As an example, you can create a table with document names, keywords, and other identifying information, then use a query to locate and display only those documents meeting criteria you establish.

The CAJUN.HTM and FORMAGGI.HTM hyperlinks point to individual Cajun.htm and Formaggi.htm files, respectively, using relative paths. Clicking either of these two hyperlinks opens Internet Explorer 3.0 (or another browser you've specified as your default browser) to display the file. Figure 16.5 shows Cajun.htm open in Internet Explorer 3.0. If you click one of the links with (on the World Wide Web) in the display text, your browser connects to the Web site and opens the specified page.



FIG. 16.4 The Home Page Hyperlink field of Northwind.mdb's Suppliers table.

> **Note**
>
> When you open IE 3.0 from an Access 97 hyperlink, Access automatically minimizes. Each hyperlink you click opens another independent instance of IE 3.0 to display the page.

Editing and Inserting Conventional Hyperlinks

To edit an existing hyperlink value or to add a new hyperlink to a table, right-click a cell in a Hyperlink field to open the popup menu, then choose Hyperlink, Edit Hyperlink (see Figure 16.6) to open the Edit Hyperlink dialog. Figure 16.7 shows the URL entry for the Plutzer home page located on the Microsoft Web site. If you're creating a new hyperlink, type the Internet URL of the page in the Link to File or URL drop-down

Oracle Ex. 1009, p. 604
Oracle v. DeBellis
IPR2024-00221

combo box. Alternatively, you can open the drop-down list and select from prior URLs that you've opened in IE 3.0. To link to a file, click the Browse button to open the Link to File dialog and select the file for the link.

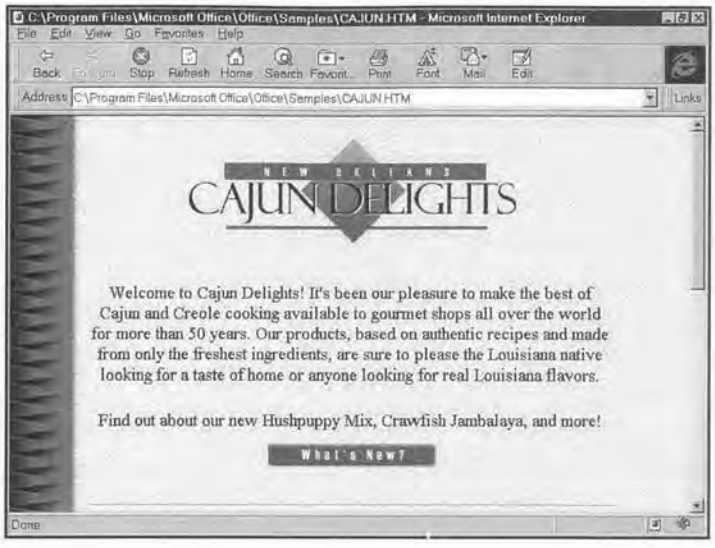


FIG. 16.5 The home page for the Cajun Delights (Cajun.htm) hyperlink of the Suppliers table opened in Internet Explorer 3.0.



FIG. 16.6 The popup menu and submenu for editing or creating a hyperlink value.

Note

You can edit the display text element of the hyperlink value in the text box of the popup submenu that opens when you choose Hyperlink from the first popup menu (refer to Figure 16.6).

Oracle Ex. 1009, p. 605
Oracle v. DeBellis
IPR2024-00221

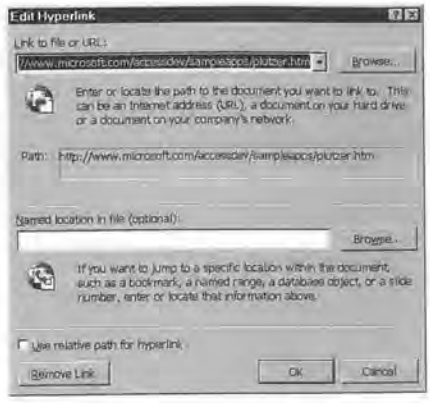


FIG. 16.7 The Edit Hyperlink dialog with an entry for a hyperlink to a page on the World Wide Web.

Linking to Bookmarks in a Word Document

To create a link to an existing Word document having one or more bookmarks that uses a hyperlink subaddress to locate a specific bookmark in the document, follow these steps:

1. Create a new table with at least one Text field to identify the subject matter and a Hyperlink field to contain the links to the bookmarks. Open the new table in Datasheet view.
2. Type the description in the text field and move to the Hyperlink field, but don't enter display text.

> **Note**
>
> If you enter display text in the Hyperlink field, an **http://** URL prefix is added to the display text to create a default URL in the Edit Hyperlink dialog's Link to URL or File combo box. This is not helpful when you want to link to a file.

3. Right-click the empty cell in the Hyperlink field, then choose Hyperlink, Edit Hyperlink to open the Edit Hyperlink dialog.
4. Click the Browse button to the right of the Link to URL or File combo box to open the Link to File dialog (see Figure 16.8). Select the .doc file to link and click the OK button to close the dialog. The fully-qualified path to the selected file appears in the Link to URL or File combo box (see Figure 16.9).
5. Type the name of the bookmark in the Named Location in File text box. If you want to use the relative path to the file, mark the Use Relative Path for Hyperlink check box (see Figure 16.10). A relative path is most useful when your document is in the same folder as the Access .mdb file; in this case only the file name appears in the Path label. Click OK to close the Edit Hyperlink dialog.

Oracle Ex. 1009, p. 606
Oracle v. DeBellis
IPR2024-00221

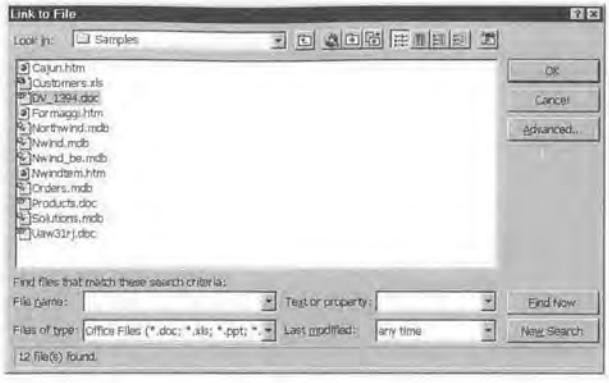


FIG. 16.8 Specifying the file for a hyperlink in the File to Link dialog.



FIG. 16.9 The full path to the file displayed in the URL or File to Link combo box.



FIG. 16.10 Specifying a bookmark for the hyperlink in the Word .doc file and the use of a relative path for the link to the file.

Oracle Ex. 1009, p. 607
Oracle v. DeBellis
IPR2024-00221

> **Note**
>
> You can't use the Browse button to the right of Named Location in File text box to display a list of bookmarks in Word .doc files or named ranges in Excel worksheets. The Browse button only displays objects in the currently open Jet database.

6. The file name (address element of the hyperlink value) appears in the Hyperlink cell of the table (see Figure 16.11). If you want display text to appear in the cell, rather than the file name, right-click the cell, choose Hyperlink, and type the entry in the Display Text text box of the popup submenu (refer to Figure 16.6).

> **Note**
>
> If you type over the file name in the current cell of the Hyperlink field that displays the file name, you replace the address element. Thus you must make the change in the popup submenu's Display Text text box.

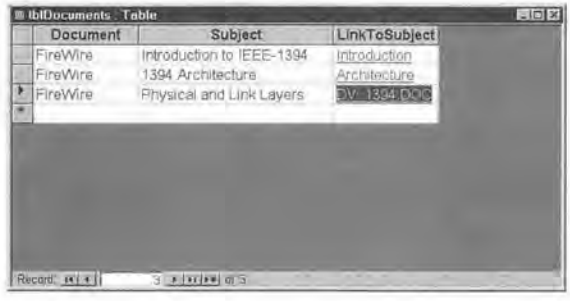


FIG. 16.11 A hyperlink entry without display text showing the address of the file (Dv_1394.doc).

7. When you click the new hyperlink entry, Word 97 opens the specified .doc file and positions the caret to the bookmark name you entered in step 5 (see Figure 16.12).



8. Click the Back button of Word's Web toolbar to return to Access.



9. Click the Forward button of Access's Web toolbar to return to Word and the open document.

10. Close Word when you're finished experimenting.

The preceding procedure also applies to Excel worksheets and PowerPoint presentations. Specify the name of an Excel 97 range or the number of a PowerPoint slide in the Named Location in File text.

Specifying Hyperlinks to Pages on an Intranet Server

You can establish links to pages on an intranet server using the following generalized URL format:

http://*servername/folder*[/*pagename*.htm]

Oracle Ex. 1009, p. 608
Oracle v. DeBellis
IPR2024-00221

As an example, to specify a link to the default page of the FireWire subfolder of the \InetPub\wwwroot folder for Internet Information Server 2.0 running on a server named OAKLEAF0, type **http://oakleaf0/firewire** in the Link to URL or File text box. Alternatively, you can specify a particular page in the folder, such as **.../1394wire.htm** (see Figure 16.13). To define a jump to an *anchor* (the term for the HTML equivalent of a bookmark) in the Web page, type the anchor name in the Named Location in File (subaddress) text box. Figure 16.17 shows IE 3.0 displaying the 1394wire.htm file, which was created by using Word 97 to save the Dv_1394.doc file in HTML format.

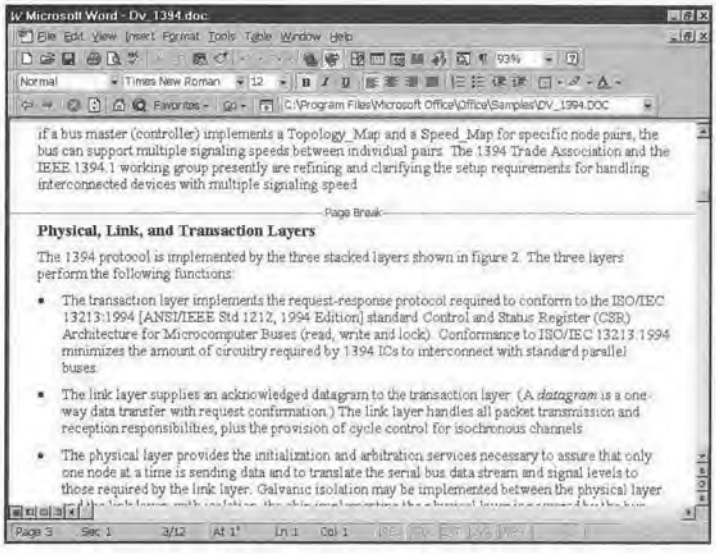


FIG. 16.12 The hyperlinked document opened in Word 97 with the caret positioned at the specified bookmark.



FIG. 16.13 Specifying a hyperlink jump to a page on an intranet server.

Oracle Ex. 1009, p. 609
Oracle v. DeBellis
IPR2024-00221

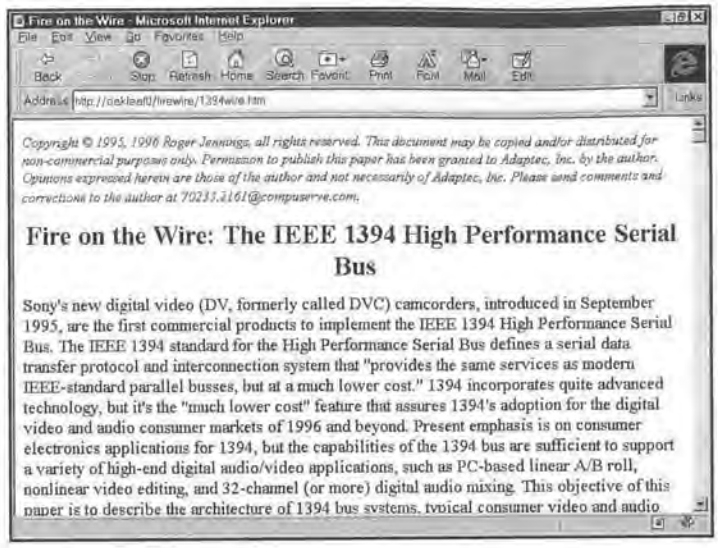


FIG. 16.14 A Web page displayed by making the jump to a document stored on an intranet server.

Using Hyperlinks to Open an Access Object

In addition to using hyperlinks to open documents in IE 3.0, Word 97, Excel 97, or PowerPoint 97, you can specify an object to open in an Access 97 database file. As an example, you can open a designated form in the Solutions.mdb database by following these steps:

1. Open the Suppliers table of Northwind.mdb and move to an empty cell in the Home Page field.
2. Right-click the empty Hyperlink cell and choose Hyperlink, Edit Hyperlink to open the Edit Hyperlink dialog.
3. Click the upper Browse button, select Solutions.mdb, and click OK to specify the Solutions database as the address of the hyperlink (see Figure 16.15).
4. Click the lower Browse button to open the Select Location dialog, a modal version of the Database window that displays the objects in the selected .mdb file.
5. Click the Forms tab and select one of the forms in Solutions.mdb, such as the SolutionsHyperForm (see Figure 16.16). Click OK to close the Select Location dialog.
6. The type of object, Form in this case, followed by a space and the name of the object appears in the Named Location in File text box (see Figure 16.17). Click OK to close the Edit Hyperlink dialog.

Oracle Ex. 1009, p. 610
Oracle v. DeBellis
IPR2024-00221

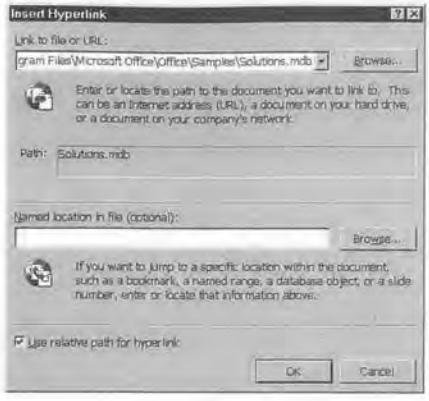


FIG. 16.15 An Access .mdb file specified as the address of a hyperlink.



FIG. 16.16 Selecting the Access object to open in a hyperlinked .mdb file.



FIG. 16.17 Selecting the Access object to open in a hyperlinked .mdb file.

7. Click the new hyperlink. A second instance of Access launches, opens Solutions.mdb, and then opens the SolutionsHyperForm form (see Figure 16.18). You might need to close the Solutions splash screen to make the form visible.

Oracle Ex. 1009, p. 611
Oracle v. DeBellis
IPR2024-00221

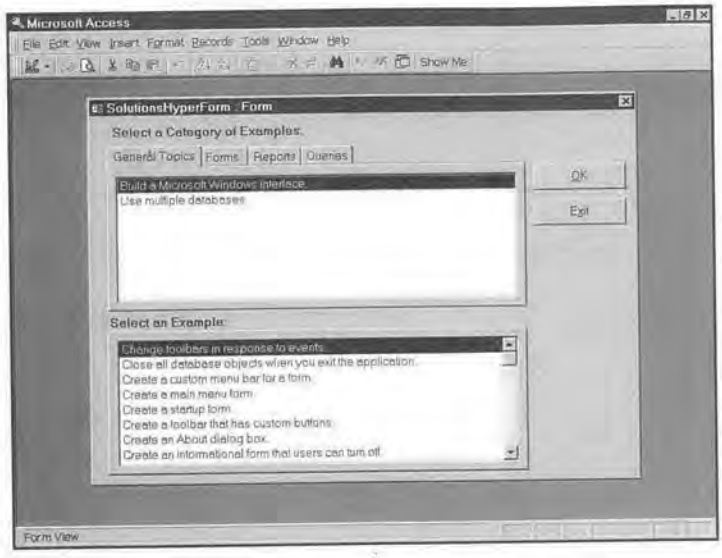


FIG. 16.18 The SolutionsHyperForm form opened by the hyperlink of Figure 16.17.

8. Click the Exit button to close Solutions.mdb, close the second instance of Access, and clear the test hyperlink value you created.

Note

Access 97 uses Windows dynamic data exchange (DDE) to process Hyperlinks to applications that reside on your PC. If you're familiar with DDE, following is the correspondence of hyperlink value components with DDE command arguments.

The address component of the hyperlink value corresponds to DDE's `TopicName` argument of the `DDEInitiate` command; the `ServiceName` argument is derived from the file extension. The subaddress component corresponds to the `ItemName` argument of the `DDERequest` command. If you're not familiar with DDE, disregard this note.

Specifying Internet Uniform Resource Locators

Internet Uniform Resource Locators consist of an abbreviation for a protocol that defines the type of object (usually a document) and the location (address) of the desired object on the Internet or an intranet. In most cases, the protocol and the address are separated by two virgules (forward slashes) and a colon, as in **http://www.microsoft.com** or **ftp://ftp.microsoft.com**. Access 97 hyperlinks support the http (HyperText Transport Protocol) and FTP URL prefixes with IE 3.0 and other Web browsers. Access 97's Hyperlink fields recognize the URL prefixes listed in Table 16.1. You need the appropriate software installed and registered on your PC to use prefixes other than http and ftp.

Oracle Ex. 1009, p. 612
Oracle v. DeBellis
IPR2024-00221

Table 16.1 Uniform Resource Locator (URL) Prefixes Recognized by Access 97 Hyperlink Fields

URL Prefix	Protocol Name	Description of Protocol
cid	CompuServe Information Dialer (CID)	Attempt to create a connection using Point-to-Point Protocol (PPP) to the Internet using the CompuServe dialer.
file	File	Opens a file specified by a fully-qualified path and file name.
ftp	File Transfer	Transfers files stored on an FTP server.
gopher	Gopher	Obtains information from a Gopher server.
http	Hypertext Transfer	Opens a Web page.
https	Hypertext Transfer with Security	Attempts to open a connection to a Web page with Secure Sockets Layer (SSL) encryption.
mailto	Electronic mail	Sends a message using your preferred e-mail program using the mailto:mailbox@domainname URL format.
mid	Musical Instrument Digital Interface (MIDI)	Uses your audio adapter card to play MID files.
mms	Microsoft Media Server (MMS)	Plays streaming media files (such as ASF files).
msn	Microsoft Network	Jumps to a location on The Microsoft Network.
news	News protocol	Starts your news reader program for the specified news group using the news: newsgroupname URL format.
nntp	Network News Transfer	Same as news, but uses the conventional nntp://newsgroupname URL format.
pnm	RealAudio protocol	Uses your audio adapter card to play Progressive Networks' RealAudio files.
prospero	Prospero	Accesses the Prospero distributed file system for UNIX hosts.
rlogin	Terminal login	Tries to start a Rlogin terminal emulation program on a UNIX host.
telnet	Terminal emulation	Tries to start a telnet terminal emulation program with a UNIX host.
tn3270	IBM 3270 terminal emulation	Tries to start an IBM 3270 terminal emulation program.
wais	Wide Area Information Servers	Connects to a WAIS search services database.

Using Hyperlinks with Access Controls

You can use hyperlinks with Access controls, such as command buttons, combo boxes, and labels, to jump to another document or open an object within the current or another Access database. To create a form with a command button to open Northwind.mdb's Orders form with a hyperlink, follow these steps:

1. Create a new form in Design view.
2. Disable the Control Wizards and add a command button to the form.

Oracle Ex. 1009, p. 613
Oracle v. DeBellis
IPR2024-00221



3. With the command button selected, click the Properties button of the toolbar to open the Properties window and click the Format tab.
4. In the Caption property text box, type **Open Orders Form** to create the display text component of the hyperlink.
5. Leave the Hyperlink Address property text box empty, specifying that the object you want to open is in the current database.



6. Select the Hyperlink Subaddress property text box and click the Builder button to open the Select Location dialog.
7. Click the Forms tab, select the Orders form, then click OK to close the dialog. `Form Orders` appears in the Hyperlink Subaddress property text box, and the command button's caption appears in blue type with an underscore (see Figure 16.19).

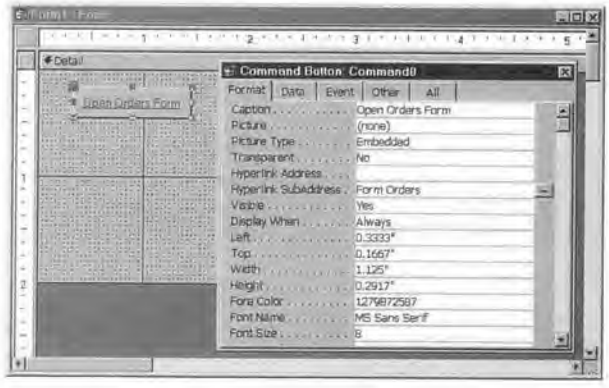


FIG. 16.19 A command button that uses a hyperlink to open a form in the current database.

8. Change to Form view and click the command button to open the Orders form.
9. Close, but don't save, your test form.

Note

Using command buttons with hyperlinks to open forms lets you take advantage of Access 97's *lightweight forms* (forms without VBA code in class modules). You can open, but not close, a form with a hyperlink. As a result, users of your application may be confronted a large number of simultaneously-open forms. Having more than two open forms at a time is likely to confuse users new to Access applications.

Oracle Ex. 1009, p. 614
Oracle v. DeBellis
IPR2024-00221

Chapter 17

Exporting Data to World Wide Web Pages

The easiest approach to creating Web pages is to export existing content from applications with which you're familiar. Each member of Office 97, except Outlook, is capable of exporting its documents to formatted HTML files with a File menu choice or a wizard. Access 97 gives you several options for creating static and dynamic Web pages from data contained in Jet tables. This chapter covers methods for generating static Web pages, which you must replace when the underlying data changes. Chapter 18, "Creating Dynamic Web Pages," describes how to create Web pages that automatically display the most current information.

You don't need to be an HTML (HyperText Markup Language) expert to export data from Access 97 objects to static Web pages. In fact, you don't need to have any knowledge of HTML and its formatting tags to perform the examples in this chapter. An elementary understanding of use of basic HTML tags, however, lets you alter the predetermined format of exported data to improve the appearance or utility of your Web pages. Thus, the HTML source for some of the examples in this chapter is included in the form of code listings or examples in the text.

Exporting Table and Query Datasheets to HTML

Northwind.mdb's Suppliers table is a good choice for exporting to a Web page because it contains relatively few fields and records. The Suppliers table also includes four hyperlinks, two of which link to sample home pages in your ...\Office\Samples folder. To create a formatted Web page from the Suppliers table, follow these steps:

1. Open the Suppliers table of Northwind.mdb in Datasheet view.
2. Choose File, Save As/Export to open the Save As dialog. With the To an External File or Database option selected, click the OK button to close the dialog and open the Save Table 'Suppliers' In dialog.

Oracle Ex. 1009, p. 615
Oracle v. DeBellis
IPR2024-00221

3. In the Files of Type drop-down list, select HTML Documents (*.html, *.htm), as shown in Figure 17.1. The .html extension is used most often by UNIX servers, while .htm is most common for Windows NT 4.0 servers. Internet Explorer (IE) 3.0 handles .html (the default) and .htm extensions equally well.



FIG. 17.1 Choosing HTML Documents as the format for exporting the content of an Access 97 table.

4. Selecting HTML Documents enables the Save Formatted check box, which you must mark to include Access 97's automatic HTML formatting. When you mark the Save Formatted check box, the Autostart check box is enabled. Mark the Autostart check box to have IE 3.0 or another default browser display the page when exporting completes (see Figure 17.2).



FIG. 17.2 Specifying HTML formatting with the Save Formatted option and opening the exported .html file in your default browser with the Autostart option.

5. Click the Export button to close the Save Table 'Suppliers' In dialog and begin the export process. Click OK to close the HTML Output Options dialog, leaving the

Oracle Ex. 1009, p. 616
Oracle v. DeBellis
IPR2024-00221

HTML Template text box empty (see Figure 17.3). The section "Using HTML Templates" later in the chapter describes how to base the design of your Web page on an HTML template.



FIG. 17.3 Bypassing use of an existing HTML template file by leaving the text box of the HTML Output Options dialog empty.

6. When the formatted Suppliers.html Web page automatically appears in your browser (see Figure 17.4), scroll to the right to display the Home Page table column (see Figure 17.5). Click one of the local hyperlinks (CAJUN.HTM or FORMAGGI.HTM) to test the links to the sample Web pages in your ...\Office\Samples folder. HTML formats hyperlinks with `<A HREF=location>` tags that include the location of the linked document.

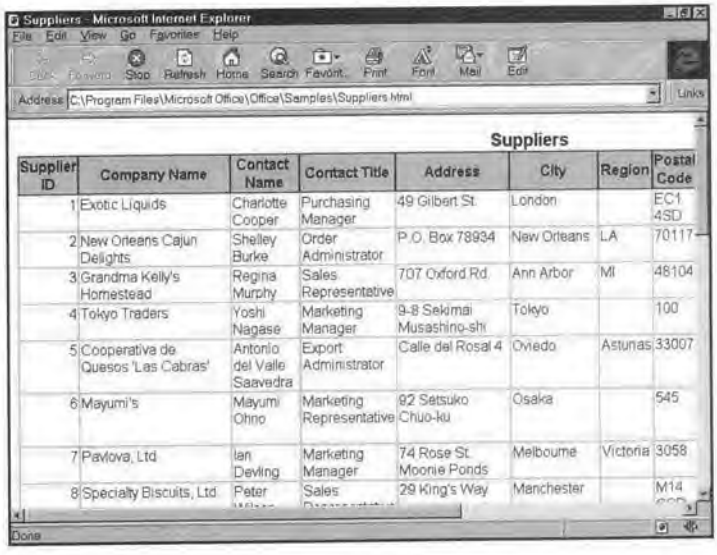


FIG. 17.4 The formatted Web page created from the Suppliers table.

Note

HTML uses *tags* to identify the beginning and end of HTML documents (`<HTML>` and `</HTML>`), titles (`<TITLE>` and `</TITLE>`), and other elements of the page. HTML 3.2, the current version of HTML when Microsoft released Access 97, has hundreds of tags for formatting, creating control objects on pages, and other uses.

Oracle Ex. 1009, p. 617
Oracle v. DeBellis
IPR2024-00221

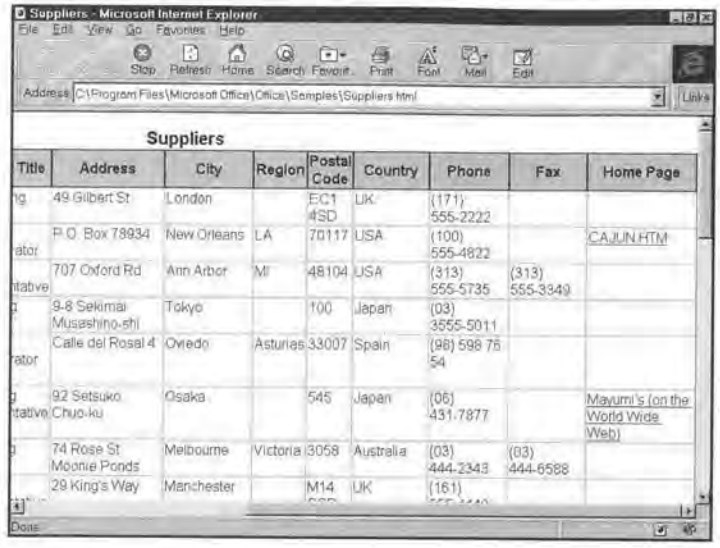


FIG. 17.5 Hyperlinks to local (CAJUN.HTM) and Internet (Mayumi's) Web pages in the Suppliers.html file.

On the Web

The World Wide Web Consortium (W3C) is responsible for maintaining the standards for HTML. You can obtain the latest information on HTML 3.2 and its successors from

http://www.w3.org/pub/WWW/MarkUp/Wilbur/

7. Choose View, Source from IE 3.0's menu to display the HTML source for the Suppliers.html file in Windows Notepad (see Figure 17.6). If the size of the file is larger than about 50K, you receive a message that the file is too large for Notepad and are given the option to open the file in WordPad.

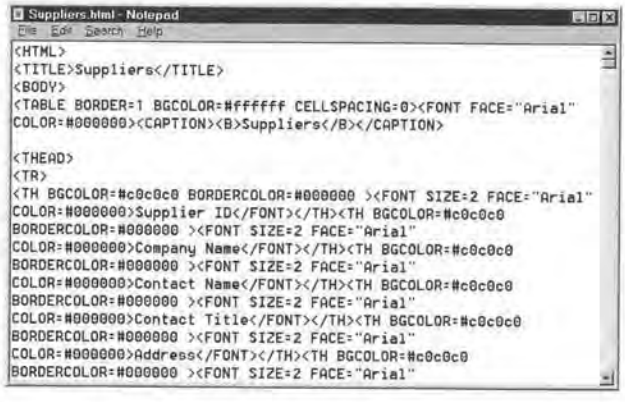


FIG. 17.6 The first few lines of the HTML source for the Suppliers.html file.

8. Close Notepad and your browser, then return to Access.

Oracle Ex. 1009, p. 618
Oracle v. DeBellis
IPR2024-00221

Creating an Unformatted Web Page

The vast majority of the content of the Suppliers.html file is HTML formatting instructions for colors and text fonts. The color and font formatting instructions obscure the HTML text that creates the tabular structure of the Web page. Much of the formatting content is duplicated throughout the source code; duplicate formatting is typical when you use tools that automatically create Web pages for you.

To create a Web page with a simple HTML table that doesn't include extra formatting instructions, modify the process described in the preceding example as follows:

1. Repeat steps 2 and 3 of the preceding example. In step 3, change the name of the file to **Suppliers (Unformatted).html**.
2. With the Save Formatted check box cleared, click the Export button to create the unformatted Web page. In this case, the HTML Output Options dialog doesn't appear.
3. Open Explorer, navigate to your ...\Office\Sample folder, and double-click the Suppliers (Unformatted).html item to launch your browser and display the unformatted Web page (see Figure 17.7). To improve the readability of Web pages in the figures of this book, the background color of IE 3.0 is set to white.



Suppliers (Unformatted)

1	Exotic Liquids	Charlotte Cooper	Purchasing Manager	49 Gilbert St.	London	
2	New Orleans Cajun Delights	Shelley Burke	Order Administrator	P.O. Box 78934	New Orleans	LA
3	Grandma Kelly's Homestead	Regina Murphy	Sales Representative	707 Oxford Rd.	Ann Arbor	MI
4	Tokyo Traders	Yoshi Nagase	Marketing Manager	9-8 Sekimai Musashino-shi	Tokyo	
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Export Administrator	Calle del Rosal 4	Oviedo	Asturias
6	Mayumi's	Mayumi Ohno	Marketing Representative	92 Setsuko Chuo-ku	Osaka	
7	Pavlova, Ltd.	Ian Devling	Marketing Manager	74 Rose St. Moonie Ponds	Melbourne	Victoria
		Peter	Sales			

FIG. 17.7 The Suppliers (Unformatted).html file displayed in Internet Explorer 3.0.

4. Choose View, Source from IE 3.0's menu to display in Notepad the simplified HTML source for the page (see Figure 17.8). All of the font and color formatting is removed, along with the source only the most basic HTML tags use to create a title for the browser's title bar, caption, and the table containing the data. The unformatted table doesn't include a header for field names and uses the default HTML table borders.

Oracle Ex. 1009, p. 619
Oracle v. DeBellis
IPR2024-00221

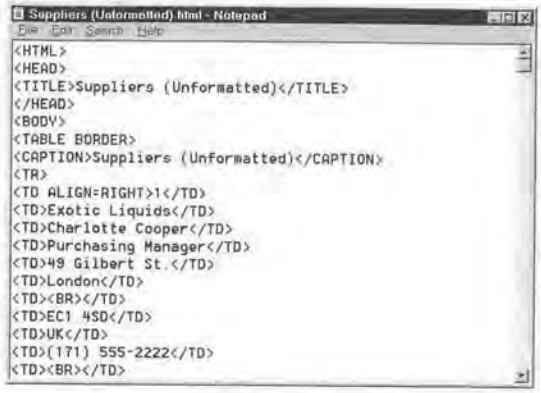


FIG. 17.8 Simplified HTML source for the unformatted version of the Page created from the Suppliers table.

5. After you've reviewed the HTML source code, close Notepad and your browser to return to Access.

Listing 17.1 shows the HTML source for the unformatted page, with only the first two and the last rows of data. (An ellipsis (. . .) replaces missing table data rows.) `<TR>. . . </TR>` tags define the beginning and end of a table row. Individual data cells of the table are defined by `<TD>. . . </TD>` pairs. Unless the `ALIGN=RIGHT` attribute for numeric values is applied to a cell, the default left alignment for text prevails.

Listing 17.1 HTML Source for the Suppliers (Unformatted).html File with Source Only for the First Two and Last Records

```
<HTML>
<HEAD>
<TITLE>Suppliers (Unformatted)</TITLE>
</HEAD>
<BODY>
<TABLE BORDER>
<CAPTION>Suppliers (Unformatted)</CAPTION>
<TR>
<TD ALIGN=RIGHT>1</TD>
<TD>Exotic Liquids</TD>
<TD>Charlotte Cooper</TD>
<TD>Purchasing Manager</TD>
<TD>49 Gilbert St.</TD>
<TD>London</TD>
<TD><BR></TD>
<TD>EC1 4SD</TD>
<TD>UK</TD>
<TD>(171) 555-2222</TD>
<TD><BR></TD>
<TD><BR></TD>
</TR>
<TR>
<TD ALIGN=RIGHT>2</TD>
<TD>New Orleans Cajun Delights</TD>
```


Oracle Ex. 1009, p. 620
Oracle v. DeBellis
IPR2024-00221

```
<TD>Shelley Burke</TD>
<TD>Order Administrator</TD>
<TD>P.O. Box 78934</TD>
<TD>New Orleans</TD>
<TD>LA</TD>
<TD>70117</TD>
<TD>USA</TD>
<TD>(100) 555-4822</TD>
<TD><BR></TD>
<TD><A HREF="CAJUN.HTM">CAJUN.HTM</A></TD>
</TR>
...
<TR>
<TD ALIGN=RIGHT>29</TD>
<TD>For&ecirc;ts d'&eacute;rables</TD>
<TD>Chantal Goulet</TD>
<TD>Accounting Manager</TD>
<TD>148 rue Chasseur</TD>
<TD>Ste-Hyacinthe</TD>
<TD>Qu&eacute;bec</TD>
<TD>J2S 7S8</TD>
<TD>Canada</TD>
<TD>(514) 555-2955</TD>
<TD>(514) 555-2921</TD>
<TD><BR></TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

HTML uses the `&charname;` format to specify special characters. As an example, `ê` inserts the letter "e" with a circumflex (ê) for Forêts, and `&eacute` inserts the letter "e" with an acute (é) in both `d'érables` and `Québec` in the last row of the table in Listing 17.1. `</TABLE>`, `</BODY>`, and `</HTML>` denote the end of the table, body part, and document.

Note

Files containing conventional (unformatted) HTML tables are much smaller than Access 97's formatted files. As an example, Suppliers.html (33K) is almost four times the size of Suppliers (Unformatted).html (9K). It's uncommon to export tables with large numbers of records to single Web pages because they are slow to load in the user's browser and make finding the desired row a chore. If you must create a static Web page from a table or query with a large number of rows, choose the unformatted version to reduce file size and speed display in the browser.

Creating a Web Page from a Query

Tables in Web pages exported from entire Access tables often contain much more than users want to know. Thus, most static Web pages include only a subset of the records and columns of large tables. The objective of reducing the number of columns is to eliminate the need for horizontal scrolling to review the information presented. Queries let you specify the columns that appear in the page. Multiple queries with different criteria let you create a series of Web pages opened by hyperlinks on a home page.

Oracle Ex. 1009, p. 621
Oracle v. DeBellis
IPR2024-00221

> **Note**
>
> If you use a parameterized query, you are prompted to enter the parameter value before Access creates the HTML file. If you use parameterized queries to create multiple pages, remember to save the resulting file with an appropriate file name to prevent overwriting files created with other parameter values. All of the Web pages you create from a single parameterized query have the same title and caption, which limits the utility of parameterized queries for creating static Web pages.

The following example uses a query to display only the North American customers of Northwind Traders in a format that doesn't require horizontal scrolling. The resulting Web page almost fills the width of the display area of a browser on PCs using either 640×480 or 800×600 video resolution. Although 800×600 is the most common resolution for today's desktop PCs with 15-inch and larger monitors, you also should make sure your Web page design is suitable for the 640×480 resolution used by the majority of the installed base of laptop computers.

Follow these steps to create the sample query and Web page:

Queries

1. Create a new query and add only the Customers table.
2. Add the CustomerID, CompanyName, City, Region, PostalCode, Phone, and Country fields to the query design grid.
3. Add **USA**, **Canada**, and **Mexico** in the Criteria cells of the Country field and save your query with the name **North American Customers** (see Figure 17.9).

> **Note**
>
> The export process uses the query object name for the title and caption of the Web page. Thus, using the `qry` prefix for query names isn't recommended when you design queries for export to Web pages.

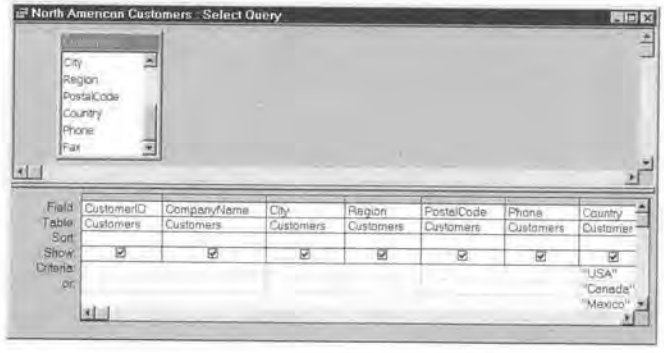


FIG. 17.9 The design of a query to display selected records in a Web page without a horizontal scroll bar.

Oracle Ex. 1009, p. 622
Oracle v. DeBellis
IPR2024-00221

4. Run your query to change to Datasheet view and test the design (see Figure 17.10).



North American Customers : Select Query

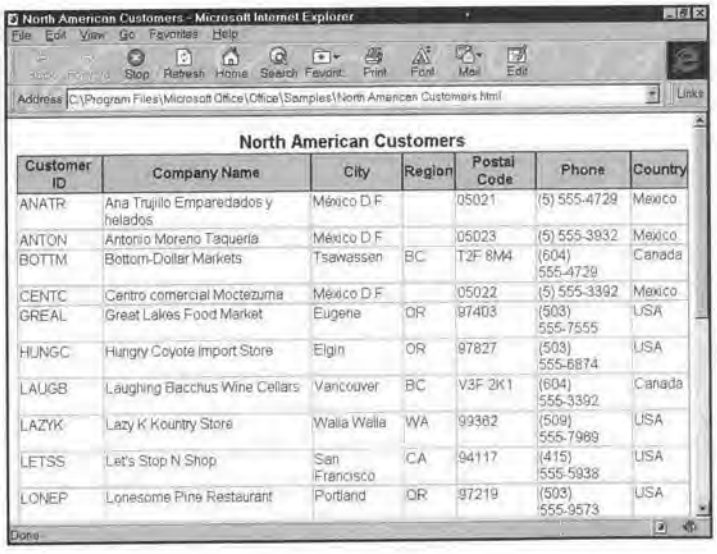
Customer ID	Company Name	City	Region	Postal
ANATR	Ana Trujillo Emparedados y helados	México D.F.		05021
ANTON	Antonio Moreno Taquería	México D.F.		05023
BOTTM	Bottom-Dollar Markets	Tsawassen	BC	T2F 8M4
CENTC	Centro comercial Moctezuma	México D.F.		05022
GREAL	Great Lakes Food Market	Eugene	OR	97403
HUNGC	Hungry Coyote Import Store	Elgin	OR	97827
LAUGB	Laughing Bacchus Wine Cellars	Vancouver	BC	V3F 2K
LAZYK	Lazy K Kountry Store	Walla Walla	WA	99362
LETSS	Let's Stop N Shop	San Francisco	CA	94117
LONEP	Lonesome Pine Restaurant	Portland	OR	97219
MEREP	Mère Paillarde	Montréal	Québec	H1J 1C
OLDWO	Old World Delicatessen	Anchorage	AK	99508
PERIC	Pericles Comidas clásicas	México D.F.		05033
RATTC	Rattlesnake Canyon Grocery	Albuquerque	NM	87110

Record: 1 of 21

FIG. 17.10 The query result set of the query design of Figure 17.9.

5. Choose File, Save As/Export, select the To an External File or Database option, and click OK to open the Save Query 'North American Customers' In dialog. Select HTML Files (*.html, *.htm) in the Files of Type drop-down list and mark the Save Formatted and Autostart check boxes. Change the name of the file from North American Customers.html in the File Name combo box, if you want to use a shorter file name.

6. Click Export, then click OK when the HTML Options dialog appears. After a few seconds, your query result set appears in your default Web browser. Figure 17.11 shows the formatted page in IE 3.0 at 800×600 resolution with large fonts specified. The appearance of the page is identical in 640×480 resolution.



North American Customers - Microsoft Internet Explorer

File Edit View Go Favorites Help

Stop Refresh Home Search Favorit Print Font Mail Edit

Address C:\Program Files\Microsoft Office\Office\Samples\North American Customers.html Links

North American Customers

Customer ID	Company Name	City	Region	Postal Code	Phone	Country
ANATR	Ana Trujillo Emparedados y helados	México D.F.		05021	(5) 555-4729	Mexico
ANTON	Antonio Moreno Taquería	México D.F.		05023	(5) 555-3932	Mexico
BOTTM	Bottom-Dollar Markets	Tsawassen	BC	T2F 8M4	(604) 555-4729	Canada
CENTC	Centro comercial Moctezuma	México D.F.		05022	(5) 555-3392	Mexico
GREAL	Great Lakes Food Market	Eugene	OR	97403	(503) 555-7555	USA
HUNGC	Hungry Coyote Import Store	Elgin	OR	97827	(503) 555-6874	USA
LAUGB	Laughing Bacchus Wine Cellars	Vancouver	BC	V3F 2K1	(604) 555-3392	Canada
LAZYK	Lazy K Kountry Store	Walla Walla	WA	99362	(509) 555-7969	USA
LETSS	Let's Stop N Shop	San Francisco	CA	94117	(415) 555-5938	USA
LONEP	Lonesome Pine Restaurant	Portland	OR	97219	(503) 555-9573	USA

Done

FIG. 17.11 The Web page created from the query datasheet of Figure 17.10.

Oracle Ex. 1009, p. 623
Oracle v. DeBellis
IPR2024-00221

Note

The Web browser is responsible for determining much of the formatting of table display. The browser attempts to display tables within the width of the display area when possible, making a best guess on the relative width of columns. There are a variety of HTML tags that you can add to the HTML source created by Access 97's export feature to format the tables. Refer to the HTML 2.0 and 3.2 specifications or Que's *Special Edition Using HTML* for further information on custom formatting of tables.

Using HTML Templates

Most commercial Web sites use HTML templates to provide a consistent corporate or organizational image and to add visual interest to Web pages without writing a lot of HTML source for each page. The majority of Web page authoring applications, such as Microsoft FrontPage, include a variety of templates from which you quickly can create Web pages with a standardized appearance.

Using the Access HTML Templates Included with Office 97

Microsoft includes a variety of HTML templates (.htm files) in your \Program Files\ Microsoft Office\Templates\Access folder. These templates each include a "Created with Microsoft Access" logo and a different background color and pattern. Figure 17.12 shows the 100.htm template displayed in IE 3.0. Other templates in the ...\Templates\Access folder include:

- Default.htm
- Gray.htm
- Grayst.htm
- Mc.htm
- Mcst.htm
- Sky.htm
- Stones.htm
- Tiles.htm
- Zigzag.htm

Background patterns are provided by .jpg graphics files with the same file name as the template. Background pattern files usually are small graphics files that the browser tiles to create a pattern over the entire page display region. The "Created with Microsoft Access" logo is provided by Msaccess.jpg.

When you choose View, Source from IE 3.0's menu to display the HTML source of an Access HTML template file, the content appears similar to that shown for 100.htm in Figure 17.13. The `<! Text >` tag normally is used to add invisible comments to a Web page. The Access 97 export feature interprets comment text in the format `<!--AccessTemplate_Element-->` to mean replace this line with the specified `Element`. Microsoft calls these comment lines *Access HTML Template Tags*. Table 17.1 lists the Access HTML Template Tags recognized by Access 97. The `...Page` anchor tags listed in Table 17.1 are used with multiple-page exports from Access reports, the subject of the section "Exporting Reports to HTML" later in the chapter.

Oracle Ex. 1009, p. 624
Oracle v. DeBellis
IPR2024-00221

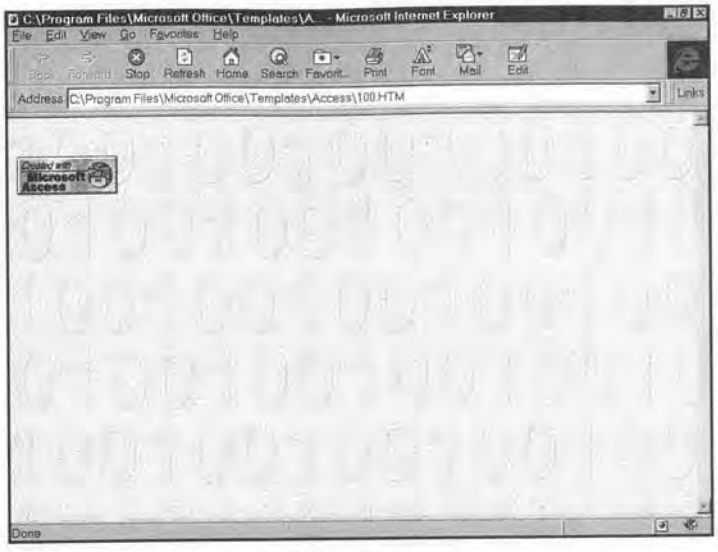


FIG. 17.12 The 100.htm template file displayed in Internet Explorer 3.0.

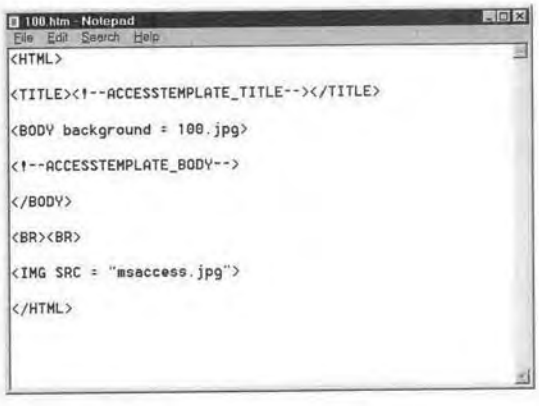


FIG. 17.13 The HTML source for the 100.htm template file.

Table 17.1 Replaceable Access HTML Template Tags Recognized by Access 97

Access HTML Template Tag	Purpose
`<!--AccessTemplate_Title-->`	The object name that appears in the browser's title bar.
`<!--AcessTemplate_Body-->`	The table created from the object's output.
`<!--AccessTemplate_FirstPage-->`	Anchors tag to first page.
`<!--AccessTemplate_PreviousPage-->`	Anchors tag to previous page.

(continues)

Oracle Ex. 1009, p. 625
Oracle v. DeBellis
IPR2024-00221

Table 17.1 Continued

Access HTML Template Tag	Purpose
`<!--AccessTemplate_NextPage-->`	Anchors tag to next page.
`<!--AccessTemplate_LastPage-->`	Anchors tag to last page.
`<!--AccessTemplate_PageNumber-->`	Displays the current page number.

Exporting a Query Datasheet with a Template

To add a template to a Web page created from a query datasheet, follow these steps:

1. Create the query design, save the query with an appropriate name for the title and caption, and execute the query to open it in Datasheet view. This example uses a modified version (US Customers) of the North American Customers query that displays only customers in the U.S.
2. Copy the graphics files required by the template to the folder in which you intend to save your exported Web page. For the 100.htm template, copy the 100.jpg and Msaccess.jpg files to ...\Office\Samples, assuming you are using this folder to store your .html files.
3. Choose File, Save As/Export, select the To an External File or Database option, and click OK to open the Save Query '*Query Name*' In dialog. Select HTML Files (*.html, *.htm) in the Files of Type drop-down list and mark the Save Formatted and Autostart check boxes. Change the name of the file in the File Name combo box, if you want to use a non-default file name. Click Export to continue.
4. When the HTML Options dialog appears, click the Browse button to select your template in the HTML Template To Use dialog—...\Templates\Access\100.htm for this example (see Figure 17.14).



FIG. 17.14 Specifying the Access HTML template for an exported Web page.

> **Note**
>
> When you specify a template in the HTML Options dialog, the template becomes the default template for the succeeding Web pages you export. The location and name of the template file appears in the HTML Template text box of the Hyperlinks/HTML page of the Options dialog. Choose Tools, Options, and click the Hyperlinks/HTML tab to display the HTML Template text box.

5. Click OK to export the query datasheet and open the Web page in your browser. Scroll to the bottom of the page to see the background and the "Created with

Oracle Ex. 1009, p. 626
Oracle v. DeBellis
IPR2024-00221

Microsoft Access" logo (see Figure 17.15). The background of the table and its caption is opaque, so the background texture of 100.jpg doesn't appear behind the table region.

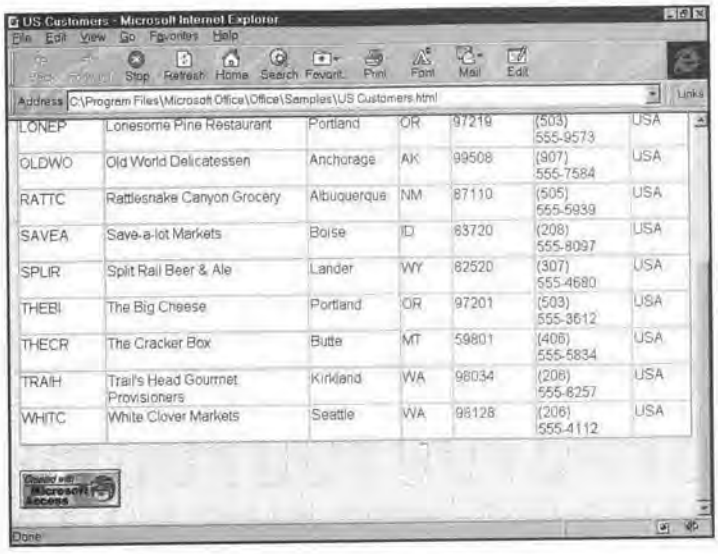


FIG. 17.15 The bottom of the Web page created with the 100.htm template that adds a background and the "Created with Microsoft Access" logo.

6. Choose View, Source from IE 3.0's menu to view in Notepad the HTML lines added by the 100.htm template. Figure 17.16 shows the `<BODY background = 100.jpg>` line added from the template. Scrolling to the end of the file displays the `<IMG SRC = "msaccess.jpg">` tag for the logo.

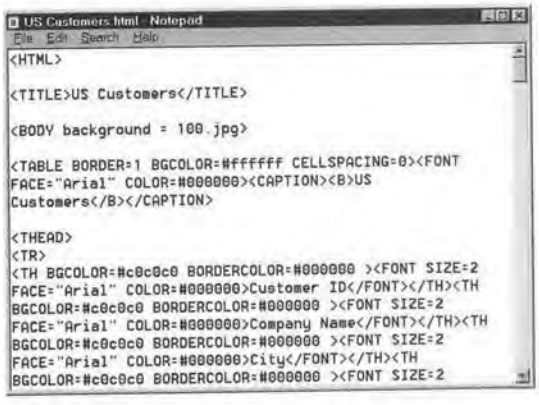


FIG. 17.16 The first few lines of the HTML source for a Web page created with the 100.htm template.

Oracle Ex. 1009, p. 627
Oracle v. DeBellis
IPR2024-00221

Note

If you want the background to show through the table, you can export the datasheet without marking the Save Formatted check box. Open the .html file in Notepad and replace the <BODY> tag with <BODY background = 100.jpg> and add the <IMG SRC = "msaccess.jpg"> tag immediately above the <\HTML> tag. Standard HTML tables with no background color specified are transparent, so the background appears under the table region.

Exporting Reports to HTML

You can export an Access report to HTML in a manner similar to that for table or query datasheets. Unlike static datasheets, exporting a multipage report creates multiple Web pages, one for each page of the report. Office 97 includes a special template, Nwindtem.htm in the ...\Office\Samples folder, specifically designed for reports. Nwindtem.htm, shown in Figure 17.17, includes hyperlinks to provide paging through of multipage reports.

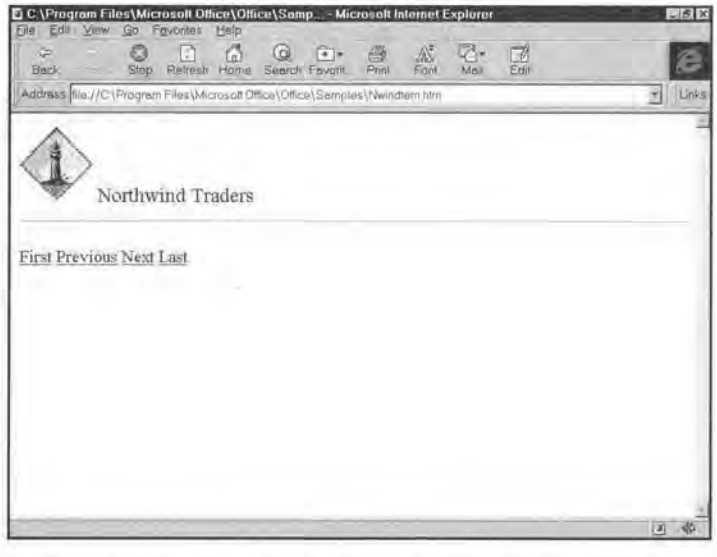


FIG. 17.17 The Nwindtem.htm template for exported Access reports.

Figure 17.18 shows the HTML source for Nwindtem.htm in Notepad. <META ...> tags provide HTML metadata to indicate the type of content of the page and other information to aid the browser in formatting the document. The <A HREF=...>*Position*</A> anchor tags provide First, Previous, Next, and Last navigation hyperlinks at the bottom of each report page.

Oracle Ex. 1009, p. 628
Oracle v. DeBellis
IPR2024-00221

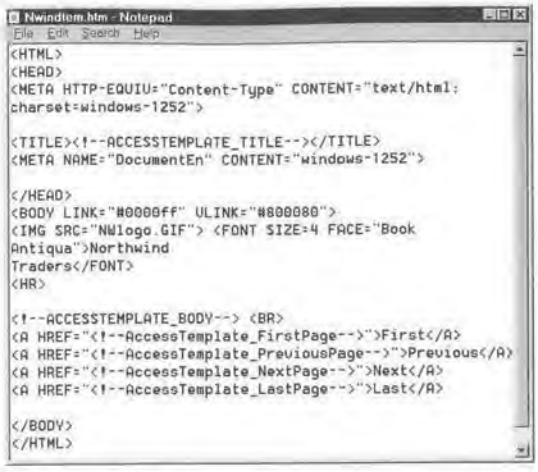


FIG. 17.18 The HTML source for the Nwindtem.htm report template with navigation hyperlinks.

Note

You can only export Access Reports to static Web pages. This limitation is logical because a Web page is the equivalent of a printed report, which must be physically replaced when updated. Unlike Word 97's File, Export to HTML menu choice, the exporting process for Access 97 reports doesn't process graphic images. If you want to add graphics to a report beyond the image(s) added by the template, you must create a .gif, .jpg, or .png file from each graphic image of the report, then manually add a `<IMG SRC="filename.ext">` tag in the appropriate location of each report page. The graphics files must be located in the same folder as the associated .html file, unless you add a well-formed path to the *filename.ext* element of the tag.

To export the Catalog report of Northwind.mdb to a series of Web pages using Nwindtem.htm to add a logo and navigation features, follow these steps:

1. Open the Catalog report in Preview mode.

   Note

   If you haven't copied the graphics files required for your template file to the folder in which you intend to store the .html files of the Catalog report, do so at this point. Step 2 in the preceding section describes how to copy the required graphics files.

2. Choose File, Save As, select the To an External File or Database option, and click OK to open the Save Report 'Catalog' In dialog. Select HTML Files (*.html, *.htm) in the Files of Type drop-down list and mark the Autostart check box. (The Save Formatted check box is marked and disabled; you can't export an unformatted report.) Click Export to continue.

Oracle Ex. 1009, p. 629
Oracle v. DeBellis
IPR2024-00221

3. When the HTML Options dialog appears, click the Browse button and select the Nwindtem.htm template in the HTML Template to Use dialog.

4. Click OK to complete the export process and open the first page of the Catalog report in your browser (see Figure 17.19). As expected, the large Northwind Traders graphic is missing from the first page, Catalog.html.

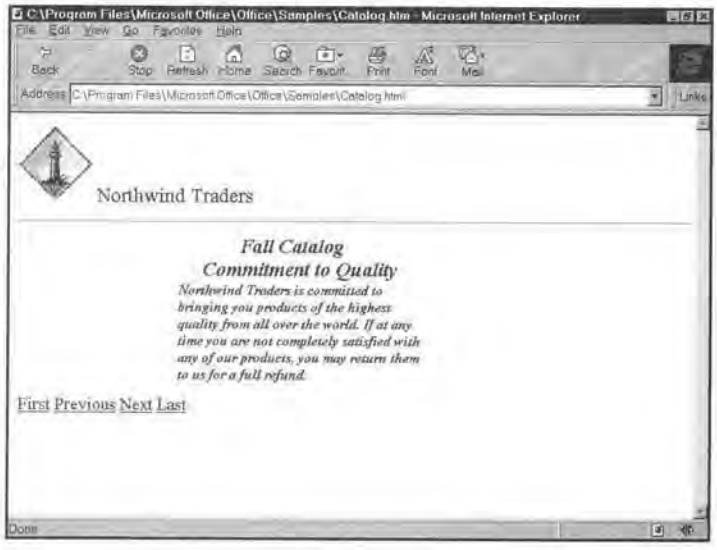


FIG. 17.19 The first Web page (Catalog.html) of the exported Catalog report.

5. Click the Next hyperlink to proceed to the second page of the Catalog, then scroll to the bottom of the page and click the Next button to display the third page (see Figure 17.20). When you export a report, the Access export feature appends `Page#` to the file name of the report for pages 2 and higher.

6. Choose View, Source from IE 3.0 to display the HTML source for CatalogPage3.html in Notepad, then scroll to the bottom of the file (see Figure 17.21). Each page of a report has a different set of HTML Previous and Next anchor lines for navigation. For page 3 of the catalog, the Previous and Next anchor lines are `<A HREF="CatalogPage2.html">Previous</A>` and `<A HREF="CatalogPage4.html">Next </A>`, respectively. The First (`<A HREF="Catalog.html">First</A>`) and Last (`<A HREF="CatalogPage9.html">Last</A>`) anchors are the same for all pages of the report.

Reports are formatted as HTML tables without borders (`<TABLE BORDER=0>`), which emulates on a Web page the appearance of Access 97's reports in Preview mode and when printed. As a general rule, Access reports are the best choice for exporting large amounts of data to Web pages. You have much more control over the appearance of the Web page with exported reports than when you export a datasheet with the same content. To

Oracle Ex. 1009, p. 630
Oracle v. DeBellis
IPR2024-00221

optimize the appearance of Web pages created from reports, you must design the report specifically for export to HTML.

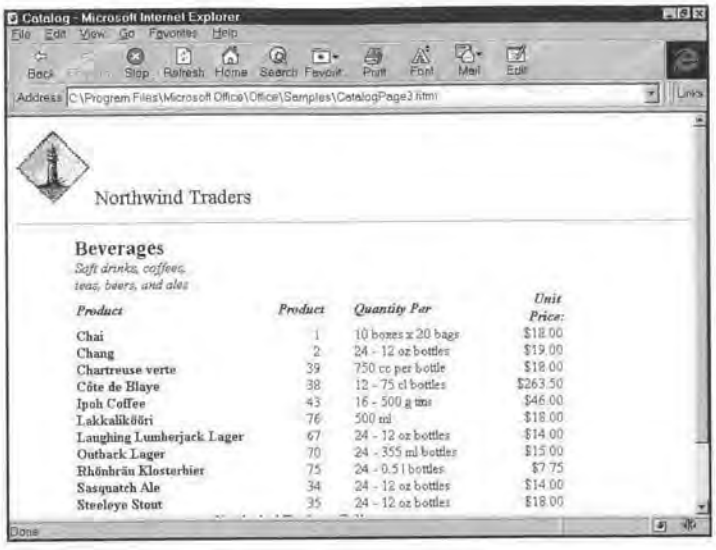


FIG. 17.20 The third Web page (CatalogPage3.html) of the Catalog report.

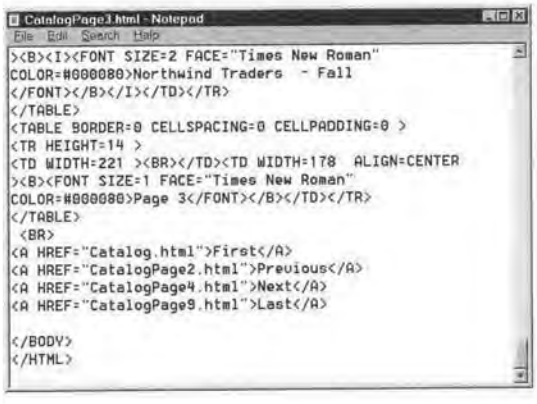


FIG. 17.21 The anchor tags of the third page (CatalogPage3.html) of the Catalog report.

Note

Report formatting fails for complex reports that contain a combination of graphic elements and text. As an example, compare the last Web page (CatalogPage9.html) of the Catalog series with the last page of the Catalog report in Access' Report Preview mode. All of the graphic elements in the order form are missing from the last Web page.

Oracle Ex. 1009, p. 631
Oracle v. DeBellis
IPR2024-00221

Using the Web Publishing Wizard to Create a Static Web Site

If you have a Web server, such as Internet Information Server 2+ or you've installed the Microsoft Personal Web Server from the Office Value Pack, you can create a complete Web site, including a rather drab home page, with the Web Publishing Wizard. The Web Publishing Wizard is an alternative to creating individual pages or sets of pages from Access objects. The wizard provides an additional level of automation that can speed the creation of larger Web sites that distribute data from Access or ODBC data sources.

To try the Web Publishing Wizard with Northwind.mdb and the Nwindtem.htm template, create a folder on your Web server, such as \InetPubs\wwwroot\Nwind, for the Web page files and follow these steps:

1. With Northwind.mdb open in Access, choose File, Save As HTML to open the first Web Publishing Wizard dialog (see Figure 17.22). When you first use the wizard, you haven't created any publication profiles, so the I Want to Use a Web Publishing Profile I Already Created with This Wizard check box is disabled. Click Next to continue.

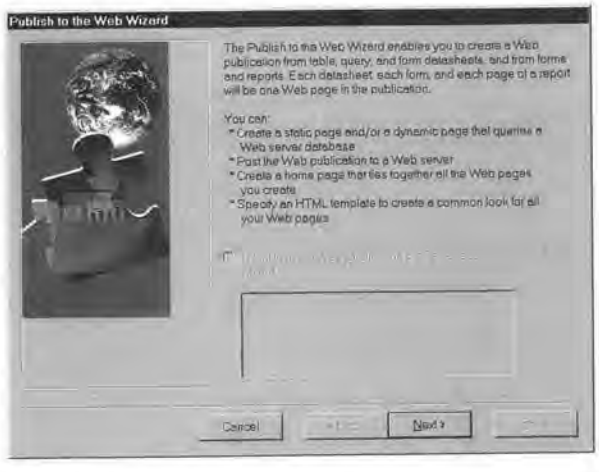


FIG. 17.22 The Web Publishing Wizard's opening dialog.

2. In the second wizard dialog, select the objects you want to export by clicking the object type tab, then marking the check boxes of each object you want to include (see Figure 17.23). You can select any combination of tables, queries, and reports for a static Web site. Forms are restricted to dynamic Web sites that use Active Server Page (.asp) files. Dynamic Web sites are the subject of the next chapter. Click Next.

Oracle Ex. 1009, p. 632
Oracle v. DeBellis
IPR2024-00221

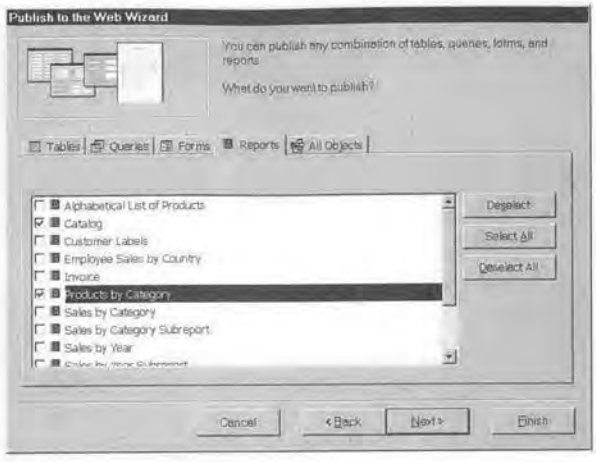


FIG. 17.23 Selecting table, query, and report objects for export to static Web pages.

3. Type the well-formed path and file name of the template you want to use for your Web pages, in this case **Nwindtem.htm** (see Figure 17.24). Alternatively, click the Browse button and locate the template file in the HTML Template to Use dialog. If you mark the I Want to Select Different Templates for Some of the Selected Objects check box, you are prompted to choose a template file for each object. Click Next.

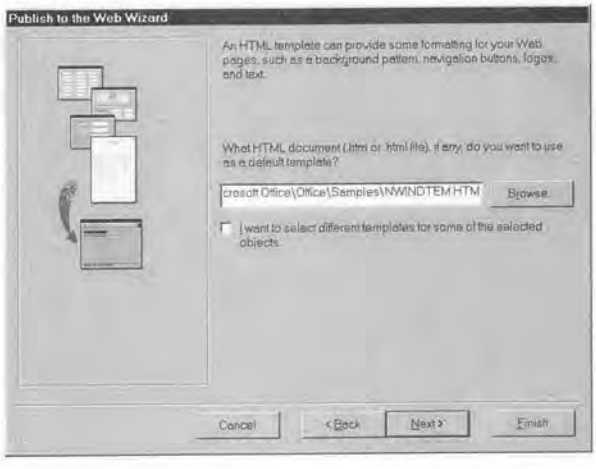


FIG. 17.24 Specifying a single template for all exported Web pages.

4. Select the Static HTML option to create static HTML pages from your selected objects (see Figure 17.25). The next chapter shows you how to use the Dynamic HTX/IDC and Dynamic ASP options. Click Next.

Oracle Ex. 1009, p. 633
Oracle v. DeBellis
IPR2024-00221

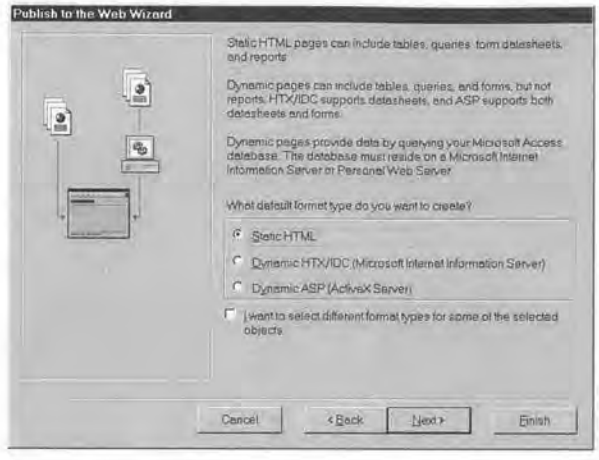


FIG. 17.25 Selecting the Static HTML option for the exported Web pages.

5. Type the well-formed path to the folder for your Web pages in the text box (see Figure 17.26); use UNC if you're connected to the server via a network. Select the No, I Want to Publish Objects Locally option if you are creating the Web pages on a PC that's running or is connected via a network to the Web server. Click Next.

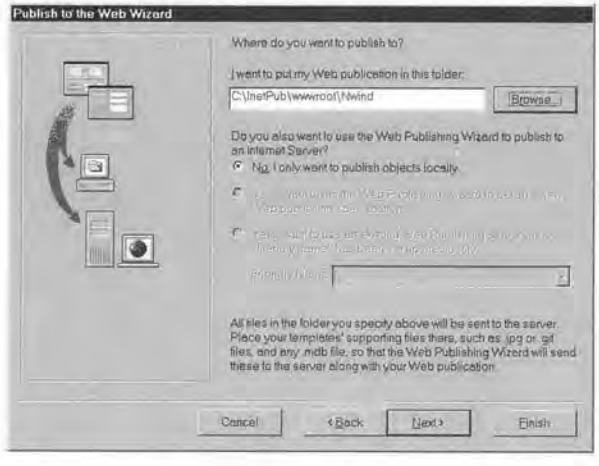


FIG. 17.26 Specifying the path to the Web server's folder for the exported pages.

6. To create a Spartan home page, consisting of a switchboard table with hyperlinks to Web pages created from your Access object, mark the Yes, I Want to Create a Home Page check box. Accept Default in the text box to use Default.htm for the home page file (see Figure 17.27). Click Next.

Oracle Ex. 1009, p. 634
Oracle v. DeBellis
IPR2024-00221

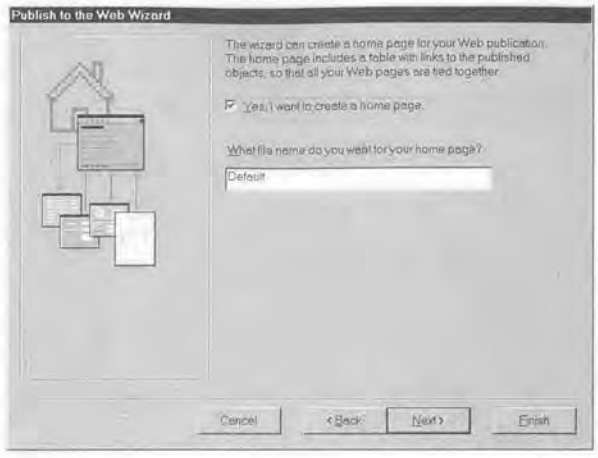


FIG. 17.27 Instructing the wizard to create a home page "switchboard."

7. To save a Web Publication Profile, mark the Yes, I Want to Save Wizard Answers to a Web Publication Profile check box. Type a name for the Profile in the Profile Name text box (see Figure 17.28). Saving a Profile lets you skip the specification process when you update your Web pages. Click Finish to create the Web site by exporting the Access objects.

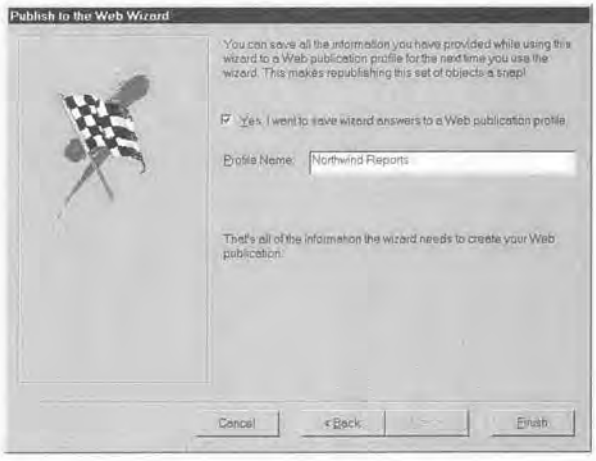


FIG. 17.28 Saving your export specification as a Web Publication Profile.

8. Copy the graphics file(s) for your template (Nwlogo.gif for Nwindtem.htm) to the Web server folder you specified in step 5.

9. Launch your browser and type the **http://*servername*/Nwind** URL in the Address text box to open the Northwind Traders home page you just created (see Figure 17.29). Click the hyperlinks in the Object column of the Switchboard table to open the individual Web pages.

Oracle Ex. 1009, p. 635
Oracle v. DeBellis
IPR2024-00221



FIG. 17.29 The home page's Switchboard table with jumps to each of the exported pages.

Note

The hyperlinks for the anchors of the Nwindtem.htm template aren't appropriate for the Switchboard page or single Web pages created from datasheets. If you include reports in your Web site, create one template for pages based on datasheets and another template for pages created from reports. Mark the I Want to Select Different Templates for Some of the Selected Objects check box in the third wizard form (in the preceding step 3) and specify the template to use with each selected object when prompted.

Importing Data from HTML Tables

Access 97 includes the capability to import or link data from an HTML table to a Jet 3.5 table. This feature appears to have been included in Access 97 for HTML symmetry; the theory apparently is that if one can export to HTML, one also should be able to import from HTML. Few Access users are likely to employ this feature because relatively little useful tabular data is available on the Internet. For completeness, however, following is an example of importing data from an HTML page created in the section "Exporting Table and Query Datasheets to HTML" earlier in this chapter:

1. Choose File, Get External Data, Import to open the Import dialog. Select HTML Documents (*.html, *.htm) in the Files of Type drop-down list (see Figure 17.30).
2. Select the Suppliers.html file you created in the "Exporting Table and Query Datasheets to HTML" section at the beginning of this chapter and click the Import button to close the Import dialog and open the first Import HTML Wizard dialog.

Oracle Ex. 1009, p. 636
Oracle v. DeBellis
IPR2024-00221

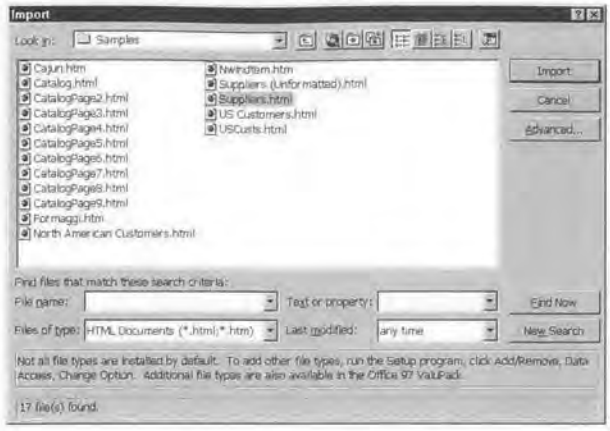


FIG. 17.30 Displaying .html and .htm files in the Import dialog.

3. The wizard imports the table header data, if present, together with the table data. The formatted version of Suppliers.html includes headers, so check the First Row Contains Column Headings check box (see Figure 17.31).



FIG. 17.31 Specifying column headers in the Import HTML Wizard's first dialog.

4. Click the Advanced button to open the Suppliers Import Specification dialog. This dialog lets you customize import operations on date and time fields and choose the decimal symbol. You can change the field names, data types, and indexing for each of the fields, as well as skip the import of specific fields. Change the data type of the Supplier ID field to Long Integer, and specify a No Duplicates index on the field (see Figure 17.32). Data types and indexes for the remaining fields are satisfactory.

Oracle Ex. 1009, p. 637
Oracle v. DeBellis
IPR2024-00221

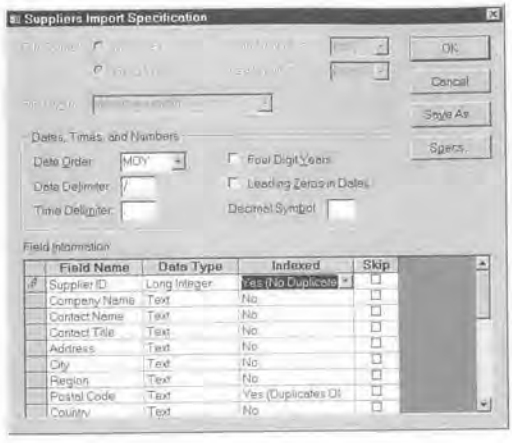


FIG. 17.32 Specifying field data types and indexes in the Suppliers Import Specification dialog.

> **Note**
>
> Alternatively, you can specify field data types and indexing in the third wizard dialog. The wizard proposes to add Duplicates OK indexes on any field that contains `ID` or `Code` in the column name.

5. Click the Save As button to save the Import Specification. Edit or type a new name for the specification in the Specification Name of the Save Import/Export Specification dialog (see Figure 17.33). Click OK to close the dialog, then click OK to close the Suppliers Import Specification dialog.

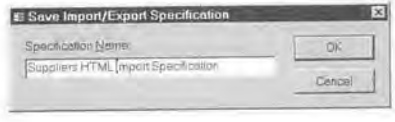


FIG. 17.33 Editing the default name for the saved Suppliers Import Specification.

Next>

6. Click Next to display the second Import HTML Wizard dialog. Select the In a New Table option to store the data in a new table whose name you specify at the end of the import process (see Figure 17.34). Click Next.

Next>

7. You can make last-minute changes to field names, data types, and indexes in the third wizard dialog (see Figure 17.35). If you made the required changes to the Suppliers Import Specification dialog in preceding step 4, click Next.

Next>

8. By default, the wizard proposes to add a numeric primary key field to the table. The Supplier ID field qualifies as a primary key, so select the Choose My Own Primary Key option and pick the Supplier ID field in the drop-down list (see Figure 17.36). Click Next.

Oracle Ex. 1009, p. 638
Oracle v. DeBellis
IPR2024-00221

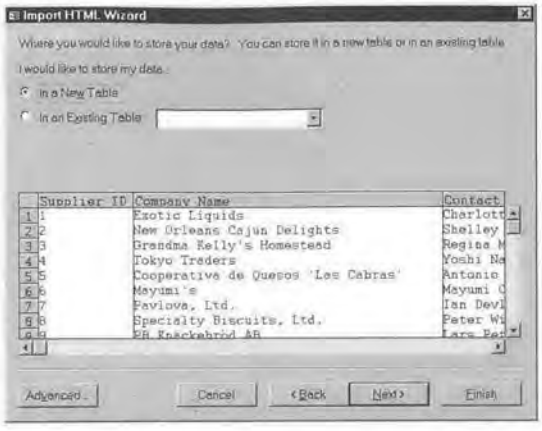


FIG. 17.34 Specifying import of the tabular HTML data to a new table.



FIG. 17.35 A second chance to change field names, data types, and indexes offered by the third wizard dialog.



9. There is a Suppliers table in Northwind.mdb, so edit the proposed table name to tblSuppliers (see Figure 17.37). You don't need to analyze the Suppliers table, so don't mark the I Would Like a Wizard to Analyze check box. Click Finish to export the HTML table to tblSuppliers and terminate the wizard.

10. Open the tblSuppliers table to verify the import wizardry. The table is essentially identical to the original Suppliers table from which the Web page was created (see Figure 17.38). The most significant differences are the field data type of the Supplier ID column (Long Integer, not AutoNumber) and the field names (derived from the Caption property of the original table).

Oracle Ex. 1009, p. 639
Oracle v. DeBellis
IPR2024-00221

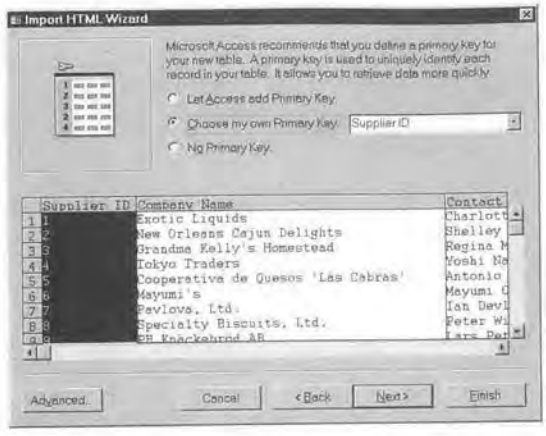


FIG. 17.36 Selecting the Supplier ID field as the primary key for the imported table.

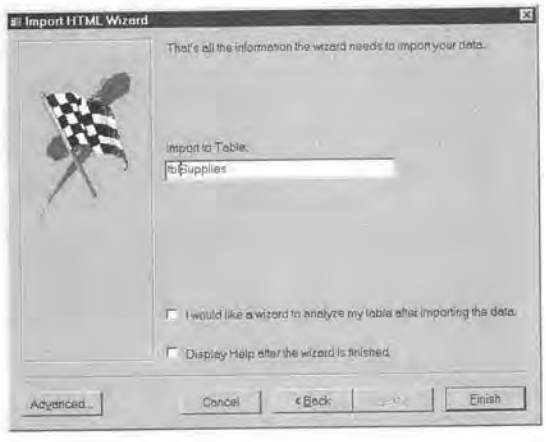


FIG. 17.37 Editing the name of the new table in the final wizard dialog.



FIG. 17.38 The exported tblSuppliers table in Datasheet view.

Oracle Ex. 1009, p. 640
Oracle v. DeBellis
IPR2024-00221

Note

Most of the information available on the Internet is subject to copyright, either explicitly (by a copyright notice on the Web site's home page) or implicitly (by statute). Importing and using copyrighted content from the Internet for most purposes is prohibited by federal copyright statutes, unless you have express permission of the copyright owner to use the content. If you intend to import and use information created by others and published on the Internet, consult an attorney before using the information.

Oracle Ex. 1009, p. 641
Oracle v. DeBellis
IPR2024-00221

Chapter 18

Creating Dynamic Web Pages

Microsoft's rallying cry, "Activate the Internet," and the firm's use of the "Active" prefix for the majority of its new Internet-related technologies are intended to do away with the static Web pages described in the preceding chapter. Static pages constitute by far the majority of the content available on today's Internet. Microsoft wants you to use Internet Information Server (IIS) 3.0, ActiveX controls, and Visual Basic Script (VBS) to create interactive Web pages. Flashy graphics, hot audio riffs, and even some digital video content now greet members of the newly revamped and highly activated Microsoft Network. Of course, users need Internet Explorer (IE) 3+ to take full advantage of what Microsoft calls "exciting, compelling, and rich" content. By advancing the technology to create interactive Web pages, Microsoft also gains market share in the browser war.

Giving intranet and Internet users the ability to define their own Web-based view of data isn't as exciting as delivering "rich multimedia content" (one of Microsoft's favorite clichés) from your Web site. Corporate intranets and the Internet offer the opportunity for users to interact with databases, displaying custom views of information and updating entries from Web browsers. Electronic commerce on the Internet depends almost entirely on consumer- and business-related interaction with databases.

This chapter introduces you to the use of new Microsoft technologies, the *Internet Database Connector* (IDC) and the *Active Server Pages* (ASP, formerly *ActiveX Server Framework*), that let you create dynamic Web pages for displaying and updating data. The IDC examples of this chapter require you to have IIS 2+ or the Microsoft Personal Web Server installed and operating. The ASP examples require IIS 3.0 (or IIS 2.0 and the ASP extensions, code-named *Denali*), which was in the beta-testing stage when this book was written. You also can use the ASP extensions with the Personal Web Server running under Windows 95.

Oracle Ex. 1009, p. 642
Oracle v. DeBellis
IPR2024-00221

Note

The term *users* to define consumers of information implies to many that information providers should be referred to as *pushers*.

Understanding Microsoft's View of Databases and the Web

Giving away Web browsers, including Web servers with Windows NT Server 4.0, and distributing free Internet enhancements for browsers and servers isn't a business model that's likely to add much to Microsoft's bottom line. To earn a return on its Internet investment, Microsoft ultimately must sell, not give away, products. The initial objective of Microsoft's Internet program is to increase sales of Windows NT Server 4.0 as an Internet/intranet server and to make SQL Server 6.5 the client/server database of choice for conducting Internet commerce and distributing information from corporate databases over private intranets. Data marts, data malls, and distributed transactions are the keys to Microsoft's success in the back-end of the Internet/intranet business.

Rolling Up Data to Marts and Malls

Most large organizations have established or plan to establish intranets that let networked users of Web browsers query data that is stored in databases running on mainframes or minicomputers. Letting users run queries directly against these business-critical transaction databases is an anathema to the vast majority of information services managers. *Data warehouses* make organization-wide information available without disturbing day-to-day database operations.

Periodically, data from the transaction databases is downloaded, often in summarized form (called *rollups*), to a client/server data warehouse. Smaller *data marts*, which hold up to about 250G of data, serve individual divisions or departments. Data malls consist of groups of data marts against which users can execute *distributed queries*. A distributed query can create joins between databases running on separate data mart servers. Data marts and malls are ideal applications for SQL Server 6.5 running under Windows NT Server 4.0. Microsoft doesn't give Windows NT Server or SQL Server away; you must pay for these products and, in addition, pay a license fee for each client PC that connects to them.

Note

The demand for data warehouses, marts, and malls became so great in mid-1996 that the large, fast fixed-disk drives (4G and up) that are needed for database servers were difficult to find. 20G and larger Wide, Fast SCSI-2 and SCSI-3 drives are expected to be available from several suppliers beginning in early 1997. If the remarkable increase in the demand for large disk drives continues its 1996 growth rate, these new drives also are likely to be scarce.

Although you're not likely to see Jet 3.5 databases used for data marts and malls, Access 97 is a candidate for designing, testing, and maintaining such client/server databases.

Oracle Ex. 1009, p. 643
Oracle v. DeBellis
IPR2024-00221

Access 97 makes it easy to link Jet 3.5 tables to SQL Server 6.5 databases by using the SQL Server ODBC driver included with Office 97. This chapter uses the Access ODBC driver for Jet 3.5 databases to create dynamic Web pages for displaying and updating data. The experience you gain in this chapter is directly applicable when you move to the client/server RDBMS model.

◀◀ See "Linking an Excel Worksheet by Using the ODBC API," p. 228
▶▶ See "Understanding ODBC Drivers," p. 918



Managing Distributed Transactions

One of the primary requirements of electronic commerce on the Internet is the ability to update multiple databases, regardless of their location. As an example, a Web-based credit-card purchase requires verifying the purchaser's account, testing for available inventory of the ordered items, placing a hold amount against the purchaser's available credit limit, and entering the order. Such a transaction might involve three or more databases residing on mainframes, minicomputers, or PCs. A transaction requires that all updates involved in a purchase succeed; if any update fails, the transaction must be rolled back. Rolling back a transaction restores the state of the participating database tables to the state that existed immediately prior to the start of the transaction. A *distributed transaction* is a transaction that takes place on tables residing in more than one database. In the case of a credit-card purchase, the databases are likely to be owned by different firms and connected over a wide-area network (WAN), such as (but not necessarily) the Internet.

A discourse on distributed transaction technology is beyond the scope of this book, but the subject is of vital importance to the future of Internet-based commerce. SQL Server 6.5 now includes a Distributed Transaction Controller (DTC) to coordinate the execution of transactions on multiple servers running SQL Server 6.5. DTC and the new *Microsoft Transaction Server* (MTx) technology, code-named *Viper* when this book was written, extends DTC to the full range of database servers, including IBM mainframes connected via Microsoft SNA Server. If MTx is successful, Microsoft will gain a greater share of the market for Internet commerce back-ends, which are typified by the firm's fledgling Microsoft Merchant Server and Commercial Internet Services.

Using the IDC for Delivering Information

The IDC is an *Internet Services API* (ISAPI) application that simplifies the creation of active Web pages. These Web pages automatically deliver updated information and let users process queries against any database for which a 32-bit ODBC driver is available. ISAPI applications use Windows *dynamic link libraries* (DLLs) as a substitute for the *Common Gateway Interface* (CGI) applications (scripts) originally developed for UNIX Web servers. ISAPI runs only on Microsoft's Internet Information Server and Personal Web Server. The advantage of ISAPI DLLs is that they run in the Web server's process space, which makes ISAPI DLLs run faster than executable CGI applications. IDC is implemented by Httpodbc.dll, a 32-bit ISAPI DLL.

Oracle Ex. 1009, p. 644
Oracle v. DeBellis
IPR2024-00221

An IDC application requires the following two types of files:

- *Internet Data Connector (.idc) files* specify the ODBC data source and the query that supplies the data for the Web page in a simple text format. The .idc file also can contain optional information, such as a user name and password for access to the data source.
- *HTML Extension (.htx) files*, commonly called template files, provide the HTML source to format the data for viewing. The data generated by the .idc file is merged with the .htx file to create a Web page. The .idc and .htx files are paired for a particular Web page, although it's possible to use a single .idc file with multiple .htx files. Templates can contain HTML forms with common Windows control objects, such as text boxes and drop-down lists to create Web pages with customized data content.

The process of generating a dynamic Web page with .idc and .htx files follows this sequence:

1. The browser sends a URL in the format `pagename.idc?` to IIS 2.0, usually from a hyperlink on another page.
2. The IDC DLL, Httpodbc.dll, reads *Pagename*.idc to obtain the information that is necessary to connect to the database and execute the query.
3. Httpodbc.dll connects to the data source and sends the query statement, which must be in the dialect of SQL for the RDBMS specified by the data source.
4. Httpodbc.dll receives the rows from the data source and merges the rows with the *Pagename*.htx file.
5. Httpodbc.dll sends the merged document, *Pagename*.htm, to IIS 2.0, which delivers the page to the requesting browser.

Fortunately, IDC does all this work for you. You need to create only an .idc text file and a relatively simple .htx file.

The IDC Demonstration Applications Included with IIS 2.0

Internet Information Server 2.0, included with Windows NT Server 4.0, offers a simple demonstration of the use of .idc and .htx files with the pubs sample database of SQL Server 6.5. To use the IDC samples included with IIS 2.0, you must have access to SQL Server 6.x from the server on which you run IIS 2.0 or a copy of the pubs database in Jet 3.0 (Access 95) or Jet 3.5 (Access 97) format. If you don't have these capabilities, the section "A Dynamic Web-Based Access Application" later in this chapter directs you to an Internet site that offers a more complex demonstration of dynamic Web-page generation with the IDC.

Oracle Ex. 1009, p. 645
Oracle v. DeBellis
IPR2024-00221

Note

Microsoft's dbWeb 1.1a, described in the "An Overview of Microsoft's Internet Products" section of Chapter 16, "Working with Hyperlinks and HTML," includes Pubs.mdb, a Jet version of the pubs sample database. If you've installed dbWeb 1.1 or 1.1a, you can use the Microsoft Access ODBC driver to connect to Pubs.mdb.

Before you can run the IIS 2.0 database example applications, you must create an ODBC data source named Web SQL. This procedure assumes that you are working at the server PC on which IIS 2.0 is running because you must create a system data source for IIS 2.0. To create the data source and run a simple IDC query, follow these steps:

1. Launch IE 3.0 or the default browser and type **http://*servername*/samples/dbsamp/dbsamp.htm** in the Address text box to open the Internet Database Connector Examples page.
2. Scroll to the line that includes click here to create it and click the hear hyperlink to display the Create ODBC Datasource on the Microsoft Internet Information Server page.

   If you have and want to use Pubs.mdb, instead of SQL Server's pubs database, click the ODBC data source tools hyperlink, and then click the Microsoft Access Driver (*.mdb) hyperlink.
3. In the Datasource Name to Create text box, type **Web SQL** (with a space between **Web** and **SQL**). In the Server Name text box, type the name of the server on which SQL Server 6.x is installed.
4. In the Attribute String text box, type **DATABASE=pubs** (see Figure 18.1).

   If you're using Pubs.mdb, type ***d*:*path*\pubs.mdb**, where ***d*:*path*** is the location of Pubs.mdb, in the Database Name text box.
5. Click the Create Datasource button to create the data source. A new page displays a message indicating that the data source was created successfully.

Note

The IDC samples assume that you are using integrated security for SQL Server. Integrated security authenticates SQL Server users based on their Windows NT logon ID and password. If you are not using integrated security, the samples assume that you can connect to SQL Server 6.x with the default system administrator account, sa, with no password.

To use another account and/or a password, you must edit the Sample.idc file, as described in the following section, before you attempt to run the query in the following steps. Unfortunately, the password you specify is accessible to others in the clear text of the .idc file, which is a breach of SQL Server security.

Oracle Ex. 1009, p. 646
Oracle v. DeBellis
IPR2024-00221

6. Use the Back button to return to the dbsamp1.htm page and scroll to the Other Database Examples section at the bottom of the page (see Figure 18.2). Click the Simple database query hyperlink to open the Internet Database Connector Example 1: Simple Query page.

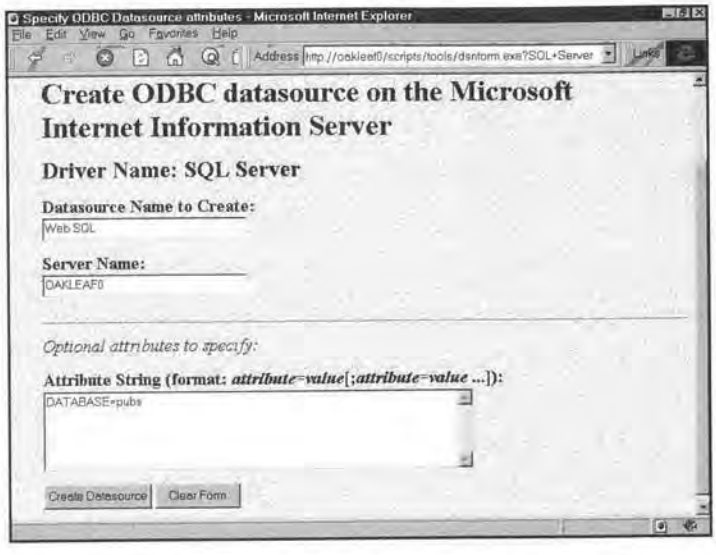


FIG. 18.1 Specifying the parameters for the Web SQL ODBC data source for SQL Server.

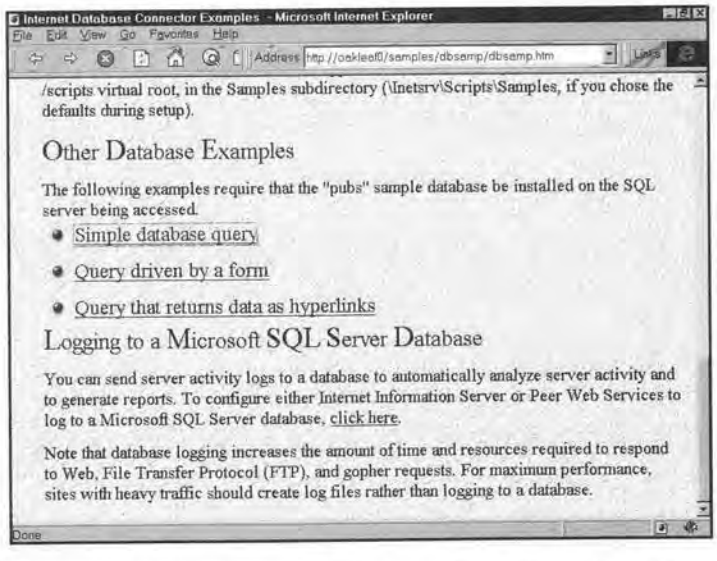


FIG. 18.2 Hyperlinks to the sample database applications that are included with IIS 2.0.

Oracle Ex. 1009, p. 647
Oracle v. DeBellis
IPR2024-00221

7. Click the single hyperlink below the caption line to run the query. The query displays the names of authors whose books have generated more than $5,000 in year-to-date sales (see Figure 18.3).

8. If you choose View, Source, Notepad displays the HTML source for Sample.htm—the name of the Web page created by the Sample.htx template file and the Sample.idc IDC file (see Figure 18.4). By default, the .htx, .idc, and .htm files have the same file name.

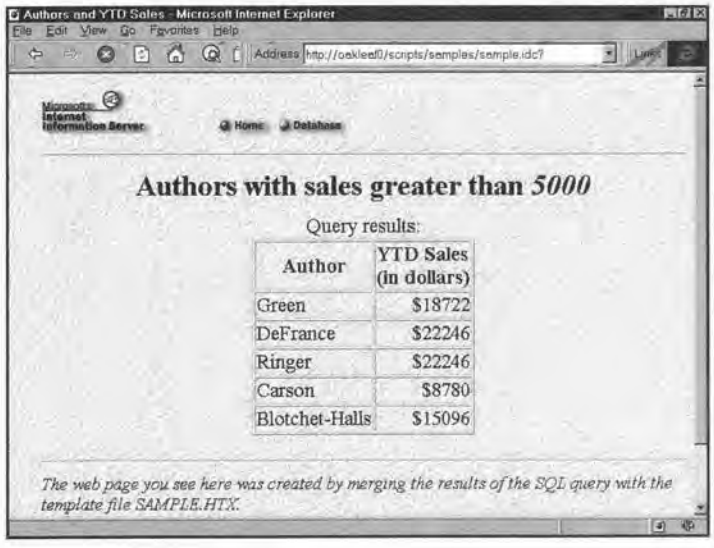


FIG. 18.3 The Sample.htm page created by Sample.htx and a static query.

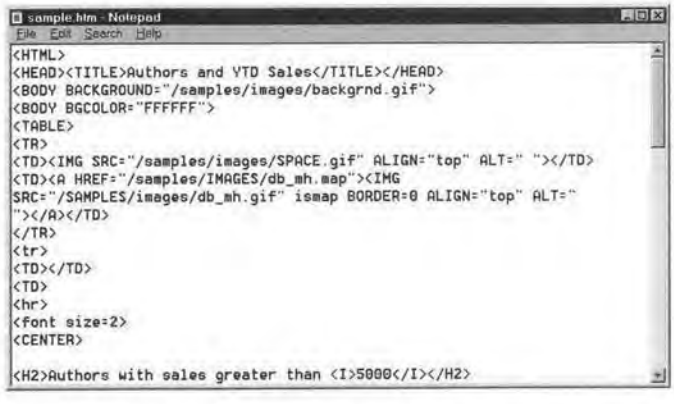


FIG. 18.4 Part of the HTML source for the Sample.htm Web page.

The Sample.idc and Sample2.idc File. The Sample.idc file, located in the \InetPub\Scripts\Samples folder, specifies the name of the ODBC data source, the user name for SQL Server, a password (if needed), the name of the template file, and the SQL statement

Oracle Ex. 1009, p. 648
Oracle v. DeBellis
IPR2024-00221

required to execute the query. For Jet data sources, you can copy and paste the SQL statement for an Access query to create the `SQLStatement:` parameter. Listing 18.1 shows the content of Sample.idc.

Listing 18.1 The Data Source Parameters and SQL Query Statement of Sample.idc

```
Datasource: Web SQL
Username: sa
Template: sample.htx
SQLStatement:
+ SELECT au_lname, ytd_sales
+ FROM pubs.dbo.titleview
+ WHERE ytd_sales>5000
```

The plus (+) symbols preceding the lines of the SQL statement are continuation characters, indicating that the text is a continuation of the line above. Minor formatting changes have been made to the text of Listing 18.1 to improve readability.

Note

If you need a password to log in to SQL Server, add `Password:` *password* after the `Username:` *userid* line. When using integrated security, IDC ignores the `Username:` and `Password:` entries.

The Internet Database Connector Examples page provides a link to a parameterized query that lets you enter the year-to-date sales criteria in an HTML form. Click the Query driven by a form hyperlink to display the Dbsamp2.htm form, shown in Figure 18.5. You enter the criterion value in the text box, and then click the Run Query button to execute Sample2.idc.

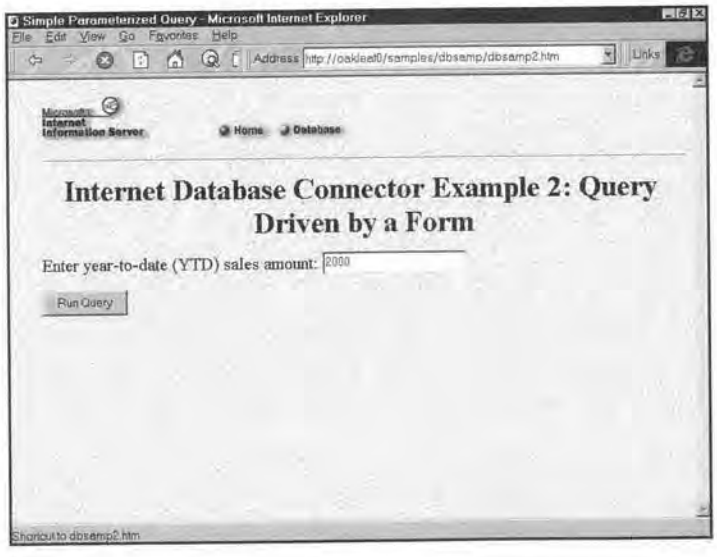


FIG. 18.5 The Dbsamp2.htm form for executing a parameterized query.

Oracle Ex. 1009, p. 649
Oracle v. DeBellis
IPR2024-00221

Listing 18.2 The HTML Source for the Form Page

```
<HTML>
<HEAD><TITLE>Simple Parameterized Query</TITLE></HEAD>
<BODY BACKGROUND="/samples/images/backgrnd.gif">
<BODY BGCOLOR="FFFFFF">
<TABLE>
<TR>
<TD><IMG SRC="/samples/images/SPACE.gif" ALIGN="top" ALT=" "></TD>
<TD><A HREF="/samples/IMAGES/db_mh.map">
<IMG SRC="/SAMPLES/images/db_mh.gif" ismap BORDER=0 ALIGN="top" ALT=" ">
➥</A></TD>
</TR>
<TR>
<TD></TD>
<TD>
<HR>
<FONT SIZE=2>
<CENTER>
<H2>Internet Database Connector Example 2: Query Driven by a Form</H2>
</CENTER>
<FORM METHOD="POST" ACTION="/scripts/samples/sample2.idc">
<P>
Enter year-to-date (YTD) sales amount: <INPUT NAME="sales" VALUE="5000">
<P>
<INPUT TYPE="SUBMIT" VALUE="Run Query">
</FORM>
</FONT>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

The `<FORM METHOD="POST" ACTION="/scripts/samples/sample2.idc">` tag defines an HTML form that executes Sample2.idc. Paths within Web pages stored on the server are relative to the Web server root folder, \InetPub, not the default \InetPub\Wwwroot folder. The `sales` text box created by the `<INPUT NAME="sales" VALUE="5000">` tag has a default value of 5000. When you click the Run Query button created by the `<INPUT TYPE="SUBMIT" VALUE="Run Query">` tag, the sales value is passed to Sample2.idc's replaceable parameter, `%sales%`. Listing 18.3 shows the text of Sample2.idc with `%sales%` as the criterion for the `ytd_sales` field of `titleview`.

Listing 18.3 The Parameterized Query of Sample2.idc

```
Datasource: Web SQL
Username: sa
Template: sample.htx
SQLStatement:
+ SELECT au_lname, ytd_sales
+ FROM pubs.dbo.titleview
+ WHERE ytd_sales > %sales%
```

Oracle Ex. 1009, p. 650
Oracle v. DeBellis
IPR2024-00221

Note

Only the `Datasource`, `Template`, and `SQLStatment` fields are required for .idc files. You can add many optional values to .idc files for fine-tuning query execution, supplying default parameter values, and setting the maximum number of records to return. For a full explanation of these optional fields, read the 08_iis.htm page of the IIS 2.0 documentation in the \Winnt\System32\Inetsrv\Iisadmin\Htmldocs folder.

The Sample.htx File. Listing 18.4 shows the content of the Sample.htx file, also located in the \InetPub\Scripts\Samples folder. Sample.htx supplies the template for the simple Sample.idc query and for the parameterized query, Sample2.idc. All IDC-specific tags in the HTML source for .htx files use the format `<%...%>`. IDC supports conditional display using the `<% if expression %>...<%else%>...<%endif%>` structure, similar to VBA's **`If`** *`expression`* **`Then...Else...End If`** statements. Sample.htx displays the `<H2>Authors with sales greater than <I><%idc.sales%></I></H2>`. The `<%idc.sales%>` tag is replaced by the value from the `sales` text box of the Dbsamp2.htm form. If the value is missing, as is the case when you use Sample.idc, the `<H2>Authors with sales greater than <I>5000</I></H2>` line appears in the generated page.

Listing 18.4 The HTML Source and IDC-Specific Tags of the Sample.htx File

```
<HTML>
<HEAD><TITLE>Authors and YTD Sales</TITLE></HEAD>
<BODY BACKGROUND="/samples/images/backgrnd.gif">
<BODY BGCOLOR="FFFFFF">
<TABLE>
<TR>
<TD><IMG SRC="/samples/images/SPACE.gif" ALIGN="top" ALT=" "></TD>
<TD><A HREF="/samples/IMAGES/db_mh.map">
<IMG SRC="/SAMPLES/images/db_mh.gif" ismap BORDER=0 ALIGN="top" ALT=" ">
➥</A></TD>
</TR>
<tr>
<TD></TD>
<TD>
<hr>
<font size=2>
<CENTER>
<%if idc.sales eq ""%>
<H2>Authors with sales greater than <I>5000</I></H2>
<%else%>
<H2>Authors with sales greater than <I><%idc.sales%></I></H2>
<%endif%>
<P>
<TABLE BORDER>
<%begindetail%>
<%if CurrentRecord EQ 0 %>
<caption>Query results:</caption>
<TR>
<TH><B>Author</B></TH><TH><B>YTD Sales<BR>(in dollars)</B></TH>
</TR>
```


Oracle Ex. 1009, p. 651
Oracle v. DeBellis
IPR2024-00221

```
<%endif%>
<TR><TD><%au_lname%></TD><TD align="right">$<%ytd_sales%></TD></TR>
<%enddetail%>
<P>
</TABLE>
</center>
<P>
<%if CurrentRecord EQ 0 %>
<I><B>Sorry, no authors had YTD sales greater than </I><%idc.sales%>.</B>
<P>
<%else%>
<HR>
<I>
The web page you see here was created by merging the results
of the SQL query with the template file SAMPLE.HTX.
<P>
The merge was done by the Microsoft Internet Database Connector and
the results were returned to this web browser by the Microsoft Internet
Information Server.
</I>
<%endif%>
</FONT>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Merged data from execution of the .idc file appears between `<%begindetail%>` and `<%enddetail%>` tags. `<%columnname%>` tags, such as `<%au_lname%>` and `<%ytd_sales%>`, are replaced by the value of the query columns for the current row of the table. If no records are returned by the query, the `<%if CurrentRecord EQ 0 %>` expression evaluates true, and the `Sorry...` message appears.

A Dynamic Web-Based Access Application

Microsoft's Access JobForum application is an example of more complex dynamic Web page generation with .idc and .htx files. The JobForum site, which is similar to newspaper classified advertising, lists job openings for Access developers and also lets developers place jobs wanted listings. Microsoft created this application in January 1996 as a demonstration of the use of IDC with Jet databases. Subsequently, the site became quite active, with many "help wanted" entries added every month.

You can view a good example of a complex, dynamic Web page at:


On the Web

http://www.microsoft.com/access/internet/jobforum/

From the JobForum home page, you can choose to view, enter, or update a listing in Job Available or Job Wanted. When you choose view, you can display all listings, all listings on or after a specified date, or a specific listing by number. Figure 18.6 shows a few of the Job Available listings on and after October 1, 1996. Clicking a listing hyperlink opens a Job Available page (see Figure 18.7) derived from data stored in the JobAvailable table of the .mdb file.

Oracle Ex. 1009, p. 652
Oracle v. DeBellis
IPR2024-00221



The JobForum white paper provides a complete description of the JobForum application with detailed explanations of the .idc and .htx files used to create the JobForum pages. You can download the white paper in Word 6 .doc format or read the paper at the following URL:

http://www.microsoft.com/AccessDev/accwhite/JobForPa.htm

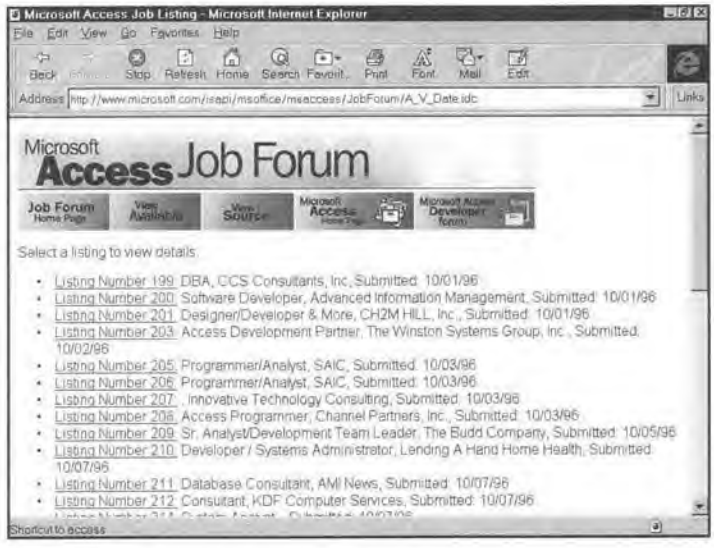


FIG. 18.6 A few of the Job Available listings as of October 1996 that are displayed by the Access JobForum application.



FIG. 18.7 A Job Information page created from data stored in a Jet database.

Oracle Ex. 1009, p. 653
Oracle v. DeBellis
IPR2024-00221

Clicking the View Source button at any point in the JobForum application displays a Web page with the HTML source for the .idc and .htx files used to create the page, together with an explanation of the source in the form of comments (<!--explanation--> tags). Figure 18.8 shows part of the commented HTML source for the Job Information page shown in Figure 18.7. The commented source pages with the JobForum white paper are very useful when learning to write .idc and .htx files.

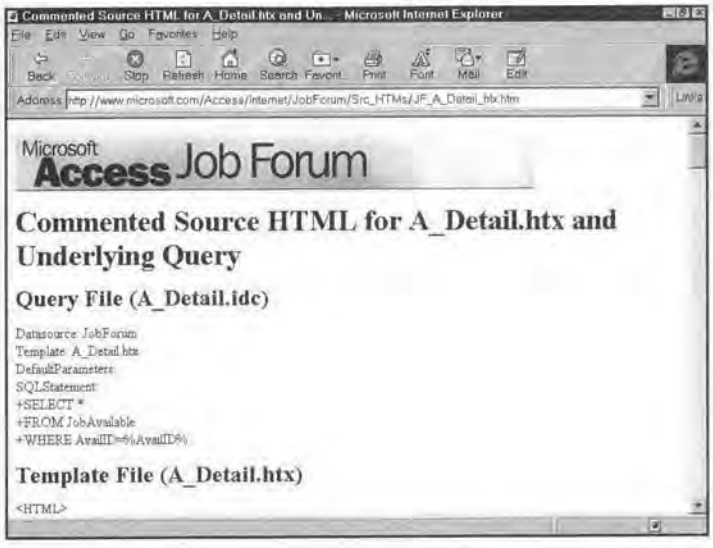


FIG. 18.8 The first part of the commented HTML source Web page for the Job Information page of Figure 18.7.

Exporting Datasheets to Dynamic Web Pages

Access 97's File, Save As/Export and File, Save As HTML menu choices let you save table or query datasheets to dynamic Web pages. The advantage of a dynamic Web page compared with the static Web pages described in the preceding chapter, "Exporting Data to World Wide Web Pages," is that a dynamic Web page displays up-to-date information. Using a dynamic Web page eliminates the need to periodically re-create Web pages.

> **Note**
>
> You can't create dynamic Web pages from Access reports. Like printed reports, you must re-create the Web page when the content of a report changes.

Creating a System ODBC Data Source by Using Access 97

Before you can use the .idc and .htx files, you must create a system ODBC data source on the Web server for the database that supplies the information for the table of the generated Web page. To create a system ODBC data source for Northwind.mdb, follow these steps:

Oracle Ex. 1009, p. 654
Oracle v. DeBellis
IPR2024-00221

1. On the PC running IIS 2+, open Northwind.mdb (or any other .mdb file) in Access 97. Select any table in the database.

> **Note**
>
> You can only export table and query objects to an ODBC database.

2. Choose File, Save As/Export. Select the To Another File or Database in the Save As dialog, and click OK to open the Save Table '*Tablename*' dialog.
3. Select ODBC Databases( ) in the Save as Type drop-down list (see Figure 18.9). When you make the selection, the Export dialog appears.



FIG. 18.9 Selecting an ODBC data source to start the ODBC data source creation process in Access.

4. Accept the default table name in the Export dialog (see Figure 18.10), and click OK to open the Select Data Source dialog.



FIG. 18.10 Specifying a temporary table name for the export process.

5. Click the Machine Data Source tab to display the user and system ODBC data sources presently installed on your computer (see Figure 18.11). The number of

Oracle Ex. 1009, p. 655
Oracle v. DeBellis
IPR2024-00221

data sources that appear depends on the software you've installed. All of the User data sources are installed by Office 97.

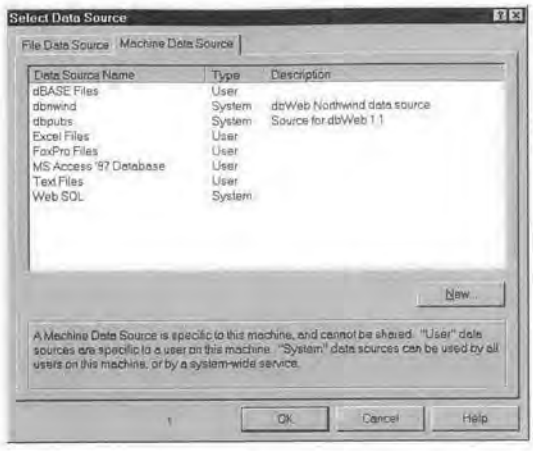


FIG. 18.11 Displaying existing system and user ODBC data sources.

6. Click the New button to display the first dialog of the Create New Data Source (Wizard). Microsoft doesn't include "Wizard" in the dialog's caption, but the dialog resembles that of other Access wizards. The New Data Source (Wizard) automates a multistep process. Select the System Data Source option (see Figure 18.12). Click Next to continue.

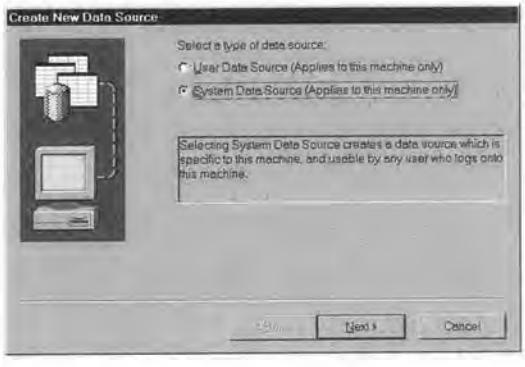


FIG. 18.12 Selecting the System Data Source option in the second Create New Data Source dialog.

7. Select the Microsoft Access Driver (*.mdb) in the list box (see Figure 18.13). Click Next.

Oracle Ex. 1009, p. 656
Oracle v. DeBellis
IPR2024-00221

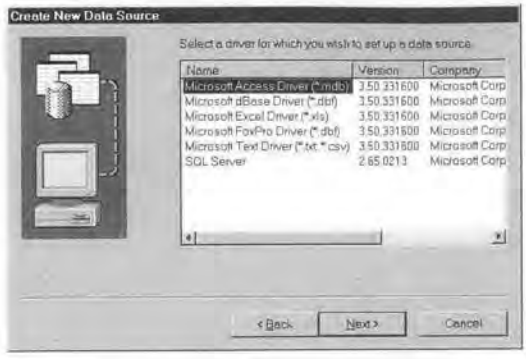


FIG. 18.13 Selecting the Microsoft Access Driver for the data source in the third Create New Data Source dialog.

8. The final (wizard) dialog confirms your prior choices (see Figure 18.14). Click Finish to open the ODBC Microsoft Access Setup dialog of the 32-bit ODBC Administrator application.



FIG. 18.14 Confirming your data source selections in the final Create New Data Source dialog.

9. Type a name, such as **Nwind Web**, in the Data Source Name text box, and add an optional description in the Description text box (see Figure 18.15). Click the Select button to open the Select Database dialog.

10. Navigate to the ...\Office\Samples folder, if necessary, and select Northwind.mdb as the Database Name (see Figure 18.16). Click OK to return to the ODBC Microsoft Access Setup dialog.

> **Note**
>
> If you want to use tables in a secure Jet database for the Web pages, select the Database option in the System Database frame, and click the System Database button. Select the workgroup file in the Select System Database dialog and click OK.

Oracle Ex. 1009, p. 657
Oracle v. DeBellis
IPR2024-00221

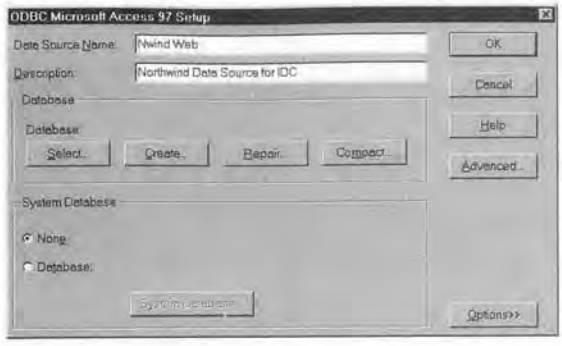


FIG. 18.15 Specifying a data source name and adding a description in the ODBC Microsoft Access Setup dialog.

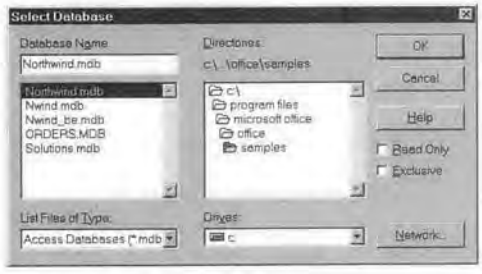


FIG. 18.16 Selecting the Jet 3.5 database for the ODBC data source.

11. Click OK to create the ODBC data source entry in the Registry. Your new data source appears as a System Data Source in the list of Machine Data Sources for the Web server (see Figure 18.17).

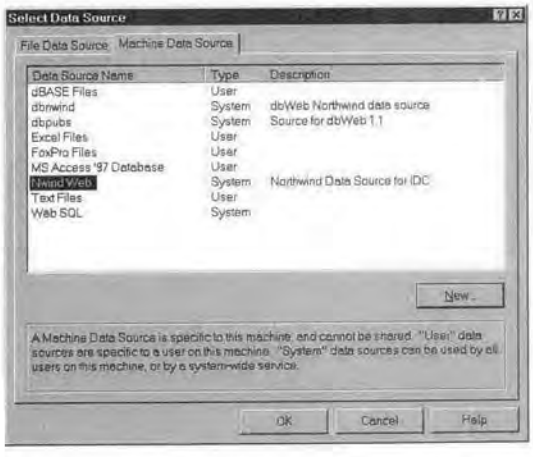


FIG. 18.17 The new data source added to the list of Machine Data Sources.

Oracle Ex. 1009, p. 658
Oracle v. DeBellis
IPR2024-00221

12. Click Cancel to stop the export process, because you only want to create the data source and don't want to export the table.

> **Note**
>
> It's important to specify a system ODBC data source, rather than a file or user data source, for ODBC data sources used with the IDC. System data sources are accessible to Httpodbc.dll immediately after the server boots up, without the need for a user to be logged on to the server. You must log on as the user who created the data source to make a user data source available to Httpodbc.dll.

Specifying a Default ODBC Data Source

The Hyperlink/HTML page of the Options dialog lets you set a default ODBC data source for the .idc files you create with Access export operations. To make the Nwind Web data source the default, follow these steps:

1. Choose Tools, Options to open the Options dialog and click the Hyperlink/HTML tab.
2. Type the name of your data source—**Nwind Web**, in this case—in the Data Source Name text box.
3. Type **Admin** (the default user name) or a valid user name for a secure Jet database in the User to Connect As text box.
4. If your Jet database is secure, type the password to use with the user name in the Password for User text box (see Figure 18.18). For the default Admin user, leave the Password for User text box empty.

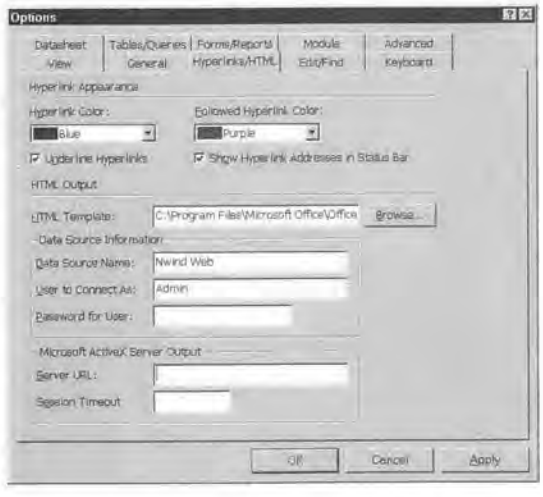


FIG. 18.18 Specifying the data source name, user name, and password (if required) for the default ODBC data source of the .idc files you export.

Oracle Ex. 1009, p. 659
Oracle v. DeBellis
IPR2024-00221

5. Click OK to save the default database information and close the Options dialog.

Using HTML Export to Create the Web Pages

You can use the File, Save As/Export command to create the .idc and .htx files for your dynamic Web page, but you need a home page (Default.htm) to execute the .idc file. Using the File, Save As HTML menu choice to activate the Publish to the World Wide Web Wizard (described in the preceding chapter) offers you the option of automatically creating a Switchboard form to execute the .idc file. The example that follows uses the North American Customers query (described in the preceding chapter) and Publish to the World Wide Web Wizard with Internet Information Server 2.0 running under Windows NT 4.0. You also can use the Microsoft Personal Web Server, included on the CD-ROM version of Office 97, for this example. If you use the Personal Web Server, only the paths to the folders in which you store the files differ.

◀◀ See "Creating a Web Page from a Query," p. 603
◀◀ See "Using the Web Publishing Wizard to Create a Static Web Site," p. 614



To create a dynamic Web page from the North American Customers query, follow these steps:

1. Create a new folder, \InetPub\wwwroot\NwindWeb, in which to create your Default.htm, .idc, and .htx files.
2. Create a second new folder, \InetPub\scripts\NwindWeb, to which you move the .idc and .htx files after the Publish to the World Wide Web Wizard creates the files in \InetPub\wwwroot\NwindWeb.

> **Note**
>
> All .idc and .htx files must be located in subfolders of the \InetPub\scripts folder. You cannot execute .idc files (or any other executable file) from subfolders of the \InetPub\wwwroot folder.

3. If you want to use a template for your Default.htm home page, copy the template files, including the required .gif or .jpg graphics files, to the \InetPub\wwwroot\NwindWeb folder. This example uses the 100.htm template, which requires 100.jpg and Msaccess.jpg. If you want to use the same or another template with the .htx file, copy the graphics files to \InetPub\scripts\NwindWeb.
4. Open Northwind.mdb in Access 97 and choose File, Save As HTML to start the Publish to the Web Wizard (see Figure 18.19). You create a new publication profile in this example, so leave the I Want to Use a Web Publication Profile check box unmarked. Click Next to display the second Wizard dialog.

Oracle Ex. 1009, p. 660
Oracle v. DeBellis
IPR2024-00221

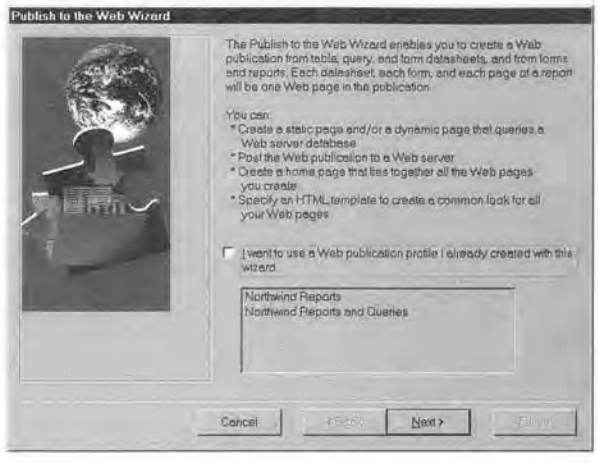


FIG. 18.19 The first dialog of the Publish to the Web Wizard.

5. Click the Queries tab and select the North American Customers query (see Figure 18.20). You can create .idc and .htx files for more than one query or combine tables and queries, but you can't create .idc and .htx files for Access reports. Click Next to display the third wizard dialog (see Figure 18.21).

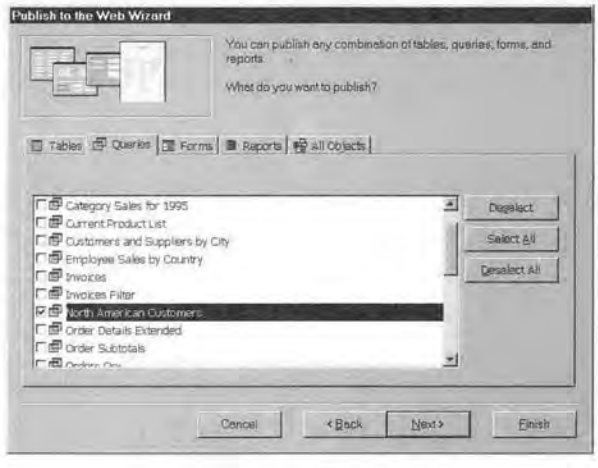


FIG. 18.20 Selecting the query for the Web page in the second wizard dialog.

Oracle Ex. 1009, p. 661
Oracle v. DeBellis
IPR2024-00221

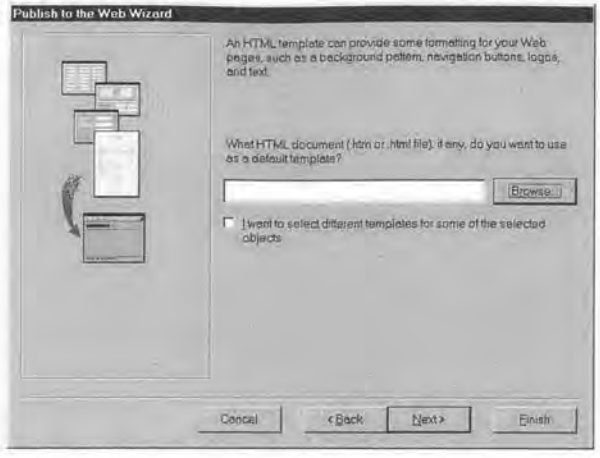


FIG. 18.21 The third wizard dialog that lets you specify an HTML template for your Default.htm home page.

6. Click the Browse button to open the Select a HTML Template dialog, navigate to the \InetPub\Wwwroot\NwindWeb folder, and select the template—100.htm for this example (see Figure 18.22). Click Next to display the fifth wizard dialog.



FIG. 18.22 Selecting a template for the Default.htm page.

7. Select the Dynamic HTX/IDC (Microsoft Internet Information Server) option to create .htx and .idc files (see Figure 18.23). Click Next to display the sixth wizard dialog.

Oracle Ex. 1009, p. 662
Oracle v. DeBellis
IPR2024-00221

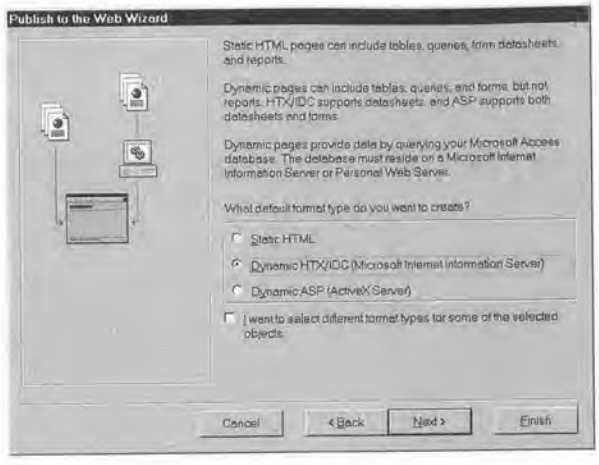


FIG. 18.23 Specifying a dynamic Web page created with the Internet Database Connector.

8. The default ODBC data source name you specified in the preceding section appears in the text boxes of the Data Source Information frame (see Figure 18.24). The Data Source Name value is required; User Name and Password entries are required only for secure Access databases for which you specify a workgroup file. Click Next.

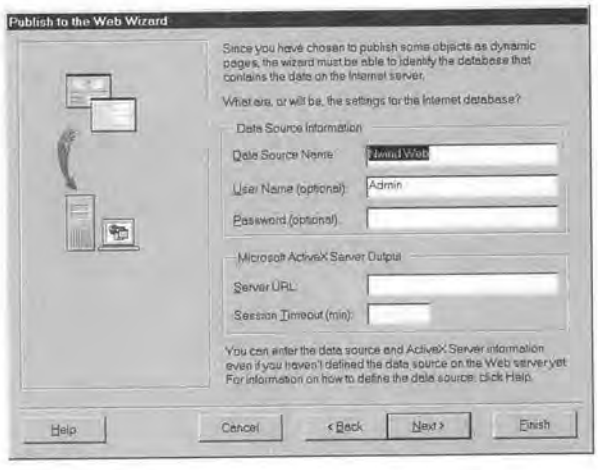


FIG. 18.24 Accepting the default ODBC data source entries from the data source specified in the Hyperlink/HTML page of the Access 97 Options dialog.

9. Type **\InetPub\wwwroot\NwindWeb** as the folder to store the files that the wizard creates (see Figure 18.25). Alternatively, click the Browse button to navigate to the correct folder. The .idc and .htx files move to the \InetPub\scripts\

Oracle Ex. 1009, p. 663
Oracle v. DeBellis
IPR2024-00221

NwindWeb folder later in this example. Accept the default No, I Only Want to Publish Objects Locally, assuming that you are creating the files on the PC that's running IIS 2+. Click Next.

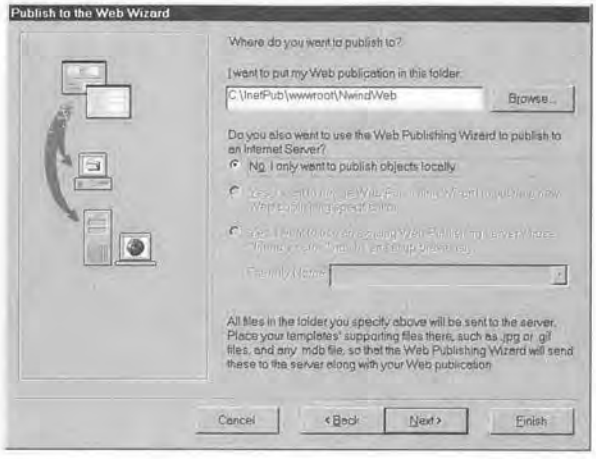


FIG. 18.25 Specifying the folder in which to store the files you create.

10. You need a home page to execute the .idc file, so mark the Yes, I Want to Create a Home Page check box and accept Default(.htm) as the name of the home page file (see Figure 18.26). Click Next.

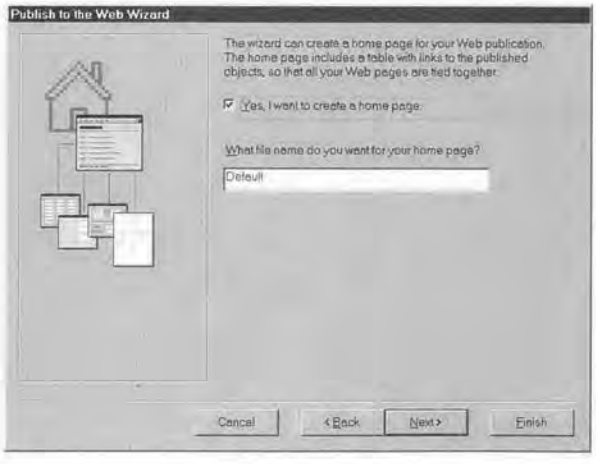


FIG. 18.26 Specifying a Switchboard home page to execute the .idc file.

Oracle Ex. 1009, p. 664
Oracle v. DeBellis
IPR2024-00221

11. In the final wizard page, mark the Yes, I Want to Save Wizard Answers to a Web Publication Profile check box, and type **NwindWeb Specification** in the Profile Name text box (see Figure 18.27). Click Finish to create the Web page files and terminate the wizard.

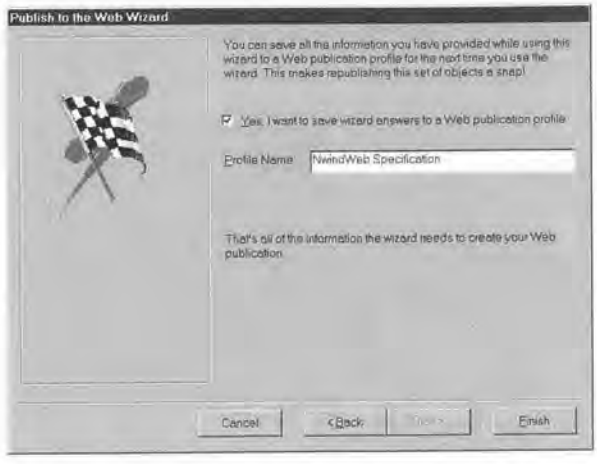


FIG. 18.27 Specifying a Web publication profile in the last wizard dialog.

After the wizard creates the files, you must take the following steps to test the dynamic Web page you designed:

1. Move the North American Customers_1.htx and North American Customers_1.idc files from \InetPub\Wwwroot\NwindWeb to \InetPub\scripts\NwindWeb.
2. Open in Notepad the Default.htm file in \InetPub\Wwwroot\NwindWeb. Find the `<A HREF="North American Customers_1.idc?">` tag, and change the link to `<A HREF="/scripts/nwindweb/North American Customers_1.idc?">` (see Figure 18.28). Close Notepad and save the changes to the file.

> **Note**
>
> The paths you specify for hyperlinks to other files on the Web server are relative to the main publishing directory, \InetPub for IIS 2.0.

3. Open IE 3.0 or your default browser and type the **http://*servername*/nwindweb** URL in the Address text box. The Default.htm Switchboard page appears (see Figure 18.29).

Oracle Ex. 1009, p. 665
Oracle v. DeBellis
IPR2024-00221

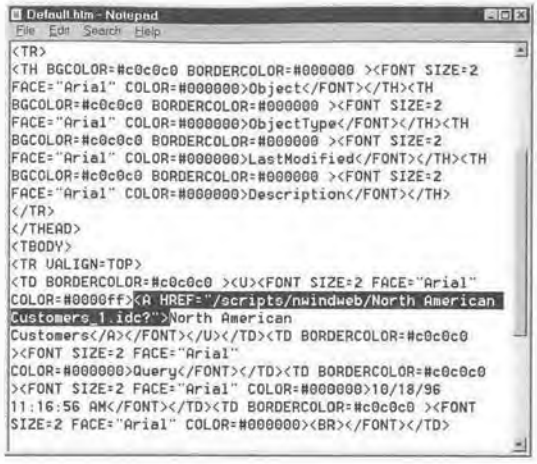


FIG. 18.28 Changing the hyperlink to the North American Customers_1.idc file in Default.htm.

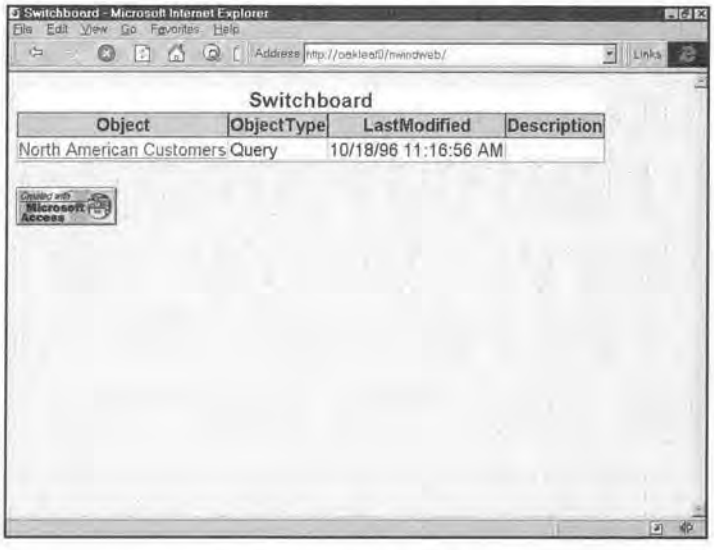


FIG. 18.29 The Switchboard page created by Default.htm with the 100.htm template.

4. Click the North American Customers hyperlink to cause North American Customers_1.idc to execute a SQL query against Northwind.mdb, to return the query result set to North American Customers_1.htx, and to display the dynamic Web page (see Figure 18.30).

Oracle Ex. 1009, p. 666
Oracle v. DeBellis
IPR2024-00221



North American Customers

Customer ID	Company Name	City	Region	Postal Code	Phone	Country
ANATR	Ana Trujillo Emparedados y helados	México D.F.		05021	(5) 555-4729	Mexico
ANTON	Antonio Moreno Taquería	México D.F.		05023	(5) 555-3932	Mexico
BOTTM	Bottom-Dollar Markets	Tsawassen	BC	T2F 8M4	(604) 555-4729	Canada
CENTC	Centro comercial Moctezuma	México D.F.		05022	(5) 555-3392	Mexico
GREAL	Great Lakes Food Market	Eugene	OR	97403	(503) 555-7555	USA
HUNGC	Hungry Coyote Import Store	Elgin	OR	97827	(503) 555-6874	USA
LAUGB	Laughing Bacchus Wine Cellars	Vancouver	BC	V3F 2K1	(604) 555-3392	Canada
LAZYK	Lazy K Kountry Store	Walla Walla	WA	99362	(509)	USA

FIG. 18.30 Internet Explorer 3.0 displaying the North American Customers dynamic Web page.

5. Open North American Customers_1.idc in Notepad to view the first half of the Publish to the Web Wizard's work (see Figure 18.31). The Wizard creates the `SQL Query` element of the file from the Jet SQL statement used to create the North American Customers query. (The field names of the query are a single line in the .idc file but appear on multiple lines in Figure 18.31 because of line wrapping.)

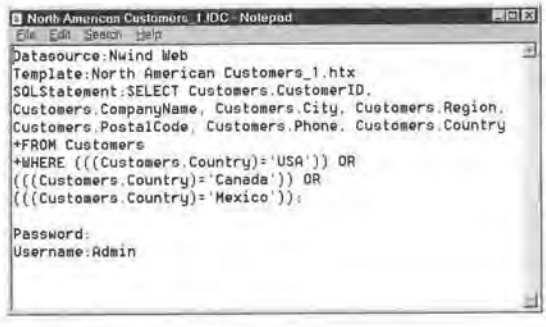


FIG. 18.31 Notepad displaying the content of the North American Customers_1.idc file.

Listing 18.5 shows the HTML source of the North American Customers_1.htx file called by the `Template:` element of North American Customers_1.idc.

Oracle Ex. 1009, p. 667
Oracle v. DeBellis
IPR2024-00221

Listing 18.5 The HTML Source of the North American Customers_1.htx Template File

```
<HTML>
<TITLE>North American Customers</TITLE>


TABLE BORDER=1 BGCOLOR=#ffffff CELLSPACING=0><FONT FACE="Arial"
OLOR=#000000><CAPTION><B>North American Customers</B></CAPTION>

THEAD>
TR>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
FONT SIZE=2 FACE="Arial" COLOR=#000000>Customer ID</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
FONT SIZE=2 FACE="Arial" COLOR=#000000>Company Name</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
FONT SIZE=2 FACE="Arial" COLOR=#000000>City</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
FONT SIZE=2 FACE="Arial" COLOR=#000000>Region</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
FONT SIZE=2 FACE="Arial" COLOR=#000000>Postal Code</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
FONT SIZE=2 FACE="Arial" COLOR=#000000>Phone</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 >
FONT SIZE=2 FACE="Arial" COLOR=#000000>Country</FONT></TH>
/TR>
/THEAD>

TBODY>
%BeginDetail%>
TR  VALIGN=TOP>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial"
OLOR=#000000><%CustomerID%><BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial" COLOR=#000000>
%CompanyName%><BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial" COLOR=#000000>
%City%><BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial" COLOR=#000000>
%Region%><BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial" COLOR=#000000>
%PostalCode%><BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial" COLOR=#000000>
%Phone%><BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial" COLOR=#000000>
%Country%><BR></FONT></TD>
/TR>
%EndDetail%>
/TBODY>
TFOOT></TFOOT>
/TABLE>
/BODY>
BR><BR>

/HTML>
```

Oracle Ex. 1009, p. 668
Oracle v. DeBellis
IPR2024-00221

Exporting Tables, Queries, and Forms to Active Server Pages

Active Server Pages (ASPs) play the primary role in Microsoft's strategy to gain for IE and IIS the predominate position in the Internet browser and server markets. ASP uses server-side scripting and *ActiveX Data Objects* (ADOs) to generate HTML files from desktop and client/server databases. ADO is an Automation wrapper (interface) for Microsoft's new OLE DB API, which first appeared in late 1996. OLE DB's objective is to provide a simpler mechanism than SQL-oriented ODBC for accessing data stored in any type of row/column format. ADO defines an Automation object, similar in concept to Access's ODBCDirect and Visual Basic's Remote Data Object (RDO). ADO supplies a `Recordset` object created from a table or query that you can manipulate with *Visual Basic Script* (*VBScript* or *VBS*) contained in .asp files stored on the Internet server. VBS is a simplified subset of VBA 5.0. Jet and ODBCDirect `Recordset`, and RDO's `rdoResultset` objects require the full version of VBA 5.0.



▶▶ See "Using ODBCDirect and the Remote Data Object," p. 1044

When this book was written, the only OLE DB data provider was Microsoft's "OLE DB Over ODBC," formerly code-named *Kagera*. You can expect Microsoft and third parties to deliver a variety of OLE DB data providers in 1997 and beyond. Until an OLE DB data provider is available for Jet 3.5 .mdb files, you must use an ODBC data source with OLE DB. The Nwind Web data source, described in the section "Creating a System ODBC Data Source Using Access 97" earlier in this chapter, is used for the ASP examples that follow.

Creating \InetPub Subfolders for ASP Files

When an Internet client requests an .asp file from the server, the code in the .asp file executes to create an .htm(l) file that that the browser can display. Thus, the folders that contain .asp files require Execute permission. By default, subfolders of the \InetPub\Wwwroot folder do not have Execute permission. To create an IIS 3.0 \InetPub subfolder in which to store the .asp files you create in the following sections, perform these steps:

1. Use Windows Explorer to create a subfolder of \InetPub, \InetPub\NwindASP, for the examples that follow. If you are using IIS 3.0 and NTFS, make sure that the anonymous IUSER_*SERVERNAME* user has at least Read, Write, and Execute permissions for the folder.
2. Choose Start, Programs, Microsoft Internet Server, Internet Service Manager to launch Internet Service Manager.
3. Double-click the Computer entry for the WWW Service to open the WWW Service Properties for *Servername* properties sheet.

Oracle Ex. 1009, p. 669
Oracle v. DeBellis
IPR2024-00221

4. Click the Directories tab and click the Add button to open the Directory Properties dialog. Click the Browse button to open the Select Directory dialog and navigate to the \InetPub\NwindASP folder (see Figure 18.32). Click OK to enter the folder name in the Directory text box.

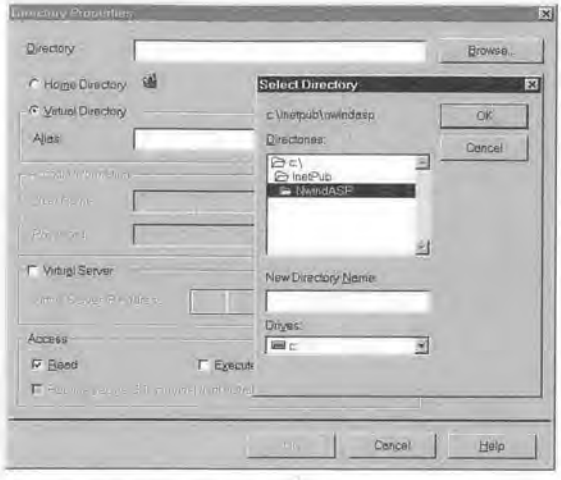


FIG. 18.32 Making the \InetPub\NwindASP folder accessible to Internet or intranet users.

5. Select the Virtual Directory option and type a name for the folder, **Northwind** for the following examples, in the Alias text box. The URL for the folder is **http://*servername*/northwind**; OAKLEAF3 is the name of the server used to create the following figures.

▶▶ See "Defining the Client/Server Environment," p. 915



6. Mark the Execute check box to give users permission to execute files contained in this folder (see Figure 18.33). Click OK to add your new folder to IIS 3.0 and close the Directory Properties dialog.

Caution

If you don't enable Execute permissions, .asp files fail to deliver .htm(l) files to the browser. Depending on the browser you use and your location on the network, you might see the source code for the .asp file or an empty browser window.

Oracle Ex. 1009, p. 670
Oracle v. DeBellis
IPR2024-00221

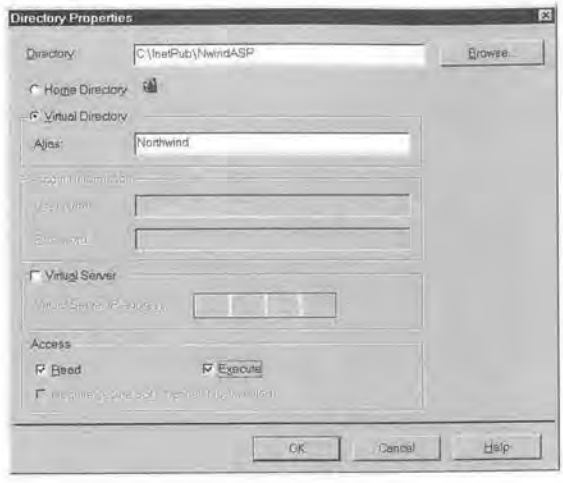


FIG. 18.33 Specifying a virtual directory alias and enabling execution of .asp files in the folder.

7. Select the C:\InetPub\NwindASP Directory item, mark the Enable Default Document check box, and modify the default entry to Default.html (see Figure 18.34). The Switchboard page created by the Publish to the Web Wizard uses the .html file extension.

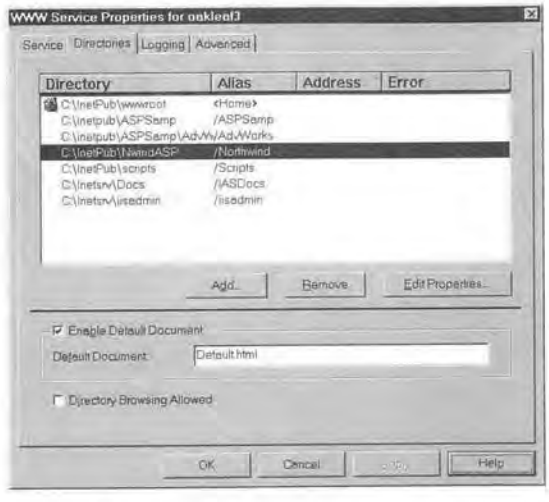


FIG. 18.34 Specifying a default document for the \InetPub\NwindASP folder.

8. Click the OK button to close the WWW Service Properties for *Servername* properties sheet, and then close the Internet Service Manager window.

9. If you want to use a template with your .asp files, copy the template files to the \InetPub\NwindASP folder. The examples that follow use the 100.htm, 100.jpg, and Msaccess.jpg files for the template.

Oracle Ex. 1009, p. 671
Oracle v. DeBellis
IPR2024-00221

◀◀ See "Using HTML Templates," p. 606



Exporting a Query to an ASP File

Creating a single .asp file is considerably simpler than using .idc/.htx files to create dynamic Web pages that display tabular data. You can export complete tables or a query result set to a single .asp file, which generates the .htm file from the database specified in the ODBC data source. The following example uses the File, Save As/Export method to create the ASP version of the IDC example described in the section "Using HTML Export to Create the Web Pages" earlier in this chapter.

◀◀ See "Creating a Web Page from a Query," p. 603



To create the North American Customers.asp file, follow these steps:

1. Open Northwind.mdb, if necessary, and select the North American Customers query that you created in the preceding chapter, "Exporting Data to World Wide Web Pages."
2. Choose File, Save As/Export to open the Save As dialog. Select the To an External File or Database option and click OK to open the Save Query 'North American Customers' In dialog.
3. Navigate to the \InetPub\NwindASP folder, and then select Microsoft ActiveX Server (*.asp) from the Save as Type list (see Figure 18.35). Click Export to open the Microsoft ActiveX Server Pages Output Options dialog.

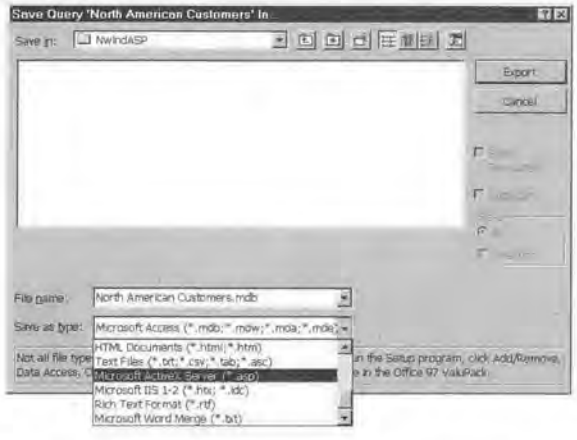


FIG. 18.35 Saving the North American Customers query in ASP format.

Oracle Ex. 1009, p. 672
Oracle v. DeBellis
IPR2024-00221

4. If you added a template file to your \InetPub\NwindASP folder in the last step of the preceding section, type the file name of the template in the HTML Template text box. Type **Nwind Web** in the Data Source Name text box and **http://*servername*/northwind** in the Server URL text box (see Figure 18.36).

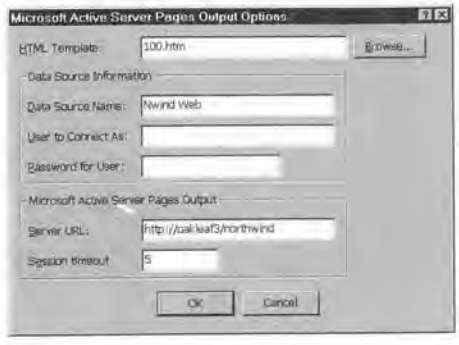
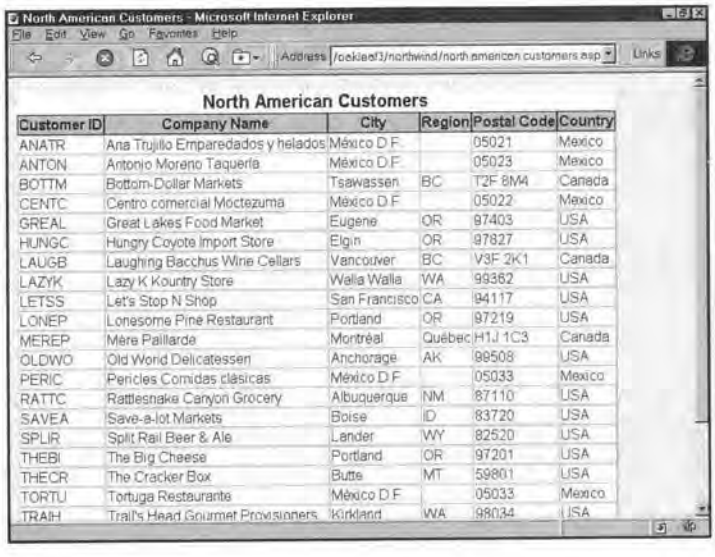


FIG. 18.36 Specifying the template file, ODBC data source, and URL for the North American Customers.asp file.

5. Click OK to export the file to North American Customers.asp and close the dialog.

6. Launch IE 3+ and type **http://*servername*/North American Customers.asp** in the Address text box to execute the .asp file. The resulting HTML page appears as shown in Figure 18.37.



North American Customers

Customer ID	Company Name	City	Region	Postal Code	Country
ANATR	Ana Trujillo Emparedados y helados	México D.F.		05021	Mexico
ANTON	Antonio Moreno Taquería	México D.F.		05023	Mexico
BOTTM	Bottom-Dollar Markets	Tsawassen	BC	T2F 8M4	Canada
CENTC	Centro comercial Moctezuma	México D.F.		05022	Mexico
GREAL	Great Lakes Food Market	Eugene	OR	97403	USA
HUNGC	Hungry Coyote Import Store	Elgin	OR	97827	USA
LAUGB	Laughing Bacchus Wine Cellars	Vancouver	BC	V3F 2K1	Canada
LAZYK	Lazy K Kountry Store	Walla Walla	WA	99362	USA
LETSS	Let's Stop N Shop	San Francisco	CA	94117	USA
LONEP	Lonesome Pine Restaurant	Portland	OR	97219	USA
MEREP	Mère Paillarde	Montréal	Québec	H1J 1C3	Canada
OLDWO	Old World Delicatessen	Anchorage	AK	99508	USA
PERIC	Pericles Comidas clásicas	México D.F.		05033	Mexico
RATTC	Rattlesnake Canyon Grocery	Albuquerque	NM	87110	USA
SAVEA	Save-a-lot Markets	Boise	ID	83720	USA
SPLIR	Split Rail Beer & Ale	Lander	WY	82520	USA
THEBI	The Big Cheese	Portland	OR	97201	USA
THECR	The Cracker Box	Butte	MT	59801	USA
TORTU	Tortuga Restaurante	México D.F.		05033	Mexico
TRAIH	Trail's Head Gourmet Provisioners	Kirkland	WA	98034	USA

FIG. 18.37 The Web page generated by North American Customers.asp.

Oracle Ex. 1009, p. 673
Oracle v. DeBellis
IPR2024-00221

Internet Explorer 3.01 is the browser used to test the .asp pages created in these examples. IE 3.01 provides a number of minor improvements to the original IE 3.0 version. You can download the current version of IE from:

On the Web

http://www.microsoft.com/ie/

7. Choose View, Source to display in Notepad the HTML source generated by the .asp file (see Figure 18.38). The HTML source is essentially identical to that created by the .idc/.htx files you created earlier in the chapter.

   Spaces ordinarily aren't allowed in URLs, so ASP replaces spaces in generated HTML file names with %20.

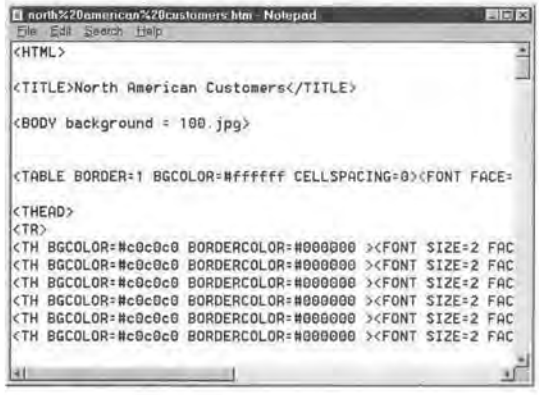


FIG. 18.38 Part of the HTML source for the Web page generated by North American Customers.asp.

8. Use Notepad to open the North American Customers.asp file to display the VBS source code used to create the .htm file (see Figure 18.39).

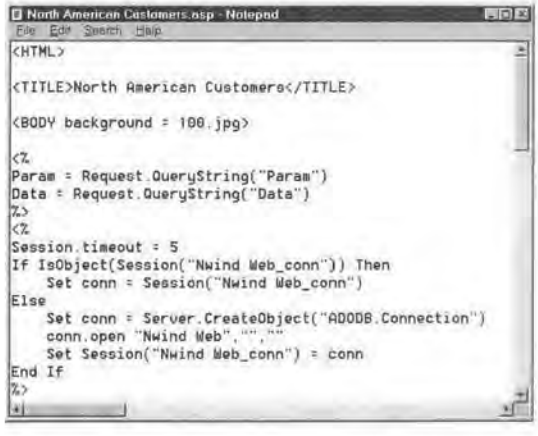


FIG. 18.39 Part of the VBS source code contained in North American Customers.asp.

Oracle Ex. 1009, p. 674
Oracle v. DeBellis
IPR2024-00221

Note

The reason for using a fully descriptive query name is to create an appropriate title for the query result set table. You can rename the North American Customers.asp file to NACusts.asp or the like to minimize the length of the URL and avoid including spaces.

Listing 18.6 contains the complete source, including VBS code, of North American Customers.asp. Server-side VBS code is enclosed within <%. . .%> tags. The following three VBS statements establish a connection to Northwind.mdb using the ActiveX Data Object (ADODB) and the Nwind Web data source:

```
Set conn = Server.CreateObject("ADODB.Connection")
conn.open "Nwind Web","",""
Set Session("Nwind Web_conn") = conn
```

The SQL statement in the `sql = ...` lines is the Jet SQL statement of the North American Customers query that opens an ADO `Recordset` object with the `rs.Open sql, conn, 3, 3` VBS statement. The VBA code between the `Do While Not rs.EOF` and `Loop` statements populates the cells of the table.

Listing 18.6 The HTML Source and VBS Code of the North American Customers.asp File

```
<HTML>
<TITLE>North American Customers</TITLE>
<BODY background = 100.jpg>


%
aram = Request.QueryString("Param")
ata = Request.QueryString("Data")
>
%
ession.timeout = 5
f IsObject(Session("Nwind Web_conn")) Then
  Set conn = Session("Nwind Web_conn")
lse
  Set conn = Server.CreateObject("ADODB.Connection")
  conn.Open "Nwind Web","",""
  Set Session("Nwind Web_conn") = conn
End If
%>
<%
    sql = "SELECT Customers.CustomerID, Customers.CompanyName,
Customers.City, Customers.Region, Customers.PostalCode,
Customers.Country FROM Customers WHERE (((Customers.Country)='USA'))
OR (((Customers.Country)='Canada')) OR
(((Customers.Country)='Mexico'))"
    If cstr(Param) <> "" And cstr(Data) <> "" Then
        sql = sql & " And " & cstr(Param) & " = " & cstr(Data)
    End If
    Set rs = Server.CreateObject("ADODB.Recordset")
    rs.Open sql, conn, 3, 3
%>
```


Oracle Ex. 1009, p. 675
Oracle v. DeBellis
IPR2024-00221

```
<TABLE BORDER=1 BGCOLOR=#ffffff CELLSPACING=0><FONT FACE="Arial"
COLOR=#000000><CAPTION><B>North American Customers</B></CAPTION>

THEAD>
TR>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 ><FONT SIZE=2
FACE="Arial" COLOR=#000000>Customer ID</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 ><FONT SIZE=2
FACE="Arial" COLOR=#000000>Company Name</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 ><FONT SIZE=2
FACE="Arial" COLOR=#000000>City</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 ><FONT SIZE=2
FACE="Arial" COLOR=#000000>Region</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 ><FONT SIZE=2
FACE="Arial" COLOR=#000000>Postal Code</FONT></TH>
TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 ><FONT SIZE=2
FACE="Arial" COLOR=#000000>Country</FONT></TH>

/TR>
/THEAD>
TBODY>

%
n Error Resume Next
s.MoveFirst
o While Not rs.EOF
% >

TR VALIGN=TOP>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial"
OLOR=#000000><%=Server.HTMLEncode(rs.Fields("CustomerID").Value)%>
BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial"
OLOR=#000000><%=Server.HTMLEncode(rs.Fields("CompanyName").Value)%>
BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial"
OLOR=#000000><%=Server.HTMLEncode(rs.Fields("City").Value)%>
BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial"
OLOR=#000000><%=Server.HTMLEncode(rs.Fields("Region").Value)%>
BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial"
OLOR=#000000><%=Server.HTMLEncode(rs.Fields("PostalCode").Value)%>
BR></FONT></TD>
TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial"
OLOR=#000000><%=Server.HTMLEncode(rs.Fields("Country").Value)%>
BR></FONT></TD>
/TR>

<%
rs.MoveNext
Loop
%>
```

(continues)

Oracle Ex. 1009, p. 676
Oracle v. DeBellis
IPR2024-00221

Listing 18.6 Continued

```
</TBODY>
<TFOOT></TFOOT>
</TABLE>
</BODY>
<BR><BR>
<IMG SRC = "msaccess.jpg">
</HTML>
```

If you're familiar with VBA, the similarities between server-side VBS and VBA code in the preceding listing are quite evident. Using .asp files to create dynamic Web pages from tables and queries saves a substantial amount of server disk space compared with exporting a corresponding static Web page.

Exporting Access Forms to ASP Files

Access 97 lets you export existing forms to .asp files. The form export process creates the necessary HTML text boxes and navigation buttons to emulate simple Access forms, including forms with subforms. IE 3+ uses the ActiveX HTML Layout Control to create the page design in a *Formname*alx.asp file; subforms are implemented by an embedded Web browser control. Both *Formname*.asp and *Formname*alx.asp files are required to emulate Access forms.

To create Customers.asp and Customersalx.asp and test the Web-based form, follow these steps:

1. Select the Customers form and choose File, Save As/Export to open the Save As dialog. Make sure that the To an External File or Database option is selected and click OK to open the Save Form 'Customers' In dialog.
2. Navigate to the \InetPub\NwindASP folder, and then select Microsoft ActiveX Server (*.asp) from the Save as Type list (refer to Figure 18.35). Click Export to open the Microsoft ActiveX Server Pages Output Options dialog.
3. The default background color of the exported form is gray, the same as the default background color of most browsers, so don't specify a template file. If necessary, type **Nwind Web** in the Data Source Name Text box and type **http://*servername*/northwind** in the Server URL text box (refer to Figure 18.36).

   If you complete this example immediately after completing the example of the prior section, the text boxes of the Microsoft ActiveX Server Pages output Options dialog retain their prior settings.
4. Click OK to export the file to Customers.asp and Customersalx.asp and close the dialog.
5. Launch IE 3+ and type **http://*servername*/Customers.asp** in the Address text box to execute the two .asp files. The resulting HTML page appears as shown in Figure 18.40.

Oracle Ex. 1009, p. 677
Oracle v. DeBellis
IPR2024-00221

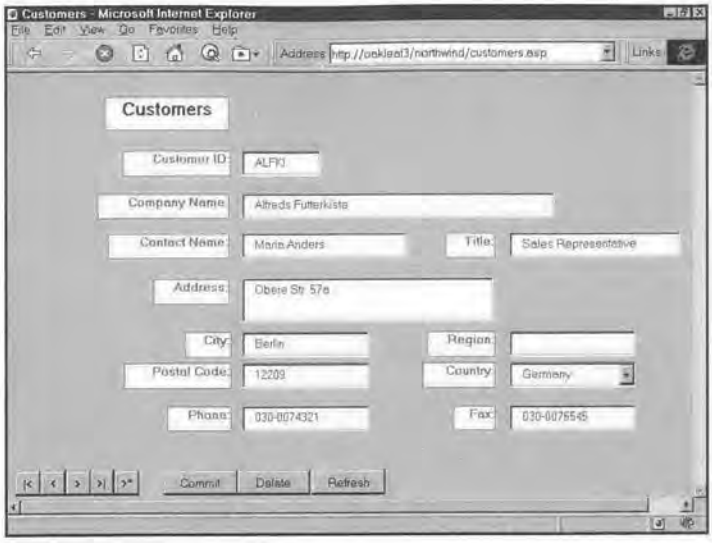


FIG. 18.40 The Web page generated by Customers.asp and Customersalx.asp.

6. Choose View, Source to display in Notepad the HTML source generated by the Customers.asp file (see Figure 18.41). Very little source is required in the Customers(1).htm file because all the action takes place in the ActiveX HTML Layout Control of the Customersalx.asp file identified by the `OBJECT ID=` and `PARAM NAME=` tags.



FIG. 18.41 The Web page generated by Customers.asp and Customersalx.asp.

You can edit any of the entries that appear in text boxes and select a country from the drop-down list. Clicking the Commit button sends changes you make to the server, which performs an `UPDATE` operation on the affected table(s)—only Customers, in this case. Each time you move the record pointer with the navigation buttons, commit changes, or delete or refresh a record, the server resends the modified page to the client. Thus, browsing a table with a Web form is a relatively slow process, even if you're

Oracle Ex. 1009, p. 678
Oracle v. DeBellis
IPR2024-00221

connected to the server via a local-area network (LAN). The ActiveX Database Connector (ADC), which wasn't released by Microsoft when this book was written, lets you store "disconnected" ADO `Recordset` objects on the client, browse and make multiple updates to the `Recordset`, and then return the changes to the server. ADC eliminates the need for a round-trip to the server each time you update, delete, or add a new record.

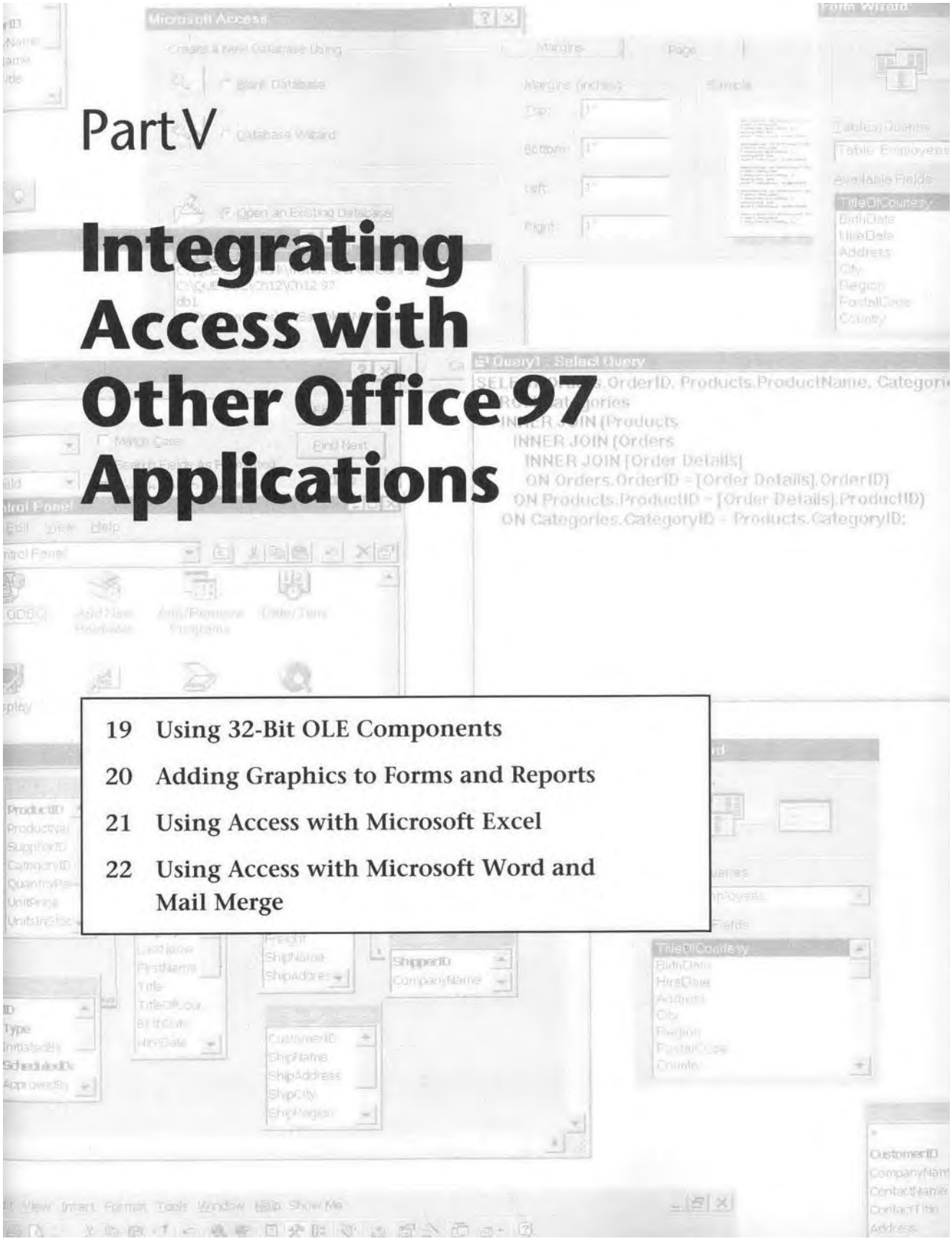
Note

Changing the value of the primary key field of a table on which other records rely violates referential integrity rules. If referential integrity is enforced by Access relationships, changes you make to primary key values (such as to the CustomerID field of the Customers table) are ignored. You don't receive an error message, because the VBS code in Customersalx.asp includes an `On Error Resume Next` statement.



▶▶ See "Handling Run-Time Errors," p. 1005

Oracle Ex. 1009, p. 679
Oracle v. DeBellis
IPR2024-00221



Part V

Integrating Access with Other Office 97 Applications

19 Using 32-Bit OLE Components

20 Adding Graphics to Forms and Reports

21 Using Access with Microsoft Excel

22 Using Access with Microsoft Word and Mail Merge

Oracle Ex. 1009, p. 680
Oracle v. DeBellis
IPR2024-00221

Chapter 19

Using 32-Bit OLE Components

Windows 95 and Windows NT 3.51 introduced 32-bit *Object Linking and Embedding* (OLE, pronounced as the Spanish *¡olé!*) 2.1 to the PC. The new version 8.0 members of the Microsoft Office 97 suite—Access 97 (in the Professional Edition only), Excel 97, Word 97, and PowerPoint 97—provide a variety of new capabilities with OLE 2.1. OLE is a method of transferring information in the form of *objects* between different Windows applications. This method is similar in concept to, although more sophisticated than, copying text or graphics to the Clipboard and pasting the copied text or graphic to other applications.

OLE 2.+ is based on Microsoft's *Component Object Model* (COM), which is the industry standard for object-oriented application programming in the Windows environment. This chapter introduces you to the principles of OLE and how your Access 97 applications can take advantage of the features offered by 32-bit OLE 2.1. The new ActiveX components included in Office 97 are also a form of OLE objects; ActiveX components are described in Chapter 31, "Exchanging Data with Automation and ActiveX Controls."

Understanding the Importance of OLE

Understanding OLE principles is important because OLE is the sole method by which you can add graphic images to your Access forms and reports, as well as add or edit data in OLE Object fields of Access. The benefits of using OLE for applying and storing nontext information (instead of using built-in graphics processing offered by some RDBMSs) include the following:

- You can use any image-processing application that functions as an OLE server to create and edit bitmapped graphics: from the simple Windows 95/NT 4.0 Paint application to photographic-quality editors, such as Adobe Photoshop and Corel PhotoPAINT.

Oracle Ex. 1009, p. 681
Oracle v. DeBellis
IPR2024-00221

- You can embed or link vector-based images from template-based OLE servers, such as Visio 4.0, or from professional illustration packages, such as CorelDRAW! 6.0—both 32-bit OLE 2.1 servers that are designed for Windows 95. Access 97 also supports 16-bit OLE 2.0 servers. Microsoft Graph 97, called MSGraph8 in this book, is included with Access 97; MSGraph8 is a 32-bit implementation of the 16-bit OLE 2.0 server applet included with Access 2.0 and Access 95.
- The added overhead that is associated with bitmap editors incorporated in the application is eliminated. Self-contained bitmap editors and drawing functions are seldom as capable as stand-alone, shrink-wrapped OLE server applications.
- You don't need to install a collection of import and export filters for different kinds of files. OLE server applications provide file import from, and export to, a variety of file types.
- You can export objects stored in OLE Object fields in Access tables or within bound or unbound object frames to other applications via the Clipboard. Bound object frames display the presentation of OLE objects stored in OLE Object fields of Access tables. Unbound object frames display the presentation of static OLE objects, such as company logos used to embellish forms and reports.
- You can store a variety of OLE objects in one OLE Object field. You can link or embed in one OLE Object field waveform audio (.wav), MIDI music (.mid), animation (.fli and .mmm), and audio/video interleaved (.avi) files. You need a large-capacity fixed disk to embed .avi and long-duration .wav files, however.
- You can choose between embedding the data within a table, form, or report and linking the OLE object to a file that contains the data. The behavior of an OLE 2.1 object differs, depending on whether you link or embed an object.

The most common kinds of OLE objects used in Access applications are 16-color and 256-color embedded bitmap graphic images. Figure 19.1 shows a portion of a page from the Northwind Traders Catalog report that includes an embedded bitmap image. The image is a Windows 95 Paint bitmap that is embedded by OLE in the Picture field of the Categories table.

Defining OLE

OLE is a member of a class of computer operations known as *interprocess communication (IPC)*. IPC operations enable different applications to send data to and receive data from other applications by an agreed-on procedure. The Clipboard is the primary IPC path for Windows; most present-day communication of data between running applications (other than by reading or writing to disk files) involves using the Windows Clipboard.

Windows defines a set of standard data types that you can copy to, or paste from, the Clipboard. OLE uses these standard Windows data types: bitmapped and vector-based graphic images, plain and formatted text, digital audio sound, and so on. You may have

Oracle Ex. 1009, p. 682
Oracle v. DeBellis
IPR2024-00221

used (or tried to use) dynamic data exchange (DDE) as an IPC method to transfer data between Windows applications. OLE is a major improvement over DDE because OLE is easier to implement than DDE. OLE 2.1+ ultimately is expected to replace DDE.



FIG. 19.1 A part of one page of the Northwind Traders Catalog report in Print Preview mode.

OLE operations differ from conventional Windows copy-and-paste operations—performed with Ctrl+C and Ctrl+V or by way of DDE—because OLE includes a substantial amount of information about the source of the data along with the actual data. An OLE object copied to the Clipboard by Excel 97, for example, includes the following information:

- The name of the application from which the data originated (in this case, Excel).
- The type of data, such as worksheet, module, or chart (Excel worksheet).
- The full path to the file, beginning with the drive letter and the file name, if the data is derived from or was saved to a file.
- A name assigned to the sheet or chart that contains the data if the data isn't derived from or saved to a file. The name usually is a long combination of numbers and letters.
- The name or coordinates of the range of the data if only a portion of an object is included.
- The presentation of the object in Windows Metafile Format (.wmf). If the object is not an image, the icon of the application that created the object is the object's presentation.
- For file-based OLE 2.1 objects, a set of property values obtained from entries in the property pages of the Properties window for the file (see Figure 19.2).

Oracle Ex. 1009, p. 683
Oracle v. DeBellis
IPR2024-00221

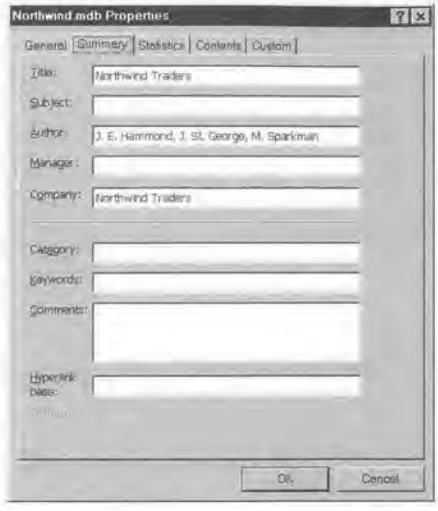


FIG. 19.2 The Summary page of the Properties window for Northwind.mdb.

> **Note**
>
> All file-based OLE 2.1-compliant server applications that qualify for the "Designed for Windows 95" logo have a File, Properties menu command that opens a standard Properties page for the file. You can edit the property values of the Summary and Custom pages; values in the General, Statistics, and Contents pages are read-only. Only documents created by OLE 2.1 servers appear as choices of the Start, Documents menu.

When you paste data copied to the Clipboard from an Excel worksheet and then double-click the cell's surface, Excel pops up to allow you to edit the data, and it disappears when you finish. With OLE, you paste complete objects, rather than just data, into an element of the application.

Classifying OLE 2.1-Compliant Applications

32-bit OLE 2.1-compliant applications fall into the following six categories:

- *Container applications*. These applications, such as Access 97, are stand-alone products that are capable of linking or embedding objects created by other 16-bit and 32-bit OLE 2+-compliant applications. Container applications, however, cannot create OLE objects for linking to, or embedding in, other OLE 2.1 applications. OLE container applications also are called *OLE client applications*.
- *Server-only applications*. These stand-alone applications create OLE objects for embedding in and linking to other applications, but you cannot embed an OLE object in or link an OLE object to them.

Oracle Ex. 1009, p. 684
Oracle v. DeBellis
IPR2024-00221

- *Mini-server applications*. Mini-servers (also called *OLE applets*) are similar to server-only applications, but mini-servers are not stand-alone products. You can execute an OLE mini-server only from within an OLE container application. MSGraph8 is an OLE 2.1 mini-server. Mini-servers also are called *components* or *ActiveX components*.
- *Full-server applications*. OLE full-servers are stand-alone applications that are capable of creating OLE objects for use by other applications. Full-server applications can embed or link objects created by other OLE servers. Excel 97, Word 97, and Project 97 are OLE 2.1 full-servers.
- *Automation-compliant applications*. Automation (formerly known as *OLE Automation*) lets an OLE 2+ container application manipulate an embedded or linked object by sending programming instructions to the OLE 2+ server that created it. Automation server applications are said to create *programmable objects*. Access 97 is an Automation server but not a conventional OLE 2.1 server, because you cannot embed Access 97 objects in 32-bit container applications. Microsoft Outlook also is an Automation server. Excel 97, Word 97, and Project 97 are all Automation-compliant container and server applications. Automation client applications, such as Access 97, use their programming language to send instructions to programmable objects. Using Automation is the subject of Chapter 31, "Exchanging Data with Automation and ActiveX Controls."

▶▶ See "Adding ActiveX Controls to Your Application," p. 1122



- *ActiveX Controls*. One of the advantages of Visual Basic as a Windows programming environment has been its extensibility through custom controls, originally called VBXs (Visual Basic eXtensions). Microsoft and third-party software publishers supply a variety of specialized custom control objects for Visual Basic 3.0 and earlier, including mini-spreadsheets, image editors, and report writers. In 1995, Microsoft created an OLE-based replacement for VBXs, called *OLE Controls*. Access 2.0 was the first product to offer extensibility through 16-bit OLE Controls. The 16-bit version of Visual Basic 4.0 accommodates both 16-bit OLE Controls and VBXs. Access 95 and 32-bit Visual Basic 4.0 use 32-bit OLE Controls.

  The file size of OLE Controls was much larger than their VBX counterparts because of the overhead needed to support the OLE 2.x specifications. To make OLE Controls practical for use in Internet applications, Microsoft developed 32-bit ActiveX Controls based directly on COM. ActiveX Controls are a lightweight version of the 32-bit OLE Controls they replace. Office 97 components (except Outlook) and Visual Basic 5.0 can embed ActiveX Controls in forms. The original OLE Control design is obsolete, but Access 97 is backwardly compatible with most of the 32-bit OLE Controls supplied with the Access 95 Developer's Toolkit and Visual Basic 4.0.

Oracle Ex. 1009, p. 685
Oracle v. DeBellis
IPR2024-00221

Note

Server-only, mini-server, and full-server OLE applications collectively are called *local servers*, because these OLE servers must reside on the same PC as the OLE client application. The Professional Edition of Visual Basic 4+ lets you create *Remote Automation Objects* (RAOs), a special type of OLE server that can reside on a network server and communicate with OLE client applications on networked workstations using *Distributed COM* (*DCOM*). RAOs let you build three-tier client/server applications with Access 97. You also can create *in-process* OLE servers by using Visual Basic 4+. These special types of OLE servers are discussed briefly in Chapter 31.

Introducing Object Building Blocks

When the term *object* is used in conjunction with a computer application or programming language, the term doesn't refer to a tangible object, such as a rock, saxophone, or book. Objects in computer programming and applications are intangible representations (called *abstractions*) of real-world objects.

A computer object combines *properties*—such as the properties you assign to an Access control object in a form or report—and *methods* that define the behavior of the object. Text box properties (such as the text the box contains, the size, the typeface, the font the text uses, and the colors and borders employed) vary widely. The ways text box controls behave when you type new text or edit existing text are the methods associated with text box objects of forms and reports. All text boxes in Windows applications use similar, but not identical, methods.

Properties and Methods Encapsulated in Objects

A musical instrument, such as a saxophone, can serve as an example of a tangible object with an intangible representation: sound. Some properties of a saxophone are size (soprano, alto, tenor, bass, and baritone), type of fingering, materials of construction, and the name of the instrument's manufacturer.

The methods that are applicable to saxophones are the techniques used to play these instruments: blowing into the mouthpiece, biting the reed, and fingering the keys that determine the note that you play.

If you create a programming object that simulates all of the properties and methods of a particular kind of saxophone, and if you have the proper audio hardware for a computer, you can create an object that imitates the sound of Charlie Parker's, Art Pepper's, or Stan Getz's style. Stanford University's new WaveGuide acoustic synthesis computer programs create objects that are capable of this kind of imitation. The properties of a particular saxophone and the methods of the artist playing it are said to be *encapsulated* in a particular kind of saxophone object. The object acts as a container for the properties and methods that constitute the object.

One more item is found in the container of an OLE object: *presentation*. Presentation is how the user perceives the object—how the object looks or sounds. The presentation of a WaveGuide saxophone is a musical sound. If the saxophone object is used in a Windows

Oracle Ex. 1009, p. 686
Oracle v. DeBellis
IPR2024-00221

95 application, the sound probably is reproduced through digital audio techniques by an audio adapter card. The presentation of a large graphic image may be a miniature copy, called a *thumbnail*.

At first glance, presentation may appear to be a property. Presentation isn't a property in the true sense, however, because the presentation of an object is dependent on factors outside the object, such as the kind of hardware available, the computer operating system used, and the application in which the object is employed. The presentation of a 256-color bitmap on a 16-color VGA display, for example, is quite different from its presentation on a 256-color display driver.

Object-Enabled Applications

Applications and programming languages—which are also applications—that programmers used to create these applications are *object-enabled* if the applications can encapsulate properties and methods within an object container. Before object-enabled programming was developed, programmers considered properties and methods as two separate entities. Programmers wrote code that defined the methods of an application. Separate data files contained the properties that the programmer's application manipulated. A classic example of this separation is xBase: a set of .dbf files contains properties (data), and a separate collection of .prg files contains the methods (programs) that apply to the set of .dbf files.

In contrast to the conventional programming technique, Access takes an object-enabled approach to database management. Access combines the data (tables) and methods (queries, forms, reports, and Access VBA code) in a single, often massive, .mdb container—known as the `Database` object. `Form` and `Report` objects act as containers for `Control` objects. `QueryDef` objects consist of Structured Query Language (SQL) methods applied to `Table` or other `QueryDef` objects. Although lacking some characteristics of a truly object-oriented programming language (OOPL), Access VBA comes close enough to the mark that you can consider Access VBA an object-enabled application programming language.

The Advantages of Objects

Combining properties and methods into an object and then adding a standard presentation provides the following advantages to users of applications and the programmers who create them:

- Objects combine data (properties) and the program code that deals with the data (methods) into one object you can treat as a "black box." You need not understand the internal elements of the box to use a box in an application. This characteristic of objects aids in the programming of large-scale applications in which many programmers participate. You can modify an object without affecting how the object is used by other programmers.
- Objects can be reused when needed. You can create and use a library of objects in many programs or applications. If you create a library of vector-based images created by CorelDRAW! in an Access table, for example, you can edit these images from within Access and then save a copy of the edited drawing in a separate file.

Oracle Ex. 1009, p. 687
Oracle v. DeBellis
IPR2024-00221

- Objects can be used with any Windows application that supports OLE 2+. You can use the same graphic or sound object with Access, Excel, or Word. If you copy a drawing embedded in an Excel spreadsheet to the Clipboard, you can paste the drawing as an unbound object in an Access form or report, or into an OLE Object field.
- Objects are easy to create with OLE. If you copy all or a portion of the data you create in an OLE-compliant application, you create a temporary OLE object in the Clipboard.

Object Classes, Types, Hierarchies, and Inheritance

Programming objects, like real-world objects, are organized in hierarchies of classes and subclasses, as well as in types without subclasses. The class hierarchy of musical instrument synthesis objects is shown in Figure 19.3 (the subclasses are shown for woodwinds only). At the top of the hierarchical structure you see the master class, Simulated Musical Instruments, that defines the presentation of all the subclasses as digital audio sound.

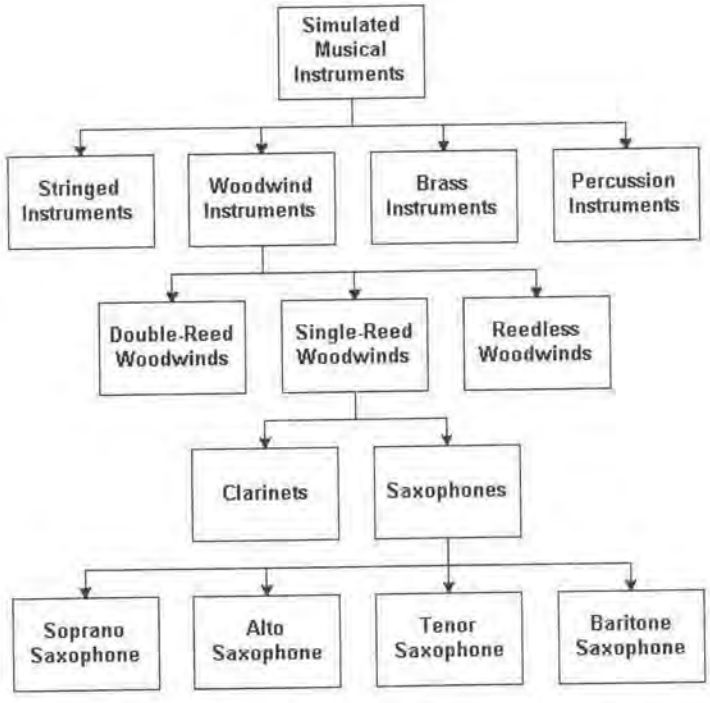


FIG. 19.3 A hierarchy diagram of objects representing the sound of simulated musical instruments.

Musical instruments are classified by the method used to create their sounds. Saxophones are members of the general woodwind class, which you can further subclass as single-reed woodwinds (refer to Figure 19.3). Single-reed woodwinds include clarinets and saxophones, which share many playing methods. Saxophones can be subclassed

Oracle Ex. 1009, p. 688
Oracle v. DeBellis
IPR2024-00221

further by size and the musical key in which these instruments play. At the bottom of the hierarchy are types that have no further subclasses.

A B-flat (tenor) saxophone has a unique set of property values (data), but the kinds of properties are common to all saxophones. Similarly, the tenor saxophone's playing methods are common to all saxophones. The tenor saxophone type inherits both the methods and the list of properties from the saxophone subclass, which in turn inherits properties from the single-reed woodwind class. Parent classes are a level above child classes in the genealogy of classes, and siblings are members of the same class or type. Access uses the term *master* to indicate a parent class object and *child* to indicate a descendent of the parent class, as in the Link Master Fields and Link Child Fields properties of subform and subreport controls.

▶▶ See "Understanding Objects and Object Collections," p. 1023



Figure 19.4 illustrates the hierarchy of objects within Access that are described up to this point in the book. `Table`, `Form`, and `Report` objects have distinctive properties and methods. `Form` and `Report` objects share the same group of `Control` objects, but some `Control` objects behave differently when contained in `Form` or `Report` objects.

Queries (called `QueryDef` objects) aren't true objects; `QueryDef` objects comprise a set of methods applied to `Table` objects. `Control` objects can be classified further into bound and unbound `Control` objects. Only object frames can incorporate OLE objects, and only a bound object frame can display or provide access to editing capability for OLE data in tables or queries without resorting to Access VBA code.

The `Control` objects in Access inherit some methods from the object that contains the control objects. Subform and subreport controls inherit all related methods and many properties from `Form` and `Report` objects, respectively. OLE child objects inherit a large number of their properties and methods from the parent OLE class that defines how OLE objects behave (or should behave). Each OLE child, however, has a complement of properties and methods that apply only to that child. OLE object subclasses at the bottom of Figure 19.4 aren't types because each subclass may have different OLE object types. The Bitmapped Graphic Class, for example, may include Paintbrush Picture, CorelPhotoPAINT Picture, and PicturePublisher Picture OLE object types.

Containing Objects in Other Objects

OLE is designed to allow a single *container document* to be built with contributions, known as *source documents*, from foreign applications. (Container documents were *destination documents* in OLE 1.0 terminology.) The container document may be created by most applications that are OLE compliant. Access forms and reports, proposals created in Word for Windows, Excel worksheets, and PowerPoint presentation visuals are common container documents.

When more than one application contributes to the content of a document, the document is referred to as a *compound document*. The application that creates the compound

Oracle Ex. 1009, p. 689
Oracle v. DeBellis
IPR2024-00221

document is the OLE container application (formerly the OLE client), and the foreign applications that contribute source documents to the compound document are OLE servers. Access 97 is an OLE 2.1 client and creates the compound container documents; Access forms, reports, and tables that incorporate OLE objects are compound documents.

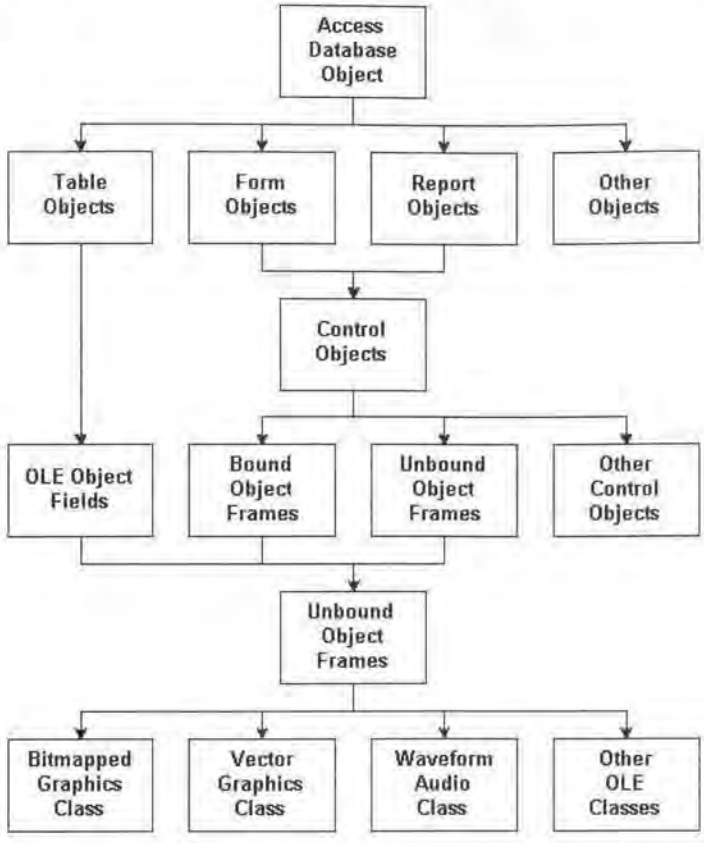


FIG. 19.4 A simplified hierarchy of database objects in Access.

Each source document created with OLE servers and contributed to a container document is an individual, identifiable object that possesses its own properties, possesses its own methods, and has its own presentation. You can choose to *embed* the source document within an Access container document that acts as a container for one or more source document objects. Embedding includes the object's data in the source document. You also can *link* the source document; in this case, the data resides in a separate file. The difference between embedding and linking objects is the subject of the next section, "Understanding Differences Between OLE and DDE."

Oracle Ex. 1009, p. 690
Oracle v. DeBellis
IPR2024-00221

Note

With OLE 1.0, you could activate an embedded or linked object within the container document, but you could not activate any objects contained within the first embedded or linked object. (To *activate* an OLE object means to open the object for editing in the source application.) OLE 2+ eliminates this restriction. Thus, if you embed a Word 97 document containing an embedded Excel 97 worksheet in the OLE object field of an Access table, you can activate the Word document in Word 97 and then activate the worksheet object in Excel 97. There is no limit to how deeply you can nest embedded or linked OLE 2+ objects.

Whether you embed or link the source document, the code to perform the server's methods isn't incorporated in the compound document. Consider the size of an Access table that contains several copies of Excel.exe. Instead, the server's methods are incorporated by a reference to the application's name, known as the *OLE type*, in the source document. You need a local copy of the server application that created the source document to edit an embedded source document or display most linked source documents. Information about OLE server applications is incorporated in your Windows 95 or Windows NT Registry, which takes the place of Windows 3.1's registration database file—REG.DAT. The importance of the Registry to OLE 2+ is explained in the section "The Windows Registry" at the end of this chapter.

Understanding Differences Between OLE and DDE

If you previously used dynamic data exchange (DDE) between Windows applications, OLE gives you the opportunity to add a new dimension to DDE operations. The differences between OLE and DDE techniques for transferring data are illustrated by Figure 19.5. DDE transfers data (properties) from the server to Access text boxes, labels, or other controls that can accommodate text and numbers; OLE transfers data and methods from the server only to unbound or bound object frames.

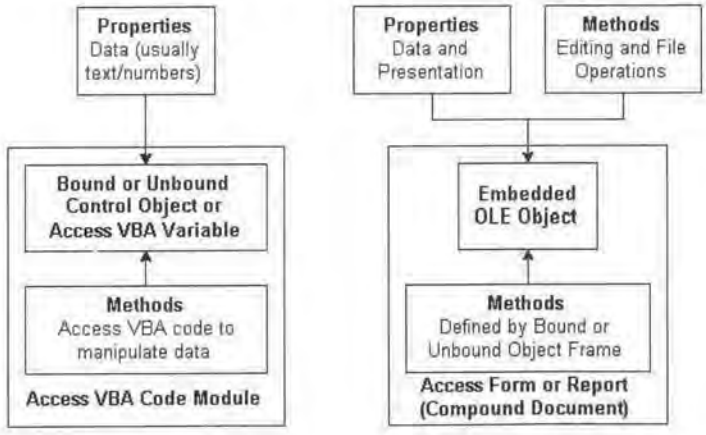


FIG. 19.5 A diagram showing the differences between DDE and OLE data transfer in Access.

Oracle Ex. 1009, p. 691
Oracle v. DeBellis
IPR2024-00221

Using DDE always requires a substantial amount of programming. Writing code to make full use of DDE with Access, or with any other Windows application, is and always was a difficult process. DDE data exchanges are also notoriously unreliable. Using the basic features of OLE, however, requires no programming. OLE data exchanges are also much more reliable than DDE. Simple OLE operations in Access use the Paste Special or Insert Object choices of the Edit menu. You can copy OLE objects in Access to the Clipboard from the Edit menu so that other OLE client applications can use the objects. There are relatively few Access VBA reserved words that apply exclusively to OLE objects; Microsoft deliberately designed OLE for use by nonprogrammers.

OLE 2.+ and ActiveX introduce a new programming methodology for OLE objects that is called *Automation*. The advantage of Automation is that you can alter the values of an object's properties and apply methods to the object by using conventional Access VBA syntax. Automation is the foundation upon which ActiveX Controls and other OLE server objects are built. Automation with OLE servers is discussed briefly in the section "Taking Advantage of Automation with OLE Objects" later in this chapter. Chapter 31, "Exchanging Data with Automation and ActiveX Controls," shows you how to write Access VBA code to manipulate Automation objects.

Embedding versus Linking Source Documents

When you embed a source document, Windows creates a copy of the source document data and embeds the copy permanently in the container document. The source document retains no connection to the embedded data. Subsequent editing or deletion of the file or data from which the source document was created has no effect on the embedded copy in the table or displayed on the forms and reports.

Embedding is the only option if you do not or cannot save the source document as a file. The Microsoft Graph 8.0 and Visio Express OLE 2.0 mini-servers allow you only to embed a file. These applets have no Save or Save As choices in their File menus. When you choose Exit from an applet File menu, a dialog appears that asks whether you want to update the destination document with the edited source document.

The Windows 95 Paint application is the OLE 2.1 source application for all graphics displayed in bound or unbound objects of the Northwind Traders sample database. Paint allows you to import and save images in monochrome, 16-bit, or 24-bit .bmp files. Thus, you can use Paint to embed or link all bitmapped graphics you possess or create in either of these file formats. (Paint can open but not save PC Paintbrush .pcx files.)

Open the Categories form and then double-click the picture. Paint appears so that you can edit the image, as shown in Figure 19.6. All bitmaps in the Northwind Traders sample database are Bitmap Image objects that are embedded in OLE Object fields of tables and unbound object frames on forms and reports. Notice in Figure 19.6 that the menubar changes when you activate the embedded image for editing. Activating Paint grafts the Image and Options choices of Paint's menu to Access's menubar.

If you link an object, you create a reference to the data of the source document: the name of a file. When you or others change the data in a file linked to the object, the data in the source document permanently changes. The next time you display the data, if you

Oracle Ex. 1009, p. 692
Oracle v. DeBellis
IPR2024-00221

set the value of the Update Options property of the bound object frame to Automatic, the presentation of the object incorporates all changes made to the data in the file. The ability to automatically update everyone's linked objects at once is useful in a networked Access application that displays data, which periodically is updated by others who share these files on the network.



FIG. 19.6 The Windows Paint application editing an embedded bitmap that is displayed in a bound object frame.

Caution

If you want to try the following process for linking a range of an Excel worksheet to a record of Northwind.mdb's Categories table, make a copy of Northwind.mdb and open the copy in Access. You can't undo an embedding or linking update to an Access table.

Note

The following steps use a dummy record in the Categories table of Northwind.mdb in which to embed data. The Categories table is displayed by using the Categories form included with Northwind.mdb. To duplicate on your own computer the figures shown in this section, open the copy of the Northwind.mdb sample database, then open the Categories form. Click the Tentative Append Record button on the toolbar to create a new record in which you can experiment with embedded or linked data.

To link rather than embed an object created by an OLE server, perform the following steps:

1. Open the OLE server—Excel 97 in this example.
2. Load the workbook or other file that contains the data you want to link.

Oracle Ex. 1009, p. 693
Oracle v. DeBellis
IPR2024-00221

3. Select the range of data (or part of an image) to copy to the Clipboard. Most OLE 2+ server applications let you select all or part of the open document.
4. Press Ctrl+C to copy the data to the Clipboard.
5. Activate the container—in this case, an Access 97 form—and select a bound or unbound object frame. For this example, use the Picture field of the Categories form.
6. Choose Edit, Paste Special in the container application. If the application supports OLE 2+, the Paste Special dialog shown in Figure 19.7 appears.



FIG. 19.7 The standard OLE 2+ Paste Special dialog.

7. Click the Paste Link option button, and then click OK to paste the image into the container document. In this example, the container is the Picture field of a dummy test record that is added to the Categories table of Northwind.mdb, and displayed by the Categories form.

> **Note**
>
> Before creating an OLE link in the destination application, you must open the source application and load the file because opening a file in most OLE servers breaks the existing OLE connection.

When you double-click to edit a linked object, the application opens its window and the title bar of the server's window displays the name of the file that created the image, as shown in Figure 19.8. This method is one quick way to determine whether an OLE object is linked, rather than embedded, in most OLE client applications (including Access).

Another advantage of linking some objects whose data resides in files is that you don't waste disk space with duplicate copies of the files in the database. Access creates a copy of the source document's presentation in the table, so you don't save disk space when you link graphic images, spreadsheets, or other source documents whose presentation consists of the data. The presentation of animation, waveform audio, and digital video files is either an icon or the first image of a sequence, so the presentation is small in size. Here, the disk space saved by linking rather than embedding is substantial.

Oracle Ex. 1009, p. 694
Oracle v. DeBellis
IPR2024-00221

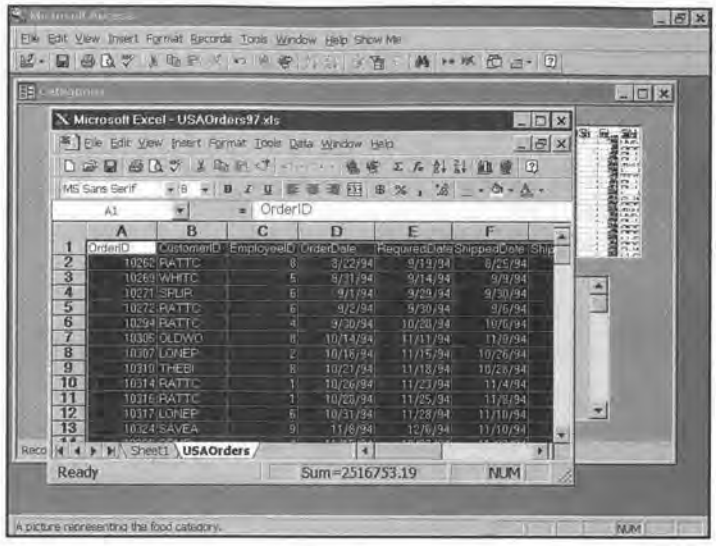


FIG. 19.8 Excel 97 editing a worksheet file with a range of cells linked to a bound object frame.

Note

Linking a bitmapped image doesn't save disk space; the presentation stored in .bmp format is the same size in bytes as the original image. (The size of the presentation is likely to be much larger if your image file was stored in a compressed format, such as .pcx, .gif, or .jpg.) If your OLE image server application lets you substitute a thumbnail image, the amount of file space saved is often substantial. A *thumbnail image* is a smaller version of the original image that often has less depth of color (8-bit instead of 24-bit color data, for instance).

The disadvantage of linking is that the linked files must be available to all users of the application. The files must reside in the same directory for their lifetime unless you edit the linkage to reflect a new location. A fully specified path, including the drive designator or UNC server address, is included with the file name in the linking information. If your application can't open the linked file, the presentation of the source document appears in the form or report, but when you double-click the bound object frame to edit the linked object, you see an error message that the linked file is not found.

Note

OLE 2+ has a limited capability to update the links to files that have moved. However, it is not a wise practice to depend on the present version, OLE 2.1, to maintain links to relocated files. If the object is contained in an unbound object frame, you can change the value of the `SourceDoc` property to point to the file in its new location. If the object is contained in a bound object frame, you need to open the file, copy its data to the Clipboard, and then re-create the link by using Paste Special to create a new link to the file.

Oracle Ex. 1009, p. 695
Oracle v. DeBellis
IPR2024-00221

Activating OLE 2+ Objects in Place

One of the advantages of embedding, rather than linking, objects is that you can activate OLE 2+ objects in place. In-place activation (also called *in-situ editing* or *in-place editing*) causes the source application for the embedded object to "take over" the container application.

You activate an OLE object for editing by double-clicking the surface of the object frame that displays a simplified presentation of the worksheet. Figure 19.6 illustrates in-place activation of a Paintbrush Picture object. Figure 19.9 shows an Excel worksheet embedded in an unbound object frame of a form. The worksheet has been activated in place for editing. When an object is activated in place for editing, the object is surrounded by a frame that consists of alternating gray and black, diagonal hash marks. Eight sizing handles (black squares at each corner and at the midpoints of the frame) let you adjust the size of the editing frame.

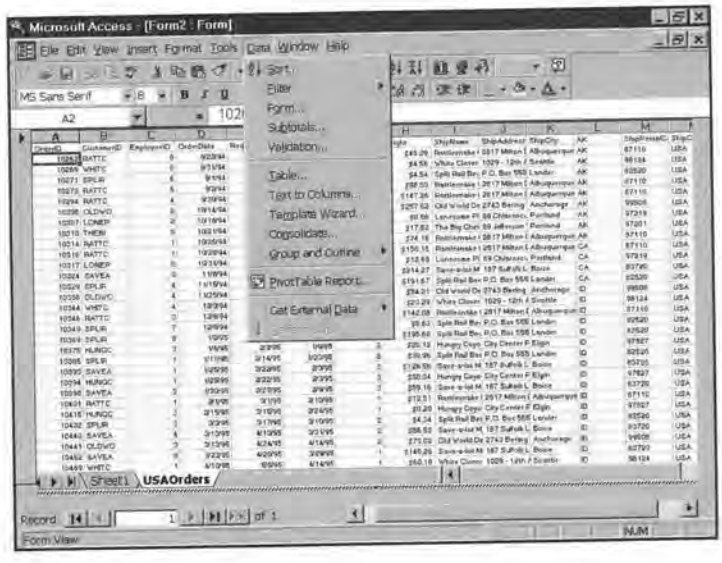


FIG. 19.9 An Excel 97 document activated for editing in an Access 97 form.

When you activate an embedded OLE 2+ object, the menubars of the application with which you created the object replace Access's menus with the same name. This process is called *grafting a menu*. If the source application has additional menus, they are added to Access's menubar.

In Figure 19.9, the Edit, View, Insert, Format, Tools, and Help menus are replaced by Excel's menus of the same name. Excel adds its own Insert, Tools, and Data menus to the menubar. Excel's anchored toolbars can become floating toolbars that you can reposition or hide. Depending on the specific object you have activated for in-place editing, various screen adornments will appear. *Adornments* are the rulers, column and row headers, sheet tabs, and scroll bars that are part of the object's presentation while being edited by its

Oracle Ex. 1009, p. 696
Oracle v. DeBellis
IPR2024-00221

OLE server. You have full control over the embedded object, and you can perform almost any operation on the object that is possible if you had opened a file containing the object directly in its server application.

Note

In-place activation is available only for embedded, not linked, OLE 2+ objects. When you double-click the presentation of a linked OLE 2+ object, the source application's window appears. In this respect, linked OLE 2+ objects behave identically to linked or embedded OLE 1.0 objects.

Taking Advantage of Automation with OLE Objects

OLE 2+ server and mini-server applications that support Automation *expose* programmable objects. As an example, Excel 97 is an Automation server application that creates programmable Excel objects. Thus, when you activate an Excel worksheet object embedded in, or linked to, an Access 97 object frame, you can gain access to the worksheet through the `Object` property of the object frame control. The `Object` property is available only with Access VBA code; the `Object` property does not appear in the Properties window for object frames. The `Object` property provides access to all of the objects, and the properties and methods of the objects, exposed by the application that created the embedded or linked object.

▶▶ See "Manipulating an Excel 97 Workbook Object," p. 1102



Using Automation requires that you learn to program in Access VBA and that you understand the hierarchy of the collections of objects that you intend to program. Collections of objects are groups of objects of the same class that are contained within another object. As an example, the collection of Excel `Worksheet` objects are contained in `Workbook` objects. Collections of Automation objects follow the same general structure as Access 97's data access objects (DAOs) described in Chapter 29, "Understanding the Data Access Object Class." Excel 97, for example, has more than 100 different types of objects and several hundred properties and methods that apply to its objects.

Access 97's `DBEngine` object is at the top of the Access DAO pyramid; the `Application` object of Excel 97 (and most other Automation servers) is the equivalent of the `DBEngine` object. All other Excel objects are subclasses of the `Application` object. Access 97 exposes its `Application` object to OLE client applications, qualifying the object for Automation Server status.

Automation lets you construct applications by using the building block approach. If you need a worksheet in your Access application, insert an Excel `Worksheet` object in an unbound object frame. You then set the values for properties of, and apply methods to, the `Worksheet` object with Access VBA code. If you want word processing features, insert a

Oracle Ex. 1009, p. 697
Oracle v. DeBellis
IPR2024-00221

Word 97 `Document` object into an unbound object frame. You add command buttons or other controls to the form to execute the procedures or functions that contain the required Automation commands.

ActiveX Controls are a special class of OLE 2+ mini-servers that use Automation to provide access to their properties, methods, and events. (Conventional OLE and Automation objects don't expose events when activated.) When you embed an ActiveX Control in an unbound object frame, the Control object's events are not added to the Property window's list of events for unbound object frames. However, you can write Access VBA event-handling code for the events exposed by the ActiveX Control as subprocedures in code contained in forms.

The preceding brief discussion of Automation may appear out of context at this point in the book because writing Access VBA code is the subject of Part VII, "Programming, Distributing, and Converting Access Applications." However, future Windows operating systems and applications will extensively use Automation. A general description of OLE 2.1 without including Automation would be incomplete, at best.

As mentioned at the beginning of this section, taking advantage of Automation and ActiveX Controls requires that you write Access VBA code. Mastering Automation programming requires a very retentive memory and much experimentation. If you need interprocess communication in your Access 97 applications, use OLE 2.1 and Automation—not DDE—whenever you can. Your investment astride the OLE 2.1 learning curve will return substantial dividends as more and better Automation applications and, especially, 32-bit ActiveX Controls become available.

The Windows Registry

In Windows 95 and Windows NT, the Registry takes the place of Windows 3.1+'s REG.DAT, plus WIN.INI, SYSTEM.INI, and application-specific INI files. (If you install Windows 95 over Windows 3.1+, Setup migrates all REG.DAT and many WIN.INI and SYSTEM.INI entries to the Registry.)

The Registry consists of two hidden system files: System.dat and User.dat. User.dat can be on your local fixed disk drive or stored on a network server. System.dat contains information on your PC, including the hardware and software installed. User.dat contains your user profile information. If your User.dat file is stored on a server, you can move to another networked computer, log on with your user name and password, and have your desktop configuration appear at the computer you're using.

System policy information, contained in .pol files, overrides entries in System.dat and User.dat. System policy files, such as Config.pol, are primarily of interest to PC administrators for maintaining control over the extent to which users can modify their Windows 95 or Windows NT Workstation 4.0 installation.

The registration process for early OLE 1.0 servers was relatively simple. Only a few entries in REG.DAT were required, primarily to create an association between individual file

Oracle Ex. 1009, p. 698
Oracle v. DeBellis
IPR2024-00221

extensions and the OLE server for the type of file. The registration process for OLE 2+ servers is not so simple; much more information is required to register OLE 2+ servers, and even more entries are added for applications that support Automation through OLE or ActiveX technology.

ActiveX and OLE Controls also register themselves in the Registry. Although the setup program registers each OLE server and control automatically, you must understand the use of the Registry and the Registry Editor application in case you encounter a problem when you attempt to insert an OLE 2+ or ActiveX object in an Access 97 object frame. The following two sections are devoted to the Registry and concentrate on the OLE 2.1 aspects of server registration.

The Registry Editor

The Registry Editor, RegEdit.exe, is included with Windows 95 and Windows NT so that users can display and edit the Registry, formerly the registration database of Windows 3.1+. The Registry Editor has a help file that explains how to use RegEdit, and the *Windows 95 Resource Kit* includes an explanation of the Registry's contents and also describes how to use RegEdit. None of this information provides even a hint about the new Registry data structures required by OLE 2.1. If you plan to use the OLE Object field data type, to use MSGraph8 to create graphs, or add sophisticated graphic images to your forms and reports, you need to know more about the Registry and RegEdit than appears in the Windows and Access 97 documentation.

Although the RegEdit program files are installed, the Windows Setup program doesn't install a program icon on the Start menu for RegEdit. To run RegEdit, choose Start, Run and type **regedit.exe** in the Command Line text box. Click OK to launch RegEdit. RegEdit's opening window appears, as shown in Figure 19.10.

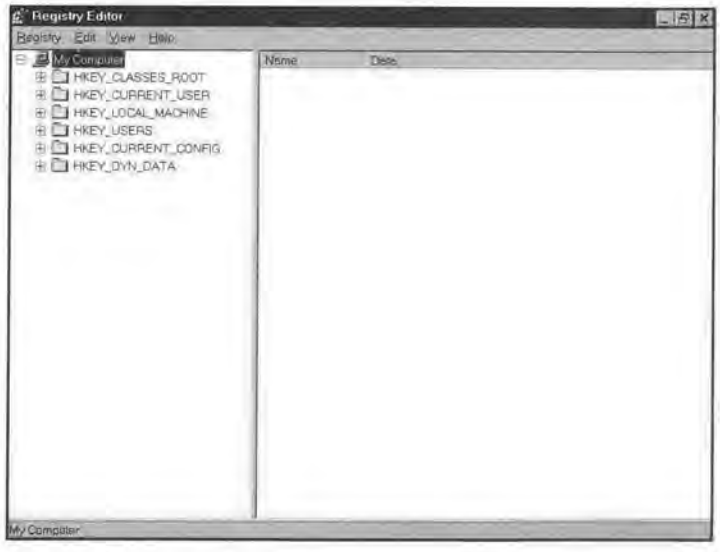


FIG. 19.10 RegEdit's opening window for viewing or editing the Registry files.

Oracle Ex. 1009, p. 699
Oracle v. DeBellis
IPR2024-00221

Information contained in the Registry is stored in the six subtrees, each beginning with HKEY, as shown in Figure 19.10. The type of information stored in each subtree is as follows:

- *HKEY_CLASSES_ROOT* contains association mappings between applications and file types identified by the file extension. HKEY_CLASSES_ROOT also contains information to support drag-and-drop operations, and contains data on the user interface. This subtree includes information similar to that stored in the REG.DAT file of Windows 3.1+ and Windows NT 3.1+.
- *HKEY_CURRENT_USER* stores a pointer to the *hive* (sub-subtree) of the HKEY_USERS subtree for the user currently logged on to Windows 95.
- *HKEY_LOCAL_MACHINE* contains information on the hardware and software installed on your PC. This subtree is employed by all users who log on to your PC.
- *HKEY_USERS* stores information on all users of your PC if Windows security is implemented. If only you use your PC, HKEY_USERS stores the data for HKEY_CURRENT_USER.
- *HKEY_CURRENT_CONFIGURATION* stores a pointer to a hive in HKEY_LOCAL_MACHINE that contains details about the current hardware setup of your PC.
- *HKEY_DYN_DATA* contains dynamic status information for Plug and Play devices, such as PC Cards (PCMCIA devices) that you "hot swap" while your PC is powered.

The primary subtrees used by OLE 2+ are HKEY_CLASSES_ROOT and the SOFTWARE hive of HKEY_LOCAL_MACHINE. To display hives below subtrees, double-click the folder symbol for the subtree, then double-click folders with a + symbol to expand the hierarchical list. HKEY_CLASSES_ROOT identifies a file extension with a particular class of object, such as .doc files with the `Word.Document.8` object class—called the *programmatic ID* or *ProgID* (see Figure 19.11).

When you double-click a .doc file in My Computer or Explorer, the association launches the application's executable file specified by the `...\shell\open` value that appears in the Data column of RegEdit's right pane. If you don't have Word 97 on your computer, Windows 95 launches WordPad to create a `Wordpad.Document.1` object from .doc files. Similar entries associate .mdb files with the `Access.Application.8` object class.

Note

You can see a list of registered file types by launching Explorer, choosing View, Options, and clicking the File Types tab. You can change the association between a file extension and an application by clicking the Edit button of the File Types tab. Editing associations in the File Types page is similar to the process used with the Modify File Type editing feature of Windows 3.1+'s REGEDIT.EXE. The most common change to file associations is changing the editing application for bitmapped graphics files.

Oracle Ex. 1009, p. 700
Oracle v. DeBellis
IPR2024-00221

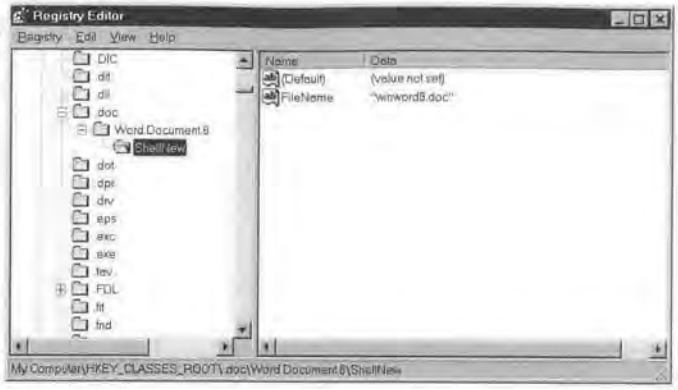


FIG. 19.11 Entries in RegEdit's HKEY_CLASSES_ROOT subtree that associate .doc files with applications.

Following the list of file extension associations in HKEY_CLASSES_ROOT is an alphabetized list of ProgIDs for registered objects. Figure 19.12 shows the HKEY_CLASSES_ROOT\Word.Document.8\shell\New\command entry that launches Word 97 with the /n parameter to specify a new (empty) document. If you move a registered application from one folder to another, you must change every Registry entry for the application that contains the pointer to its executable file.

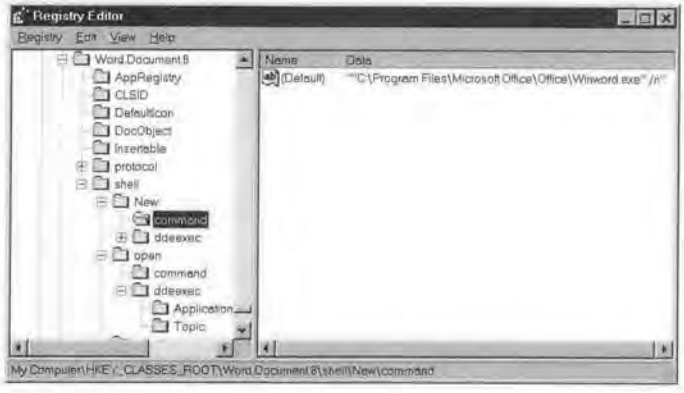


FIG. 19.12 The Registry entry that enables Windows to create a new Word 97 document without first starting Word.

Windows' RegEdit includes a flexible Find feature that allows searching for entries' keys, values, and/or data (see Figure 19.13). Figure 19.14 shows the Edit String dialog that appears when you double-click the "ab" icon that indicates the entry is a string of characters. Unfortunately, RegEdit doesn't include a find and replace feature, so you must make multiple Registry entries manually.

Oracle Ex. 1009, p. 701
Oracle v. DeBellis
IPR2024-00221



FIG. 19.13 Searching for the Registry entries that contain the `Word.Document.8` programmatic ID.



FIG. 19.14 The Edit String dialog for changing a Registry value.

▶▶ See "Registry Entries for Automation Servers," p. 1108

Applications that support OLE 2+ add a variety of other entries to the Registry; an explanation of the purpose of each of these entries is beyond the scope of this chapter. What is important, however, is your ability to recover from problems that result from improper entries for OLE 2+ servers in the Registry.

Troubleshooting

When I try to insert an object into an unbound object frame, I get a message that Windows can't find the OLE server or one of its components.

The Registry contains an entry that points to the wrong location of the OLE server, the server executable has been erased (or is corrupted), or one of its DLLs or other supporting components is missing. Although you can manually change the Registry entries for relocated OLE servers, reinstalling the server from its distribution diskettes or CD-ROM is a far more foolproof process. You're likely to have a better chance of satisfactorily correcting the Registry entries for the server if you launch the Add/Remove Programs feature of Control Panel, remove the server, and then reinstall the server. Many older OLE server applications, however, don't display entries in the Install/Uninstall page of the Add/Remove Programs dialog.

How Access and Other OLE Applications Use the Registry

The primary Registry entries for each OLE server appear in the Object Type list box of the Insert Object dialog that appears when you choose Insert, Object in Form Design view. Figure 19.15 shows the Insert Object dialog that is standard for all OLE 2-compliant applications with the Microsoft Graph 97 entry (`MSGraph.Chart.8` ProgID) selected.

Oracle Ex. 1009, p. 702
Oracle v. DeBellis
IPR2024-00221

If you have installed prior versions of current OLE 2.0 servers, such as Excel 5.0 and Word 6.0, entries may appear for these servers even if you have deleted the applications from your fixed disk. If you use the Uninstall option of the Maintenance Setup applications to remove applications, the Registry entries for these applications are deleted. Unfortunately, many older OLE 2.0 servers don't provide an uninstaller. All applications that display the "Designed for Windows 95" logo must include an uninstall feature that clears registry entries during the application removal process.

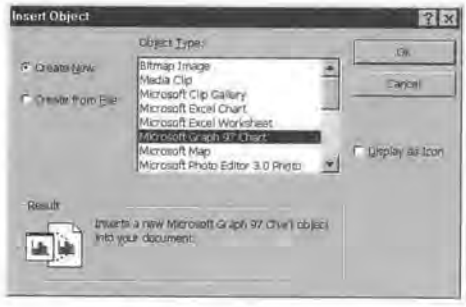


FIG. 19.15 The OLE 2+ Insert Object dialog that allows you to select an available object type based on the Registry entries.

Each Object Type in the Object Type list box is a specific type of the OLE object. The entries in the list box are derived from the Registry's value of each registered OLE server. The Microsoft Graph 97 type, for example, is entered from the `MSGraph.Chart.8 = Microsoft Graph 97 Chart` in `\HKEY_CLASSES_ROOT`. If you double-click the Microsoft Graph 97 Chart entry (or select the entry and click OK), Access creates an unbound object frame, and Graph8.exe opens with the example graph shown in Figure 19.16.

You can enter data in the datasheet at the upper left of Microsoft Graph's window to alter the height of the bars of the chart, or you can import an Excel .xlc chart file, but that's about all you can do without having a working knowledge of how graphs obtain data from Access 97. Creating graphs from Access data is one of the subjects of Chapter 20, "Adding Graphics to Forms and Reports."

Replacing the Registry Files with Backups

When you start Windows, a backup copy of System.dat and User.dat (System.da0 and User.da0, respectively) is created automatically. If either of these files becomes corrupted, you may not be able to start Windows. If the `HKEY_CLASSES_ROOT` or `HKEY_LOCAL_MACHINE\SOFTWARE` entries for an OLE server in System.dat are incorrect or missing, the server won't appear in the Object Type list of the Insert Object dialog, or you receive an error message when attempting to insert an object. Corruption of System.dat isn't a common occurrence, but it occasionally happens during the installation of a new application. You can replace the corrupted System.dat file with the backup System.da0 version by following these steps:

Oracle Ex. 1009, p. 703
Oracle v. DeBellis
IPR2024-00221

1. If your corrupted System.dat file prevents Windows from starting, launch Windows and immediately press the F8 key for a fail-safe boot. Then choose Safe Mode Command Prompt Only. Skip the next two steps.
2. If you can run Windows, open Explorer and choose View, Options. In the View page of the Options dialog, click the Show All Files option, and then click OK to close the dialog.
3. Locate System.dat and System.da0 with Explorer. The location of these two files depends on how you installed Windows, but they are located most commonly in your \Windows folder.
4. If you are working from the MS-DOS command prompt, skip to step 7. In Explorer, right-click System.dat and choose Properties from the resulting popup menu. Clear the Read-Only and Hidden check boxes on the file's properties page. Repeat this step for the System.da0 file.
5. Click the System.da0 file to select it, choose Edit, Copy, then choose Edit, Paste; Windows creates a copy of System.da0 that is named Copy of System.da0.
6. Delete System.dat, and then rename Copy of System.da0 to **System.dat**. Skip to step 9 to finish this procedure.
7. If you're working from an MS-DOS prompt, change to the directory containing System.dat. System.dat and System.da0 are hidden, read-only system files, so entries for these files do not appear when you execute a `dir` command.
8. At the MS-DOS command prompt, type the following three instructions:

   attrib -h -r -s system.dat

   attrib -h -r -s system.da0

   copy system.da0 system.dat

9. Reboot your computer. (It is not necessary to reset the attributes of System.dat.)

To replace User.dat with User.da0, follow the preceding steps and substitute **user** for **system** in steps 5, 6, and 8.

Oracle Ex. 1009, p. 704
Oracle v. DeBellis
IPR2024-00221

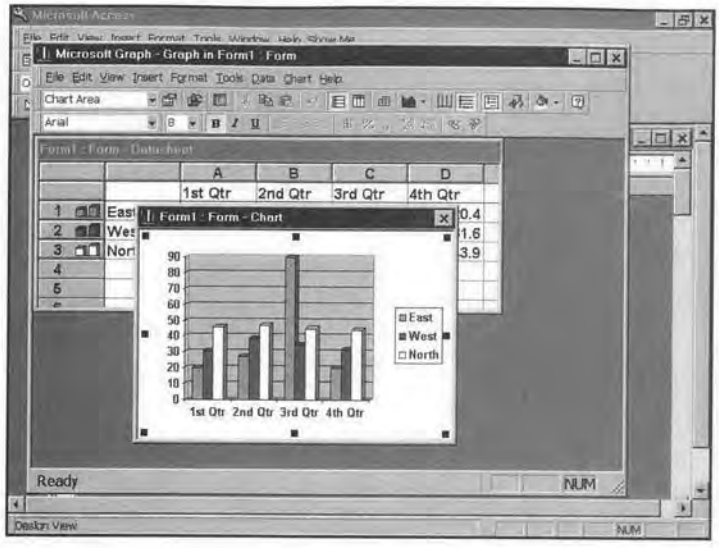


FIG. 19.16 An example graph created by inserting a new `MSGraph.Chart.8` object in Form Design view.

Oracle Ex. 1009, p. 705
Oracle v. DeBellis
IPR2024-00221

Chapter 20

Adding Graphics to Forms and Reports

One of the principal incentives for using a Windows desktop database manager is the ability to display graphic images contained in (or linked to) database tables. Early Windows RDBMSs could display images stored in individual bitmapped graphic files with common formats such as .bmp, .pcx, and .tif, but they could not store the bitmap data within the database file.

A few publishers enhanced some of these early desktop database products with Binary Large OBject (BLOB) field data types. A BLOB field is of variable length and can hold any type of data, regardless of its format. Other Windows desktop RDBMSs use auxiliary files (similar to dBASE's .dbt memo files) to store graphic images and other types of non-text data. When an auxiliary file is used, a field in a database table provides a reference (called a *pointer*) to the location of the data in the auxiliary file.

Many Windows graphic images are stored as combinations of lines, shapes, and patterns, rather than as copies of the pixel pattern of an image on your video display unit. Images of this type are called *vector-based graphics*. Windows illustration applications (such as Corel Systems' CorelDRAW! and Visio Corporation's Visio) create vector-based graphics. Microsoft Graph 97 (MSGraph8) uses vector-based graphics to create graphs from data in Access tables. Although each of these products has a proprietary file format, the illustration applications communicate with other applications through the Clipboard in standard Windows Metafile Format (.wmf).

This chapter describes how to use both bitmapped and vector-based graphics in conjunction with Access forms and reports. It also describes how you use MSGraph8 to create graphs and charts from Access 97 data.

Adding a Bound Object Control to a Form or Report

Graphic images and other OLE objects stored in OLE Object fields of Access tables use a bound object frame control to display their presentation. The

Oracle Ex. 1009, p. 706
Oracle v. DeBellis
IPR2024-00221

bound object frame control is an OLE container within which a bitmapped or vector-based image can be displayed. Other OLE objects that rely on data stored in OLE Object fields, such as Sound Recorder and Media Player objects, plus data-bound OLE Controls use the bound object frame.

In the case of still graphic images, the presentation within the bound object frame is a copy of the object's data property. Animated images and video objects usually display the first image in the animation sequence or video clip. Sound objects substitute the icon of the OLE server with which their file type is associated in the Registry. Double-clicking the bound object frame launches the OLE server that was used to add the object to a data cell in an OLE Object field of your table.

In Chapter 13, "Designing Custom Multitable Forms," you were introduced to adding a bound object frame to an Access form. Here, you learn the details of displaying and editing a photograph in the Personnel Actions Entry form you have built in the preceding chapters. You also learn how to scale the photograph within the bound object frame so that you get exactly the look you want.

Including Photos in the Personnel Actions Query

The majority of Access applications use the OLE Object field data type to store only graphic images. The Northwind Traders Employees table includes a photograph for each employee as one such graphic image. You use the OLE Object field data type to add the photograph to your Personnel Actions form. The bound object frame is linked to the Photo field of the qryPersonnelActions query (see Figure 20.1).



◀◀ See "Creating the Query on Which to Base the Main Form," p. 452

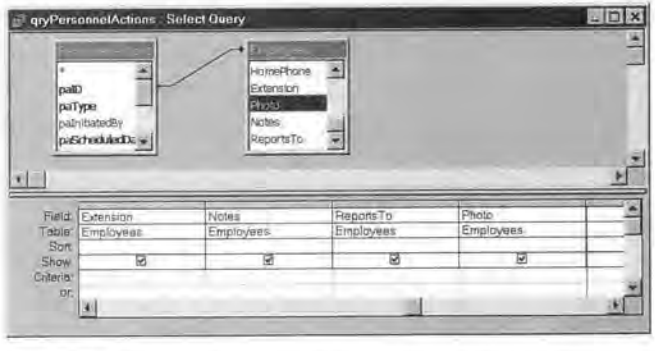


FIG. 20.1 The Photo field of the qryPersonnelActions query in Design view.

To run the qryPersonnelActions query so you can view one of the photographs in the Windows 95 Paint window, follow these steps:

1. From the Database window, open the qryPersonnelActions query.
2. Drag the horizontal scroll bar button to the right to display the Photo field in the resulting table.

Oracle Ex. 1009, p. 707
Oracle v. DeBellis
IPR2024-00221

3. Double-click one of the Bitmap Image data cells to display the image in Windows 95 Paint. The chosen image appears in the Paint window (see Figure 20.2).

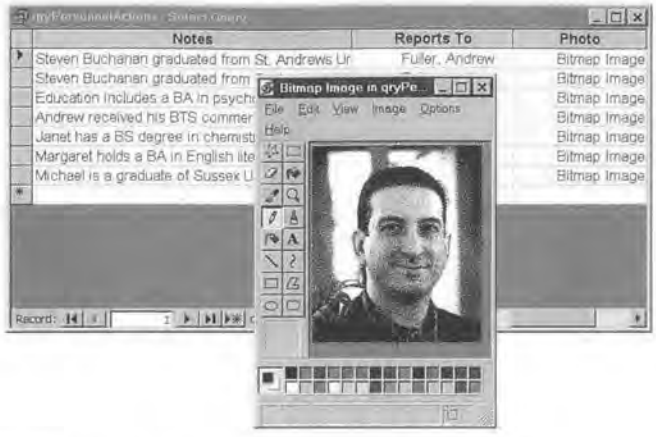


FIG. 20.2 Editing the Photo field of the qryPersonnelActions query.

4. Choose Exit & Return to qryPersonnelActions:Select Query from Paintbrush's File menu to close the editing window.

5. Choose Close from Access's File menu to close the query.

◀◀ See "The Registry Editor," p. 683



Troubleshooting

An Insufficient memory or Application not properly registered message appears after double-clicking a data cell in the OLE Object field.

Either of the two messages can occur under low-memory conditions. First, try closing all other running applications and then double-clicking the data cell again. If you continue to receive registration error messages, exit and restart Windows. If this procedure doesn't solve the problem, open the Registry Editor (RegEdit), choose Edit, Find, and search for the **Bitmap Image** value. Expand the `CLSID` entry for Bitmap Image and select the `LocalServer32` item. Verify that the value for `LocalServer32` is `C:\PROGRA~1\ACCESS~1\MSPAINT.EXE`—or the location of Mspaint.exe on your PC. (The `LocalServer32` entry contains the DOS name of the folder in which Paint is installed: C:\Program Files\Accessories, by default.) If you need to change the entry, double-click the "ab" icon in the right pane to open the Edit String dialog and correct the entry as necessary.

The behavior of Windows Paint is similar to that of other OLE 2+ local server applications used to add or edit the values (contents) of data cells in OLE Object fields.

Displaying the Employee's Picture in the Personnel Actions Form

You can edit OLE objects in Access tables and queries only through the window of the OLE server you used (Paint, in this example) to add the objects to the table. The

Oracle Ex. 1009, p. 708
Oracle v. DeBellis
IPR2024-00221

presentation of OLE objects is, however, stored in the OLE field and displayed automatically in a bound or unbound object frame. You double-click within the object frame to edit the object.

You added a bound object frame to the Personnel Actions Entry form in Chapter 13, "Designing Custom Multitable Forms." For this exercise, you'll create a temporary form for the purpose of experimenting with bound object frames. To add a bound object frame to your experimental form so you can display the Photo field of your qryPersonnelActions query, follow these steps:



1. Click the Forms tab of the Database window, and then click New to display the New Form dialog.
2. Select qryPersonnelActions as the data source for the new form, select Design View in the list, and then click OK to display the new form in Design view.



3. Click the Field List button, and select Photo from the Field List window.
4. Select the Photo field and drag the Field symbol to the approximate position of the upper-left corner of the new form. Access creates a bound object frame rather than a text box when you create a control directly from a field of the OLE Object type in the Field List dialog.
5. Position and size the new bound object frame, as shown in Figure 20.3.



6. Click the Form View button to display your form with the photograph. The form appears as in Figure 20.4. Notice that the photo doesn't fill the entire frame. You may still need (or want) to scale the image, which is the subject of the next section.

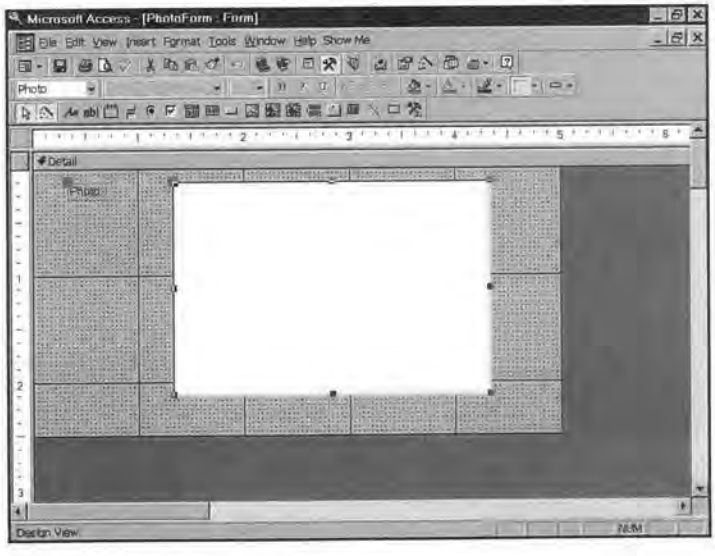


FIG. 20.3 Adding a bound object frame to a form.

Oracle Ex. 1009, p. 709
Oracle v. DeBellis
IPR2024-00221



FIG. 20.4 Viewing the form with the added Photo bound object frame.

Troubleshooting

The employee's photo does not appear in Form view.

If you add the Photo field to your query and follow the procedure for adding a bound object frame while the form is open, your frame is likely to be empty when you display the form in Run mode. In this case, the Photo field is added to the Datasheet view of your form, but the field's cells have `Null` values. Pressing Shift+F9 (Requery) does not replace the `Null` values with Bitmap Images. If this problem occurs, you must close the form and then reopen it from the Database window. Access runs the modified query when you open the form, and the picture appears.



An alternative method for creating a bound object frame is to click the bound object frame tool of the toolbox, click the Photo field, and drag the Field symbol to the form. The extra step involved in this process serves no purpose because Access chooses a bound object frame for you when you choose a field in the Field list that is of the OLE Object data type.

When you use either the bound object frame tool or the toolbox, the value of the Enabled property is set to Yes and the Locked property is set to No. (Access 2.0 set the Enabled property to No and the Locked property to Yes when using the toolbox to create a bound object frame.) The effect of these two properties is as follows:

- When an object frame is disabled (the Enabled property is set to No), you can't double-click the object to launch the OLE server that created the object's content. The setting of the Locked property has no effect in this case.
- When an object is enabled and locked (the Locked property is set to Yes), you can launch the OLE server, but any edits you make to the content of the object are discarded when you close the server application.

Oracle Ex. 1009, p. 710
Oracle v. DeBellis
IPR2024-00221

Note

In Form view with the default values of the Enabled and Locked properties—Yes and No, respectively—double-click the Photo object frame to launch Windows Paint, which assumes the role of the active application through in-place activation. Paint is an OLE 2.1 server, so Paint grafts its menu choices to the Access menubar, a process also called *menu negotiation*. Paint's View, Image, and Options menu choices replace several of Access's menu choices (see Figure 20.5). You can edit the bitmap in Paint and then save the changes to the OLE data in the Employees table when you close Paint. You click the form outside the Photo image to close (deactivate) Paint.



FIG. 20.5 Windows Paint activated in place to edit a Bitmap Image object.

Scaling Graphic Objects

Access provides three methods for scaling graphic objects within the confines of a bound object frame. You select one of these methods by choosing the value of the Size Mode property in the Bound Object Frame Properties window, displayed in Figure 20.6.



FIG. 20.6 Setting the Size Mode property of an object frame.

Oracle Ex. 1009, p. 711
Oracle v. DeBellis
IPR2024-00221

The three options offered for the value of the Size Mode property display the image in the following ways:

- *Clip*, the default, displays the image in its original aspect ratio. The aspect ratio is the ratio of the width to the height of an image, measured in pixels or inches. A *pixel* is the smallest element of a bitmap that your computer can display—a single dot. The aspect ratio of the standard VGA display, for example, is 640×480 pixels, which is 1.33:1. If the entire image does not fit within the frame, the bottom or right of the image is cropped. *Cropping* is a graphic arts term that means cutting off the portions of an image outside of a window of a specified size, as shown in the top picture of Figure 20.7.



FIG. 20.7 Comparing the Clip, Stretch, and Zoom values of the Size Mode property.

- *Stretch* independently enlarges or shrinks the horizontal and vertical dimensions of the image to fill the frame. If the aspect ratio of the frame is not identical to that of the image, the image is distorted, as illustrated by the center image of Figure 20.7.
- *Zoom* enlarges or shrinks the horizontal or vertical dimension of the image so that it fits within the frame, and the original aspect ratio is maintained. If your frame has an aspect ratio different from that of the image, a portion of the frame is empty, as shown in the bottom image of Figure 20.7.

The bound object frames in Figure 20.7 have been shortened vertically and expanded horizontally to accent the effects of the Stretch and Zoom property values.

Note

The figures in this book were created using a screen resolution of 800×600 pixels. If you're working on a computer system using the standard VGA resolution of 640×480 pixels, the effects of the default Clip property on your image are likely to be much more noticeable than in the figures shown here.

Oracle Ex. 1009, p. 712
Oracle v. DeBellis
IPR2024-00221

Access cannot specify a particular area of the image to be clipped, so zooming to maintain the original aspect ratio is the best choice in this case. When you scale or zoom a bitmapped image, the apparent contrast is likely to increase, as shown in the center and bottom images of Figure 20.7. This increase results from deleting a sufficient number of pixels in the image to make it fit the frame, which increases the graininess. The increase in graininess and contrast is less evident in 256-color (8 bits per pixel) bitmaps; the photos of employees are 16-color (4 bits per pixel) bitmaps.

To apply the Zoom property to your bound object frame in Design mode, follow these steps:

1. Select the Photo bound object frame.
2. Click the Properties button on the toolbar to open the bound object frame Properties window. Click the Format tab.
3. Click the Size Mode text box, and open its list box.
4. Select Zoom.

To display the form so that you can view the photograph with its new property, follow these steps:

1. Click the Form View button of the toolbar to display your form, which now appears as shown in Figure 20.8.



FIG. 20.8 The bound object frame with Zoom Size Mode applied.

2. If your frame includes an empty area, as illustrated in Figure 20.8, return to design mode, adjust the size of the frame, and rerun the form to verify that the frame has the correct dimensions.
3. Choose File, Close, and save your changes to the form; you might save this experimental form as **frmPhotoForm**.

The technique described in this section allows you to add a bound object frame containing a vector image created with a drawing application, to add a sound clip from a .wav or .mid file, or to add any other OLE object type that you can select from the Insert Object dialog.

To add a bound object frame to the Detail section of a report, you also use the same method outlined here. The quality of the printed image depends on the type of image

Oracle Ex. 1009, p. 713
Oracle v. DeBellis
IPR2024-00221

and the laser or inkjet printer you use. Vector-based images (such as drawings created in CorelDRAW! 6.0 or Visio 4.0) result in more attractive printed reports than do bitmapped images, especially when the bitmap is scaled. The contrast problem discussed previously may be aggravated when color images are printed in black and white.

Examining Bitmap Image File Formats

Graphics files are identified by generally accepted file extensions; these serve to define most (or all) of the format's basic characteristics. The following file extensions identify bitmap image files that have achieved the status of "industry standards" for the PC. Most commercial bitmap image editing applications support these formats, though some do not import .gif files. The "standard" extensions follow:

- *.bmp* is for Windows bitmap files in 1-, 2-, 4-, 8-, and 24-bit color depths. .bmp files contain a bitmap information header that defines the size of the image, the number of color planes, the type of compression used (if any), and information on the palette used. A *header* is a block of data in the file that precedes the image data.
- *.dib* is for device-independent bitmap files. The .dib file format is a variant of the .bmp format; to define the RGB values of the colors used, it includes a color table in the header.
- *.pcx* is for files that are compatible with ZSoft Paintbrush applications. The .pcx file format is the common denominator for most bitmap file format conversions; almost every graphics application created in the past five years can handle .pcx files, but Windows Paint cannot save files in .pcx format. (Paint refers to .pcx files as PC Paintbrush format.) .pcx files are compressed by a method called *run-length encoding (RLE)*, which can decrease the size of bitmap files by a factor of 3 or more, depending on their contents.
- *.tif* (an abbreviated form of TIFF) is for tagged image format files. The TIFF format was originally developed by Aldus Corporation and now is managed by Microsoft Corporation. Originally, TIFF files were used primarily for storing scanned images, but now they are used by a substantial number of applications (including those for Windows) as the preferred bitmap format.

  A special version of TIFF that uses file compression is used for fax transmission. TIFF files for conventional bitmaps are found in both uncompressed and compressed formats. A tag in the header of the file defines information similar to that found in the information header of .bmp files.
- *.jpg* is for files created by applications that offer compression and decompression options for Joint Photographic Experts Group (JPEG) graphics. JPEG has developed a standard methodology to compress and decompress still-color images. Special JPEG adapter cards are available to speed the compression and decompression processes.

  JPEG compression often is used for video images, especially for digital video editing, but Moving Pictures Experts Group (MPEG) compression is expected to predominate in the video field; it provides better compression ratios for video images

Oracle Ex. 1009, p. 714
Oracle v. DeBellis
IPR2024-00221

than the JPEG method. MPEG compression is used by the Hughes Direct Broadcast System (DBS) satellites that beam DirecTV and USSB programming to 18-inch microwave dishes connected to RCA or Sony set-top boxes. MPEG compression requires special PC adapter cards to display live-motion video images at the standard 30 frames-per-second rate, but Pentium PCs can use software decoding at a somewhat lower frame rate.

- *.pcd* (an abbreviation for *Photo CD*, a Kodak trademark) is for photographic images that are digitized and stored on CD-ROMs by photofinishers. Photo CD files use a special compression system devised by the Eastman Kodak Company; you need Kodak's Photo CD Access application for Windows or an image editor, such as Adobe Photoshop or Corel PhotoPAINT, to display the images and save them to .pcd files.
- *.gif* is for the graphics interchange file format used to archive bitmapped images on CompuServe and other online services. Shareware and freeware .gif file conversion applications for all popular types of personal computers are available for downloading from CompuServe's Graphic Support forum (**GO GRAPHSUP**). The.gif format has been the standard bitmap format for background images of the Internet's World Wide Web pages, partially supplanted by the use of JPEG compression.



- *.png* is for Portable Network Graphics files. PNG is a new standard proposed by the World Wide Web Consortium (W3C), a group dedicated to "realizing the full potential of the Web." Graphics stored in PNG format load faster, and can be displayed on more hardware platforms (PC, Mac, and others) without alteration of the image's qualities. (Other graphics file formats display with brighter or darker colors, or shifts in the specific color tints depending on the specific system on which you display the image.) The PNG format was developed in late 1994, and was finalized in early 1995. PNG is now the native graphic image format for Microsoft Office 97 applications.
- *.tga* is for files in the TARGA file format developed by Truevision for its TARGA product line of graphics adapter cards. TARGA cards were the first to offer relatively high-resolution, wide-spectrum color images with PCs by employing a separate video monitor.

If you don't have a Windows image-editing application with OLE file server capability, you need to convert bitmap files in .tif, .gif, .pcd, or .tga format to .bmp or .pcx format. You can use Microsoft Word's bitmap file-conversion capability to insert a .tif file in a new document and then copy the image to the Clipboard as a picture. You cannot edit the Word picture that you paste into an Access table with Paste Special, however, because the image is not an OLE object; it is a bitmap picture created from a DDE link to Word.

Oracle Ex. 1009, p. 715
Oracle v. DeBellis
IPR2024-00221

Note

If you have an OLE server application that supports the .pcx format, store individual bitmap images in .pcx and not the .bmp files used by Windows Paint. Using .pcx files saves disk space compared with the .bmp format, because .pcx uses RLE file compression. Line-art files (black on a white background) compress the most; 24-bit, full-color photographic images compress the least. The JPEG compression offered by most current Windows image editing applications provides greater compression than .pcx for color images, especially images with more than 256 colors (16-bit or 24-bit color).

If your use of images is intended primarily for printing (as in desktop publishing) and you do not have a color printer, use shades of gray for vector-based images. The 256-grayscale palette is preferred for printing bitmapped images; change color images to grayscale if your image-editing application supports this conversion.

Adding an Unbound Object Frame to a Form or Report

Instead of using OLE Object fields, images in unbound object frames store their properties as data in the area of your .mdb file that is devoted to forms or reports. Like the methods you use with bound object frames, the methods that create (or edit) unbound objects are contributed temporarily by the OLE server. After you embed or link the unbound object, the Access application supplies the methods used to display the images.

◀◀ See "Adding a Linked Subreport to a Bound Report," p. 564



The use of unbound object frames differs from that of bound object frames in the following ways:

- You set the Enabled property of most unbound object frames to No so that the OLE server that supplied those unbound objects does not appear if you double-click the object in Run (Form view) mode. The Enabled property does not affect your ability to edit the object in Design mode. A bound object's Enabled property is usually set to Yes (the default value when the bound object frame is created).
- Unbound object frames have properties (such as Row Source, Link Child Fields, Link Master Fields, and Column Count) that are not applicable to bound object frames. Graphs and other unbound objects use these properties to obtain or present data in an unbound object field.
- You can create a master-child linkage between the content of an unbound object frame and the value of a field in the underlying table or query (or the value entered in a text box on your form).

Oracle Ex. 1009, p. 716
Oracle v. DeBellis
IPR2024-00221

- Multimedia objects, such as sound, video, or animated graphics, are often contained in unbound object frames.

The following sections provide examples that use the important additional properties that are available when using unbound object frames.

Creating a New Logo for Your Personnel Actions Form

Logotypes and symbols that identify an organization are among the most common graphic objects on forms and in reports. The bitmap example in this section uses the image of a lighthouse from the Northwind Traders database, but you can substitute your organization's logo if you have a bitmap file or a scanner to create an image of suitable dimensions. Creating a logo from a bitmap file is the subject of the "Using the Image Control" section of this chapter.

The easiest way of adding an image to a new form is to copy to the Clipboard an existing unbound object frame of a form or report that contains the image. Then duplicate the image by pasting it into another form. This process is similar to the process for copying OLE objects that are to be used by other applications. You can edit the image as necessary with Windows Paint or another OLE-compliant image editor.

Northwind Traders' Forms Switchboard form contains an image that can easily be adjusted to fit on your Personnel Actions Entry form. However, the bitmap is contained in an image control, so the process is a bit more complicated than simply copying an unbound object frame from one form to another. Before adding the graphic image to the Personnel Actions Entry form, you should make a backup copy of the form by following these steps:



1. Click the Forms tab of the Database window and then click the Copy button on the toolbar.



2. Click the Paste toolbar button; Access displays the Paste As dialog.



3. Type a name for the copy of the form, such as **frmPersonnelActionsEntryCopy**, and then click OK. Access creates a copy of the form.

Follow these steps to copy the Northwind Traders logo from the Forms Switchboard to your Personnel Actions Entry form:



1. Click the Forms tab of the Database window, and open the frmPersonnelActionsEntry form. Rearrange the fields at the top of the form to make room for the logo; refer to Figure 20.9 for the sizing and placement of fields.



◀◀ See "Selecting, Editing, and Moving Form Elements and Controls," p. 419

Oracle Ex. 1009, p. 717
Oracle v. DeBellis
IPR2024-00221

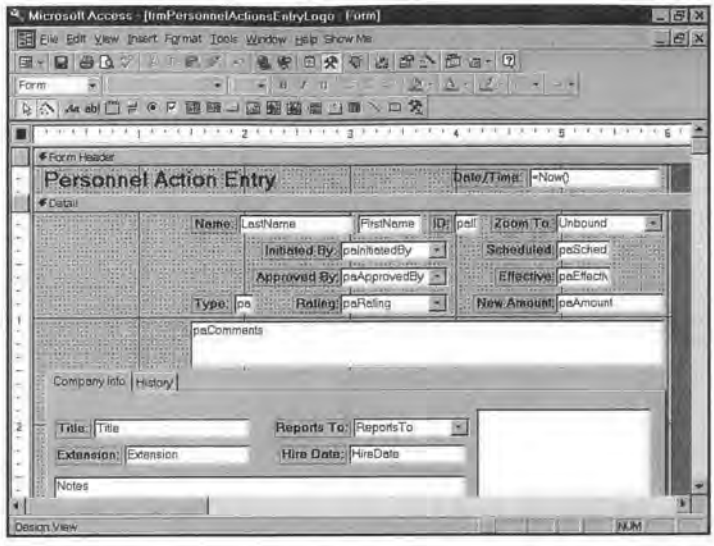


FIG. 20.9 The Personnel Actions Entry form with the main form controls rearranged to make room for the new logo.



2. Switch to the Database window, click the Forms tab, and choose the Main Switchboard form from the list; click the Design button to open the form in Design view.
3. Select the Northwind logo by clicking the lighthouse in the logo. The logo is an image control, which was new to Access 95. Image controls are the subject of the section "Using the Image Control," which follows shortly.
4. Press Ctrl+C or choose Edit, Copy to copy the logo's bitmap to the Clipboard. Close the Forms Switchboard form.
5. Click the Detail header bar of the Personnel Actions Entry form to select the Detail section of the form.
6. Choose Insert, Object to display the Insert Object dialog. Select Bitmap Image from the Object Type list, making sure that the Create New option button is selected (see Figure 20.10).
7. Click OK to open Paint with an empty image, and then press Ctrl+V or choose Edit, Paste to paste the image into Paint's window (see Figure 20.11). Click outside of the image to expose the bitmap sizing handles.
8. Click and drag the middle-bottom sizing handle to eliminate the white (empty) region below the image. Repeat the process with the middle-right sizing handle to eliminate the empty region to the right of the image (see Figure 20.12).

Oracle Ex. 1009, p. 718
Oracle v. DeBellis
IPR2024-00221

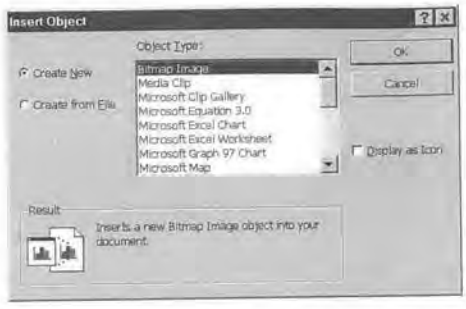


FIG. 20.10 The Insert Object dialog settings for opening an instance of Paint.

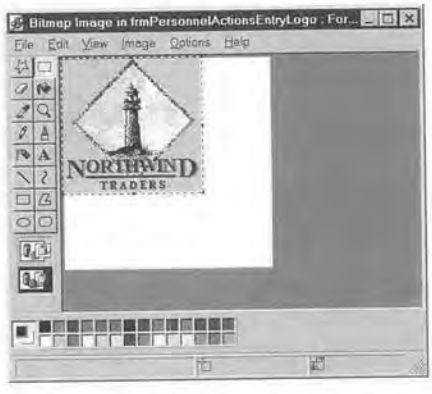


FIG. 20.11 Copying the bitmap from the Northwind logo into the open instance of Paint.

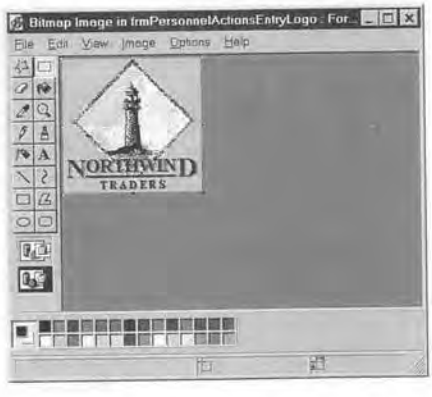


FIG. 20.12 The logo bitmap with white (empty) regions removed.

Oracle Ex. 1009, p. 719
Oracle v. DeBellis
IPR2024-00221

9. Choose File, Save Copy As and save a copy of the bitmap as **Nwind.bmp** in your working folder (usually C:\My Documents). You'll use this bitmap file later with the image control.

10. Choose File, Exit & Return to frmPersonnelActions:Form to close Paint and place the unbound object frame in your form. The default position of the frame is at the upper left of the selected section of your form (see Figure 20.13).

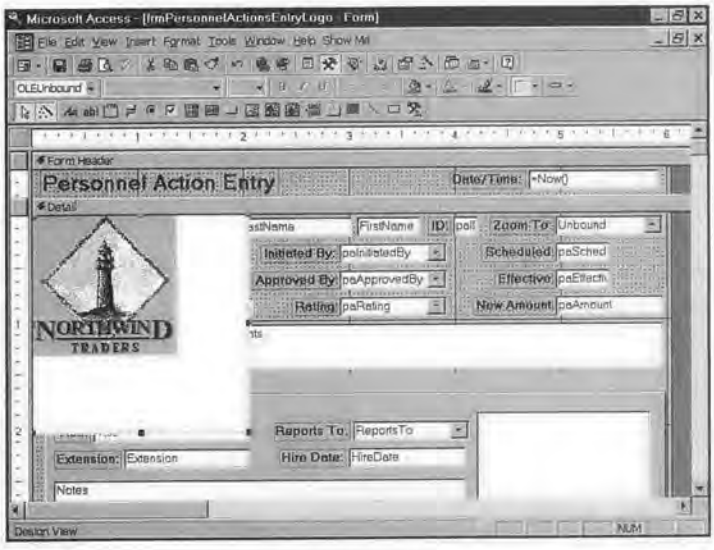


FIG. 20.13 The unbound object frame for the logo added to the Detail section of the Personnel Actions Entry form.

Access applies the Size Mode property to the presentation of the image; Size Mode does not modify the image itself. The copied image covers several controls. This problem is corrected when you manipulate the image in the next section.

Sizing and Setting the Properties of the Unbound Object Frame

Sizing and setting the properties of unbound object frames is similar to the process for bound object frames. To finish adding the logo to your form, follow these steps:

1. Click and drag the unbound frame so that the top-left corner is about two grid marks from both the top and left edges of the Detail section of the form.

2. Use the bottom-middle and right-middle sizing handles to adjust the size of the frame to fit the available space (see Figure 20.14).

3. Open the Properties window and set the Size Mode property to Zoom for scaling the bitmap to the available space while retaining the original aspect ratio.



Oracle Ex. 1009, p. 720
Oracle v. DeBellis
IPR2024-00221

4. You don't need a border around the logo, so set the Special Effect property to Flat, the Border Style property to Transparent, and the Back Color property to light gray (12632256), as shown in Figure 20.15.



5. Click the Form View button to display the logo added to the Personnel Actions Entry form (see Figure 20.16).

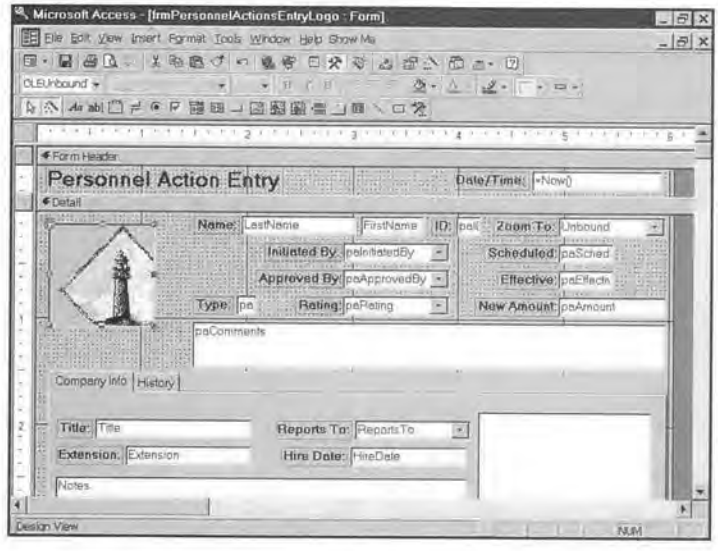


FIG. 20.14 Sizing the clipped bitmap image in the unbound object frame to the available space.

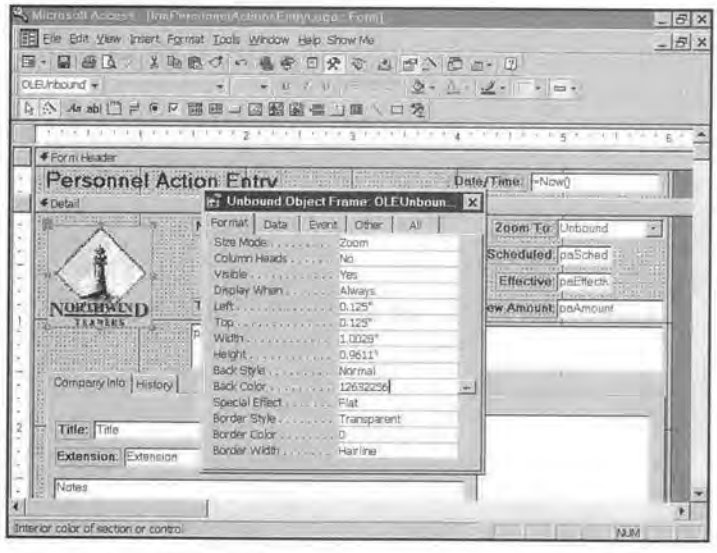


FIG. 20.15 Setting the properties for the Northwind Traders logo.

Oracle Ex. 1009, p. 721
Oracle v. DeBellis
IPR2024-00221

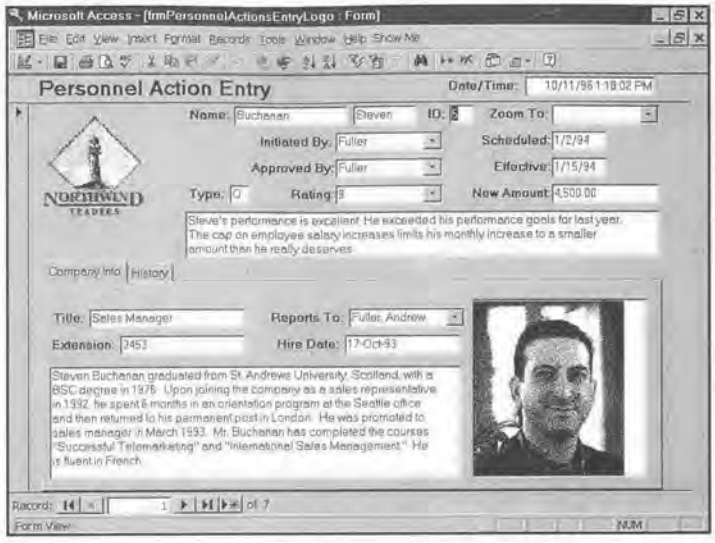


FIG. 20.16 The Northwind Traders logo in Form view.

Using the Image Control

Versions of Access prior to Access 95 required use of object frames to contain bitmapped images. Access 95 and 97 add the image control—similar to the image control of Visual Basic 4.0—to display bitmap (.bmp), device-independent bitmap (.dib), and Windows Metafile Format (.wmf) or enhanced metafile (.emf) vector images in forms and reports. To substitute the image control for the unbound object frame of your Personnel Actions Entry form, follow these steps:

1. Open frmPersonnelActions in Design view and select the logo. Press Delete to delete the unbound object frame from the form.



2. Click the Image button of the toolbox and draw the image control to the same dimensions as the deleted unbound object frame. The Insert Picture dialog appears.



3. Maneuver to the folder in which you stored the Nwind.bmp file you created with Paint in the section "Creating a New Logo for Your Personnel Actions Form" earlier in this chapter (see Figure 20.17). If you like another image better, you can substitute its .bmp file in this example.
4. Double-click Nwind.bmp or select Nwind.bmp and click OK. Your image control appears, as shown in Figure 20.18. Unlike object frames, image controls by default center clipped images in their frame. Image frames have a Picture Alignment property that lets you choose how the image is aligned.
5. Image controls have default Special Effect, Border Style, and Back Color properties that are suitable for adding logos to forms. Thus, you need only change the Size Mode property of the image control to Zoom. Figure 20.19 shows the result of the preceding steps in Form view.

Oracle Ex. 1009, p. 722
Oracle v. DeBellis
IPR2024-00221



FIG. 20.17 Selecting the Nwind.bmp file created earlier in the chapter for the image control.

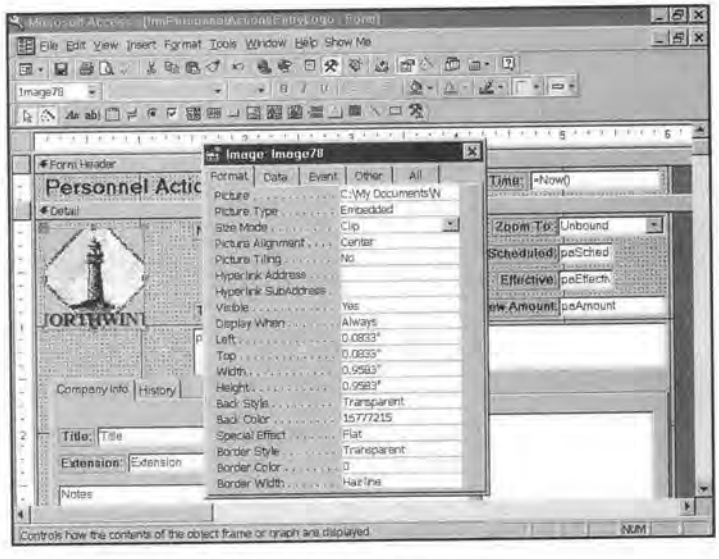


FIG. 20.18 The clipped Northwind logo in an image frame.

If you don't need to edit a static image from within your Access application, image controls are preferred over unbound object frames for display images of the file types supported by the image control. Image controls respond faster and use fewer Windows resources than unbound object frames.

Image controls also have a variety of additional properties that you can use to customize the presentation of the image. One of the most useful of these new properties is the Palette property that you can set to a Windows palette file (.pal) or a variety of other image file types to provide a custom set of 240 colors that enhance the appearance of 256-color bitmaps. Like its predecessors, Access 97 is limited to displaying 256-color bitmaps. Third-party, 32-bit ActiveX controls designed for image presentation and editing let you use high-color (16-bit color depth) and full-color (24-bit color depth) bitmaps.

Oracle Ex. 1009, p. 723
Oracle v. DeBellis
IPR2024-00221

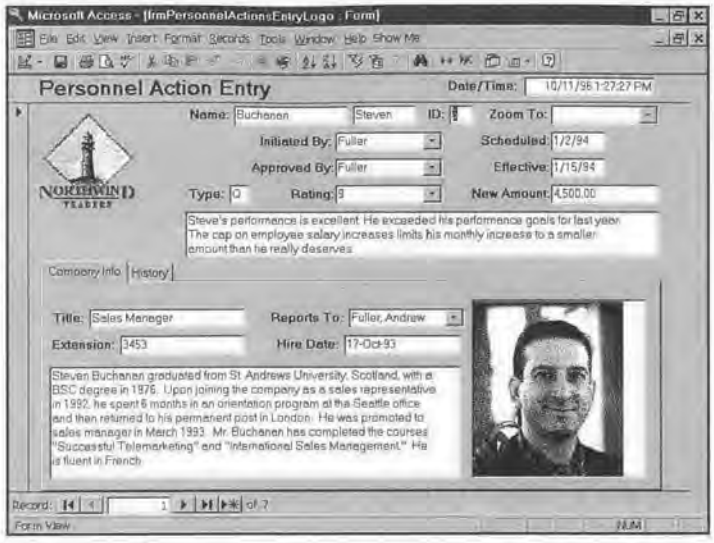


FIG. 20.19 The Northwind logo zoomed in an image frame.

Creating Graphs and Charts with Microsoft Graph 97

Microsoft Graph 97 (called in this book by its Registry programmatic ID, MSGraph8) is a 32-bit OLE 2.1 miniserver application. MSGraph8 is the 32-bit replacement to MSGraph5; MSGraph8 is an OLE mini-server that provides graphing capabilities to all Microsoft productivity applications. The sections that follow describe how to use the Graph Wizard to add MSGraph8 graphs and charts to Access 97 forms and reports.

Creating the Query on which to Base the Graph

Most graphs required by management are the time-series type; these track the history of financial performance data, such as orders received, product sales, gross margin, and the like. In smaller firms, this data comes from tables that store entries from the original documents (such as sales orders and invoices) that underlie the summary information.

This type of data often is called a *line-item source*. Because a multibillion-dollar firm can accumulate several million line-item records in a single year, larger firms usually store summaries of the line-item source data in tables; this technique improves the performance of queries. Summary data is referred to as *rolled-up* data or, simply, *rollups*. Rollups of data on mainframe computers often are stored in client/server RDBMSs running under UNIX or Windows NT to create *data warehouses*. Although rolling up data from relational tables violates one of the guiding principles of relational theory—don't duplicate data in tables and don't store derived data in tables—databases of rolled-up data are very common.

Oracle Ex. 1009, p. 724
Oracle v. DeBellis
IPR2024-00221

Note

The Chart Wizard of Access 95 and Access 97 differs substantially from the Graph Wizard of Access 2.0 and its predecessors. The Chart Wizard creates a crosstab query for you; the older Graph Wizard required you to create your own crosstab query. This section describes how to use the Chart Wizard in the manner Microsoft intended: from a table or SELECT query. The section "Creating a Graph from a Crosstab Query" later in this chapter shows you how to change the Row Source property of a graph object to use an existing crosstab query as the source of the data.

Northwind Traders is a relatively small firm, so it isn't necessary to roll up line-item data to obtain acceptable query performance on an Intel 80486-based or faster computer. To create a query designed specifically for the Chart Wizard, follow these steps:

Queries

New

1. In the Database window, open a new query and add the Categories, Products, Order Details, and Orders tables to the query. Joins are created for you between the primary-key and foreign-key fields of each table.
2. Drag the CategoryName field of the Categories table to the first column.
3. Drag the UnitPrice field of the Order Details table to the second column. Edit the Field row of the column to read **Amount: [Order Details].[UnitPrice]* [Order Details].[Quantity]**.
4. Drag the ShippedDate field of the Orders table to the third column. Add an ascending sort on this column.
5. Add the criterion **Like "*/*/95"** to the ShippedDate column so as to only include 1995 orders. The year 1995 is used instead of 1996 in this example because data is available for all 12 months of 1995. Your query design appears, as shown in Figure 20.20.

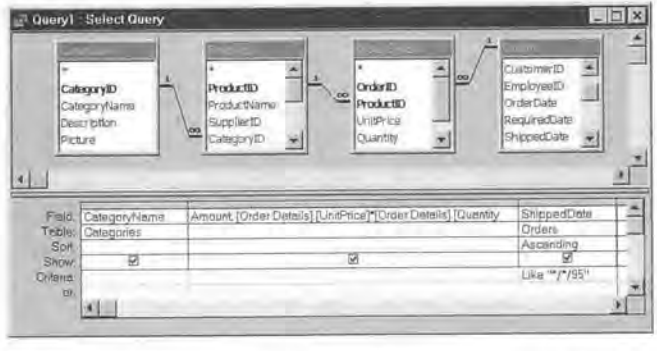


FIG. 20.20 A query design for creating a graph with Access 97's Chart Wizard.

6. Click the Run button to test your query (see Figure 20.21.) Close the query and save it with the name **qryChartWizard**.

Oracle Ex. 1009, p. 725
Oracle v. DeBellis
IPR2024-00221

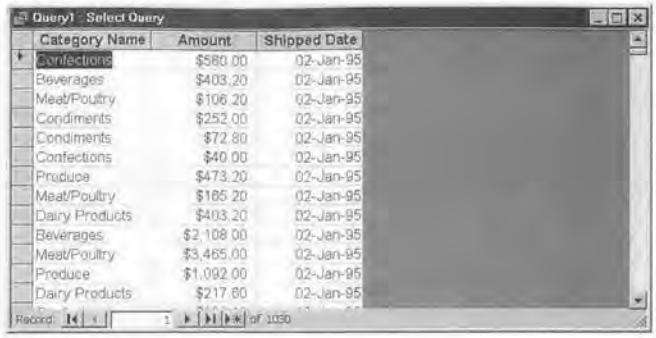


FIG. 20.21 The query result set of the query in Figure 20.20.

Using the Chart Wizard to Create an Unlinked Graph

Although it's possible to create a graph or chart by using the Insert Object method and selecting the Microsoft Graph 97 object type, the Graph Wizard makes this process much simpler. You can use the Graph Wizard to create two different classes of graphs and charts:

- *Unlinked* (also called *non-linked*) line graphs display a line for each of the rows of your query. You can also create unlinked stacked-column charts and multiple-area charts.
- A *linked* graph or chart is bound to the current record of the form on which it is located and displays only a single set of values from one row of your table or query at a time.

This section shows you how to create an unlinked line graph based on a query. The next section describes how to use MSGraph8 to display alternative presentations of your data in the form of bar and area charts. In the last section of this chapter, you create a graph that is linked to a specific record of a query result set.

To create an unlinked graph that displays the data from the qryChartWizard query, follow these steps:



1. Open a new form, select Chart Wizard in the list box, and select qryChartWizard in the drop-down list (see Figure 20.22). Click OK to launch the first dialog of the Chart Wizard.
2. Click the >> button to add all three fields to your graph (see Figure 20.23). Click the Next button to display the Chart Wizard's second dialog.
3. Click the Line Chart button (the third from the left in the third row of the buttons that display the available graph styles), as shown in Figure 20.24. Click the Next button to display the third Chart Wizard dialog.
4. The Chart Wizard attempts to design a crosstab query based on the data types of the query result set. In this case, the Chart Wizard makes a mistake by assuming you want months in the legend box and product categories along the graph's horizontal x-axis (see Figure 20.25).

Oracle Ex. 1009, p. 726
Oracle v. DeBellis
IPR2024-00221

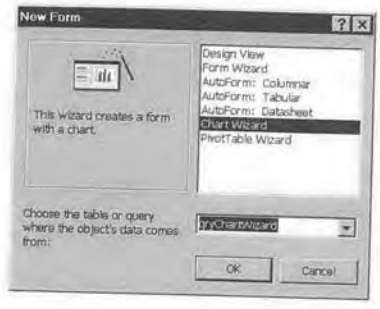


FIG. 20.22 Selecting the Chart Wizard and the query on which to base the graph.

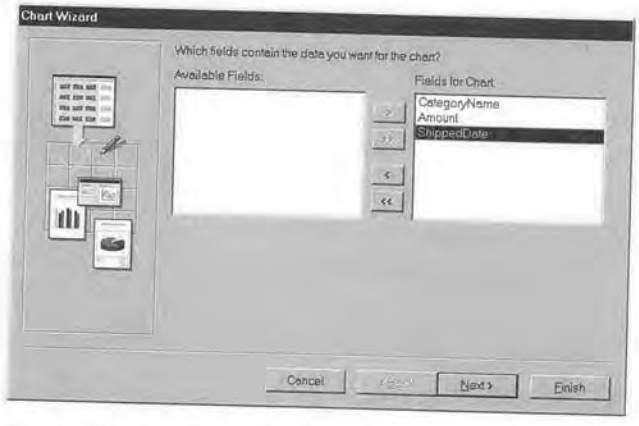


FIG. 20.23 Selecting the fields to include in the graph.



FIG. 20.24 Selecting the type of graph or chart.

Oracle Ex. 1009, p. 727
Oracle v. DeBellis
IPR2024-00221

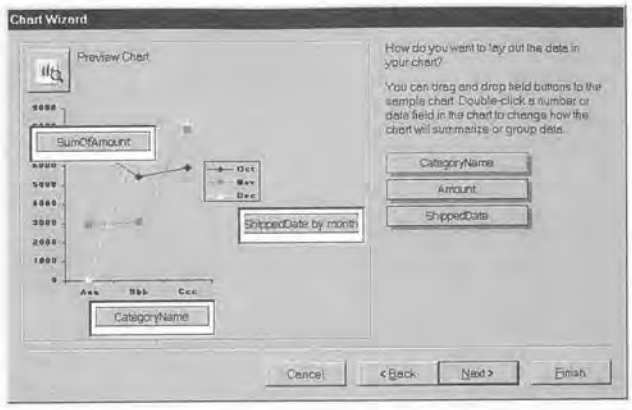


FIG. 20.25 The Chart Wizard's first try at guessing the type of crosstab query to create.

5. You want the categories in the legend and the months of 1995 across the x-axis. Drag the CategoryName button from the right side of the dialog to the drop box under the legend, and drag the ShippedDate button to the drop box under the x-axis. The button title, partly obscured, is ShippedDate by month (see Figure 20.26). You can double-click the ShippedDate by month button and select from a variety of `GROUP BY` date criteria, ranging from Year to Minute. Click the Next button to go to the fourth and final Chart Wizard Dialog.

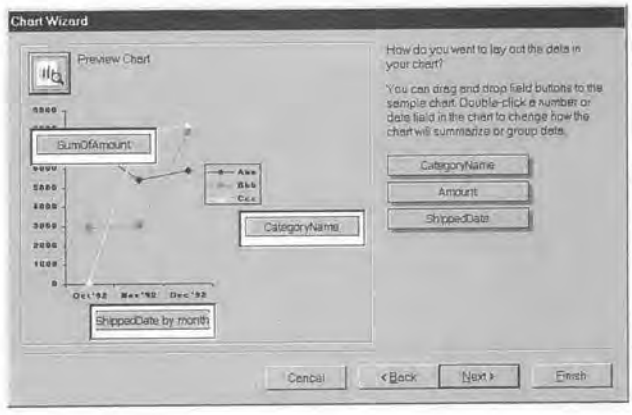


FIG. 20.26 Correcting the Chart Wizard's crosstab query guesswork.

6. Type **1995 Monthly Sales by Category** in the text box to add a title to your graph. Click the Yes, Display a Legend option button, if necessary, to display the Category Name legend (see Figure 20.27). Accept the remainder of the defaults.

Oracle Ex. 1009, p. 728
Oracle v. DeBellis
IPR2024-00221



FIG. 20.27 Adding a title and legend to your graph.

7. Click the Finish button to display your graph in Form view. In the miniature version illustrated by Figure 20.28, some month labels are missing and the legend crowds the graph and label. You fix these problems in the next section of this chapter, "Modifying the Design Features of Your Graph."

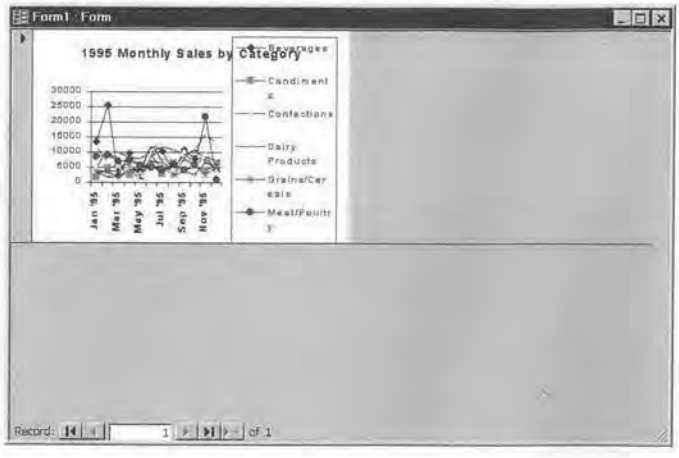


FIG. 20.28 The unbound graph in Form view, as completed by the Chart Wizard.



8. Click the Design View button of the toolbar and increase the size of your graph to at least 5.5 inches wide by 2.5 inches high (see Figure 20.29). The graph that appears in Design View at this point is an example graph included with MSGraph8.

9. Set the Enabled property of the unbound object frame to Yes and the Locked Property to No.

10. The chart is in an unbound object frame, so you don't need form adornments for record manipulation. Select the form and set the Scroll Bars property of the form to Neither, the Record Selectors to No, and the Navigation Buttons to No.

Oracle Ex. 1009, p. 729
Oracle v. DeBellis
IPR2024-00221

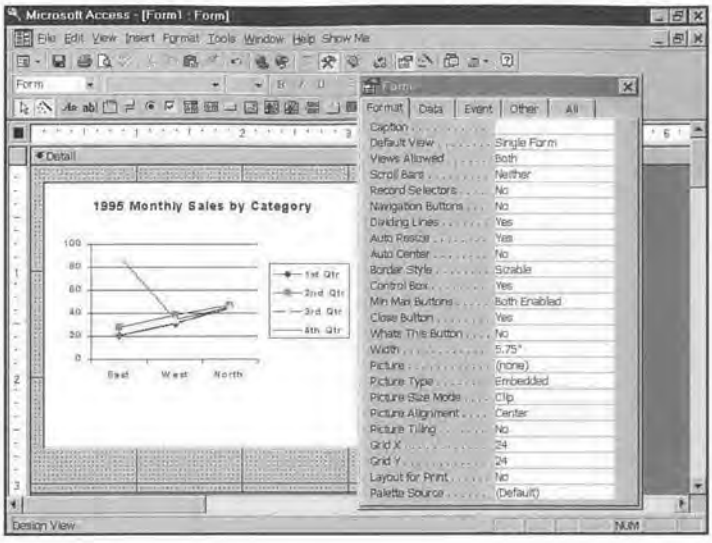


FIG. 20.29 The expanded object frame in Form Design view.

11. Use the sizing handles of the unbound object frame to create a 1/8-inch form border around the frame. Leaving a small form area around the object makes the activation process more evident.

12. Save your form with a descriptive name, such as **frmChartWizard**. Return to Form view in preparation for changing the size and type of your graph.



Tip

When you complete your design, set the value of the Enabled property to No so that users of your application can't activate the graph and alter its design.

Modifying the Design Features of Your Graph

MSGraph8 is an OLE 2.1 mini-server; you can activate MSGraph8 in place and modify the design of your graph. MSGraph8 also supports Automation, so you can use Access VBA code to automate design changes. This section shows you how to use MSGraph8 to edit the design of the graph manually, as well as how to change the line graph to an area or column chart.

▶▶ See "Understanding Automation," p. 1097



To activate your graph and change its design with MSGraph8, follow these steps:

1. Display the form in Form view, then double-click the graph to activate MSGraph8 in place. A diagonally hashed border surrounds the graph; MSGraph8's menus replace and supplement those of Access 97. (The activation border is missing from

Oracle Ex. 1009, p. 730
Oracle v. DeBellis
IPR2024-00221

the left and top of the object frame if you didn't create some space on the form around the object frame in step 11 of the preceding section.)

2. Drag the middle-right sizing handle to the right border of your enlarged unbound object frame. Drag the middle-bottom sizing handle to the bottom border of the object frame.

3. Choose View, Datasheet to inspect the data series that Access has transmitted to MSGraph8 (see Figure 20.30).

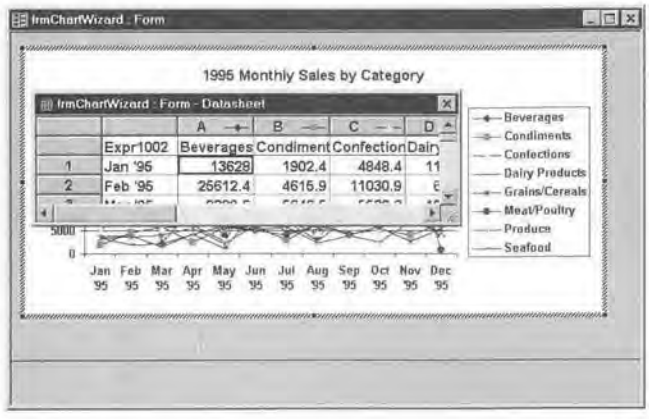


FIG. 20.30 The expanded version of the chart displaying part of the datasheet for the graph.

4. You can change the type family and font size of your chart's labels and legend. Select the graph title, and then choose Format, Selected Chart Title to open the Format Chart Title dialog. Select the Font tab, and set the size of the chart title to 12 points (see Figure 20.31), and then click OK to close the dialog. Select the legend, and then open the Format Legend dialog and set the size of the legend font to 7 points.

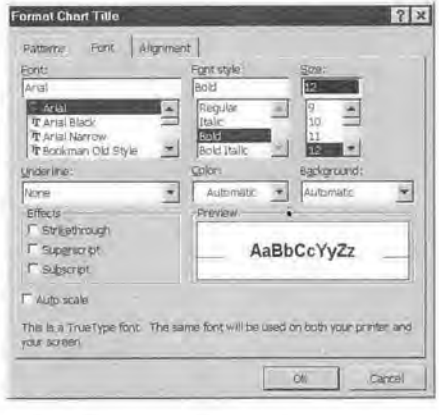


FIG. 20.31 Changing the font size of the graph's title.

Oracle Ex. 1009, p. 731
Oracle v. DeBellis
IPR2024-00221

5. The y-axis labels should be formatted as currency, so click one of the labels to select the y-axis. Then choose Format, Selected Axis to display the Format Axis dialog.

6. Click the Number tab, select Currency formats in the Category list, and enter **0** in the Decimal Places text box (see Figure 20.32). Click OK to close the dialog and apply the new format. Your line graph appears as shown in Figure 20.33.



FIG. 20.32 Formatting the numeric values of the y-axis.

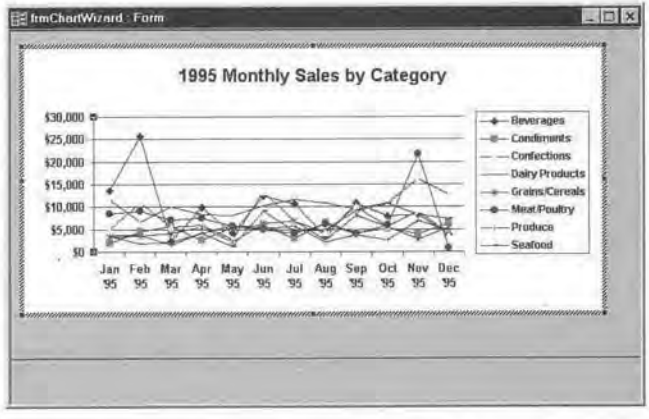


FIG. 20.33 The line graph with reformatted y-axis labels and a larger graph title.

You might want to change the line graph to some other type of chart (such as area or stacked column) for a specific purpose. Area charts, for example, are especially effective as a way to display the contribution of individual product categories to total sales. To change the line graph to another type of chart, follow these steps:

1. Choose Chart, Chart Type to open the Chart Type dialog.

2. Click the Standard Types tab to display the standard chart selections, and select Area in the Chart Type list (see Figure 20.34).

Oracle Ex. 1009, p. 732
Oracle v. DeBellis
IPR2024-00221

3. Select Stacked Area chart in the Chart Sub-Type list (the second chart in the first row—see Figure 20.34). Click OK to change your line graph into an area chart, as shown in Figure 20.35. The contribution of each category appears as an individually colored area, and the uppermost line segment represents total sales.



FIG. 20.34 Changing the line graph to a stacked area chart.

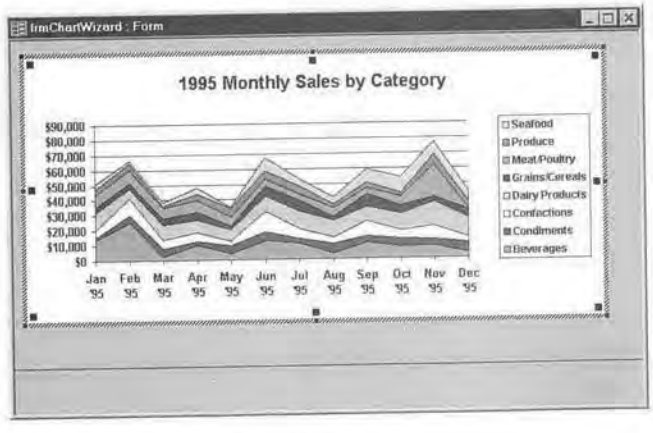


FIG. 20.35 The stacked area chart in Form view.

Tip

To get help with the Chart Type dialog, click the Office Assistant button at the lower left of the Chart Type dialog.

4. To convert the area chart into a stacked column chart, choose Chart, Chart Type, display the Standard Types page of the Chart Type dialog, select Column in the

Oracle Ex. 1009, p. 733
Oracle v. DeBellis
IPR2024-00221

Chart Type list, and then select Stacked Column (the second button in the first row) of the Chart Sub-Type list (see Figure 20.36).

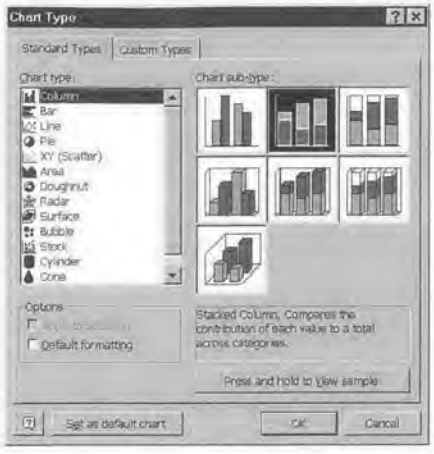


FIG. 20.36 Selecting a stacked column chart type.

5. Click OK to close the Chart Type dialog. Your stacked column chart appears, as shown in Figure 20.37.

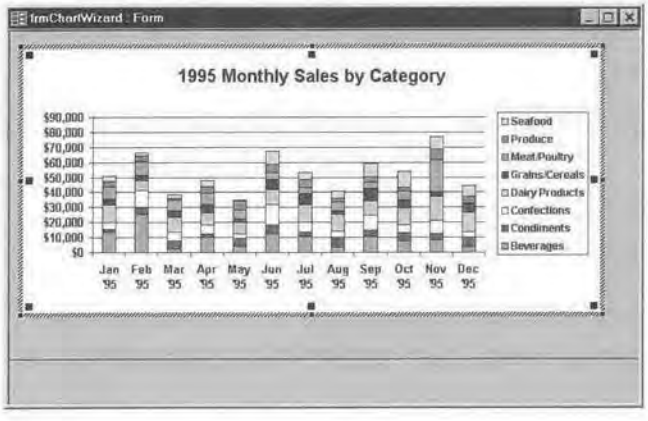


FIG. 20.37 The stacked column chart in Form view.

6. Another subtype of the area chart and stacked column chart is the *percentage distribution* chart. To create the distribution of the sales graph shown in Figure 20.38, repeat steps 4 through 5, but select the 100% Percent Column picture with equal column heights in the Chart Sub-type list of the Chart Type dialog.

7. Because you set the format of the y-axis previously to eliminate the decimals, you need to change the format of the y-axis manually to Percentage. Select the y-axis, choose Format, Selected Axis, select Percentage in the Category list, make sure that

Oracle Ex. 1009, p. 734
Oracle v. DeBellis
IPR2024-00221

Decimal Places is set to **0**, and then click OK to apply the format. Your chart appears as shown in Figure 20.38.

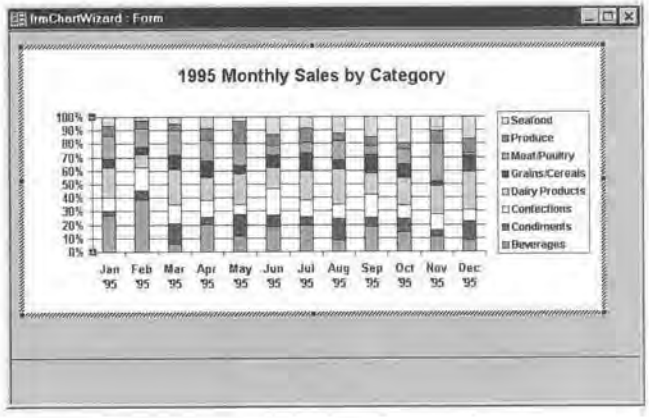


FIG. 20.38 The percentage distribution column chart.

8. Change the Chart Type back to a line graph in preparation for the linked graph example of the next section, "Creating a Graph from a Crosstab Query." Click inside the form border but outside the object frame to deactivate the graph, and save your form. Of the four types of charts demonstrated, most users find the area chart best for displaying time-series data for multiple values that have meaningful total values.

> **Note**
>
> The process of adding an unbound graph to an Access report is identical to that for forms. Unless you have a color printer, you need to select a line graph subtype that identifies data points with a different symbol for each category. For area and stacked column charts, select a series of hatched patterns to differentiate the product categories.

Creating graphs with MSGraph8 is not a speedy process. Each time you change from Design view to Form view, Access runs the query, launches MSGraph8 in the background, and passes the query data to MSGraph8. This process, using the qryChartWizard query, takes about 13 to 15 seconds on an 80486DX4/100 computer with a PCI bus video accelerator and a fast fixed-disk drive. The 32-bit graph OLEControl of Visual Basic 4.0 is much faster in operation than MSGraph8, but you must master programming Automation code in Access VBA to use Graph32.ocx.

Another problem with MSGraph8 is that you cannot easily transfer data to MSGraph8's datasheet with Access VBA code. (The `Cells` and `Range` properties of MSGraph8's `DataSheet` object are read-only, so you can't use Automation to set data series values directly.) You can expect third-party suppliers to provide faster-operating graphing applications in the form of OLE 2.x miniservers and ActiveX controls. You need to write Access VBA code, however, to send the data series to these applications.

Oracle Ex. 1009, p. 735
Oracle v. DeBellis
IPR2024-00221

Creating a Graph from a Crosstab Query

Access 97's Chart Wizard is quite parochial: It insists on creating a crosstab query for you. Once you've created a chart with the Chart Wizard, however, you can change the Row Source property value to specify a previously created crosstab query of your own design. Thus, you can take one of the queries you created in Chapter 10, qry1995QuarterlyCategorySales, and modify this query to create the qry1995MonthlyCategorySalesChart query that provides the same result as the crosstab query created by the wizard. You need to specify a table or query as the Record Source for the chart to create the linked chart described in the next section, "Linking the Graph to a Single Record of a Table or Query."

◀◀ See "Creating a Monthly Product Sales Crosstab Query," p. 362
◀◀ See "Decreasing the Level of Detail in Crosstab Queries," p. 365



Note

You must create the qry1995MonthlyCategorySalesChart query and use the query as the Row Source of the unbound object frame to complete the linked graph example in the following section. The linked graph example does not work with the crosstab query created by the Chart Wizard in the preceding steps.

To create the qry1995MonthlyCategorySalesChart query, follow these steps:



1. Open the qry1995MonthlyCategorySales query in Design mode.
2. Click the Show Table button of the toolbar and add the Categories table to the query. A join is created between the CategoryID fields of the Categories and Products tables.
3. Delete the CategoryID column that provided the Row Heading.
4. Drag the CategoryName field from the Categories table to the first column of the query, and select RowHeading in the Crosstab row.
5. Alias the CategoryName field by typing **Categories:** at the beginning of the field text box.
6. Change the Expr1 statement in the Fields row to **Expr1:Format([OrderDate], "mmm")** to use three-letter month abbreviations. Your query appears as shown in Figure 20.39.
7. Double-click an empty region of the upper pane to open the Query Properties window's General page. Delete the current entry in the Column Headings text box of the General page. Enter the 12 month abbreviations, **Jan** through **Dec**, separating the month abbreviations with commas. Access adds the double quotation marks for you, as shown in Figure 20.40.

Oracle Ex. 1009, p. 736
Oracle v. DeBellis
IPR2024-00221

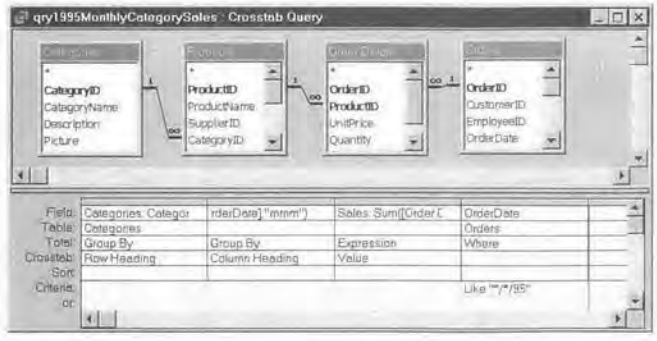


FIG. 20.39 The design of a query that displays monthly sales by product category.



FIG. 20.40 Adding fixed column headings for 12 months.

8. Choose File, Save As, and save your query as **qry1995MonthlyCategorySalesGraph**.



9. Click the Run button of the toolbar to check your query result set (see Figure 20.41).



FIG. 20.41 Part of the result set of the qry1995MonthlyCategorySalesChartquery.



10. Open frmChartWizard in Design view, if necessary, select the unbound object frame, and open the Properties window.

Oracle Ex. 1009, p. 737
Oracle v. DeBellis
IPR2024-00221

11. Click the Data tab, open the Row Source list box, and select qry1995MonthlyCategorySalesChart as the value of the Row Source property.
12. Change to Form view and verify that your line graph appears the same as that created by the Chart Wizard.



Troubleshooting

After changing the Row Source property of my chart to the qry1995MonthlySalesCategoryChart query, the product categories appear in the chart as the x-axis labels, and the total month sales are plotted on the chart.

MSGraph8 might determine that the column headings of the Row Source query are the series to be plotted, resulting in the Category Name appearing as the x-axis legends. To obtain the correct data plotting, display the chart in Form view, and double-click it to activate in-place editing with MSGraph8. Next, choose Data, Series in Rows to change the series markers from the column headings to the rows; this causes the dollar amounts to be plotted on the chart, and the months of the year appear as the x-axis legend.

Linking the Graph to a Single Record of a Table or Query

You create a linked graph or chart by setting the values of the MSGraph8 object's Link Child Fields and Link Master Fields properties. The link is similar to that between a form and subform. A linked graph displays the data series from the current row of the table or query that serves as the Record Source of the form. As you move the record pointer, the graph is redrawn to reflect the data values in the selected row.

To change the frmChartWizard form to accommodate a linked graph, follow these steps:

1. Open frmChartWizard in Design view; then click the Properties button of the toolbar to open the Properties window for the form.



2. Click the Data tab, open the Record Source list box, and select qry1995MonthlyCategorySalesChart as the value of the Record Source property of the form, which binds the form to the query.



3. Your form needs record-navigation buttons for a linked query, so click the Format tab and set the value of the Navigation Buttons property to Yes.
4. Select the unbound object frame, and then click the Data tab. Type **Categories** as the value of the Link Child Fields and Link Master Fields properties (see Figure 20.42). Using this technique, you create the link between the current record of the form and the row of the query that serves as the Row Source property of the graph (through the Categories field of the query).
5. To test your linked graph, click the Form View button of the toolbar. If (in the preceding section) you saved the line graph version of the form, your graph appears as shown in Figure 20.43.



Oracle Ex. 1009, p. 738
Oracle v. DeBellis
IPR2024-00221



FIG. 20.42 Linking the graph's Row Source property to the current record of the form.



FIG. 20.43 The linked version of the 1995 Monthly Sales by Category graph.

6. The single line appears a bit anemic for a graph of this size, so double-click the graph to activate it in place. Double-click anywhere on the line to display the Format Data Series dialog. Click the Patterns tab to bring that page to the front of the dialog. Open the Weight drop-down list, and choose the thickest line it offers. To change the data-point marker, open the Style drop-down list and select the square shape. Use the drop-down lists to set the Foreground and Background colors of the marker to a contrasting hue, such as red (see Figure 20.44). Click OK to close the dialog and implement your design changes.

7. Double-click the legend box to open the Format Legend dialog. On the Patterns page, click the None option button in the Border frame to remove the border from the legend (see Figure 20.45). Click the Font tab, set the Bold attribute on, and change the font size to 11 points. Click OK to close the dialog and apply your modification to the legend.

Oracle Ex. 1009, p. 739
Oracle v. DeBellis
IPR2024-00221

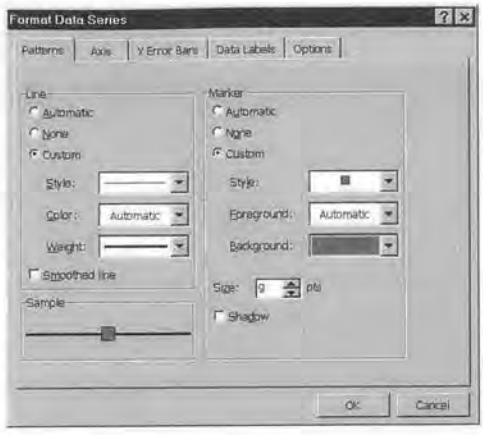


FIG. 20.44 Increasing the thickness of and changing the markers for the data series line.

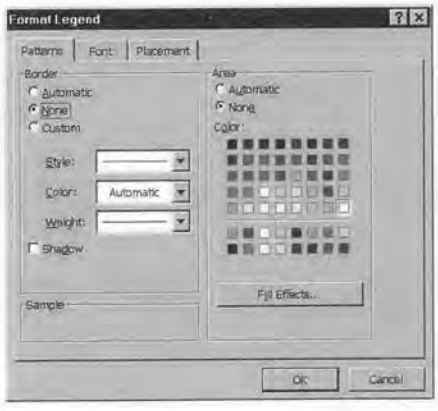


FIG. 20.45 Removing the border from the legend.

8. To use your enhanced legend as a subtitle for the chart, click and drag the legend to a location under the chart title, as shown in Figure 20.46. Click over the plot area to display the chart's sizing handles; drag the middle sizing handle to the right to increase the size of the plot area (see Figure 20.46).

9. Click anywhere on the form outside of the chart to close MSGraph8 and return to Access. Click the record selection buttons to display a graph of the sales for each of the eight categories.

10. You can smooth the data series line by activating the graph and double-clicking the line. Mark the Smoothed Line check box in the Line frame of the Patterns page, and then click OK. Your graph now appears as shown in Figure 20.47.

Oracle Ex. 1009, p. 740
Oracle v. DeBellis
IPR2024-00221

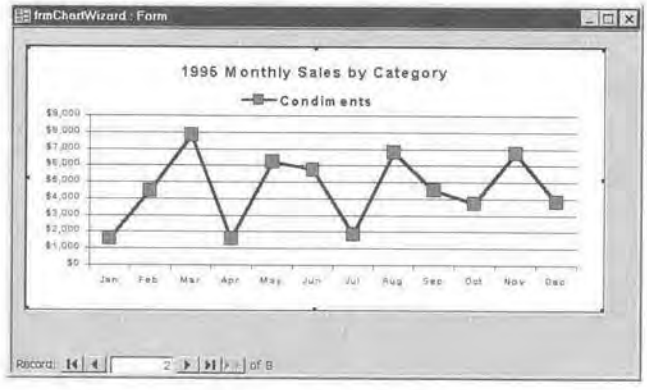


FIG. 20.46 The Form view of the graph with added design features.

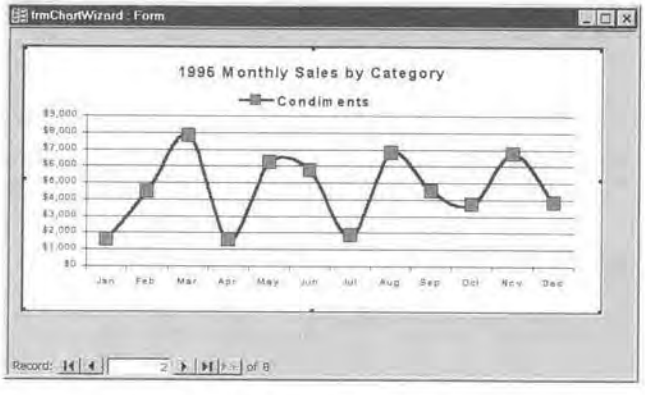


FIG. 20.47 Smoothing applied to the data series line, using best-fit curves.

11. MSGraph8 offers a variety of three-dimensional chart formats. Figure 20.48 illustrates a 3-D column chart. You can change the perspective of the graph by activating the chart and selecting the Corners part of the chart. Click one of the Corners' selection squares, and drag the square to change the perspective.

Tip

Use the Chart Objects list (at the top left of the MSGraph8 toolbar) to select objects in the chart, such as the corners, walls, series, and other elements of a bar chart.

Oracle Ex. 1009, p. 741
Oracle v. DeBellis
IPR2024-00221

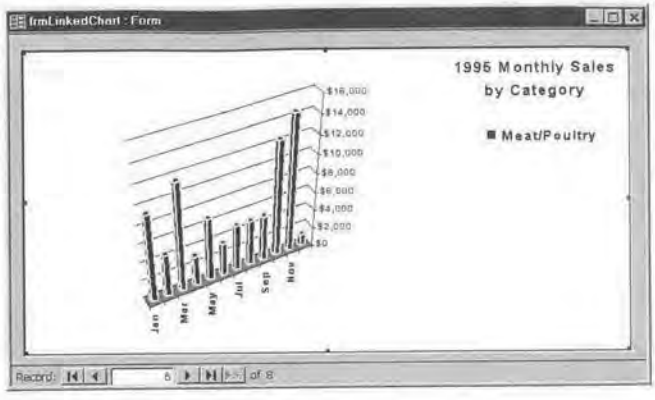


FIG. 20.48 The line graph converted to a 3-D column chart.

12. Deactivate the graph by clicking the form outside the bound object frame. Choose File, Save As, and save your bound form with a new name, such as **frmLinkedChart**.

Oracle Ex. 1009, p. 742
Oracle v. DeBellis
IPR2024-00221

Chapter 21

Using Access with Microsoft Excel

Spreadsheet and word-processing applications dominate the Windows productivity application software market. According to industry reports, more than 70 percent of all Windows installations in the business environment include a spreadsheet application. Microsoft Office 97, which includes 32-bit Excel 97 and Word 97, is likely to take over the dominant position of 16-bit Microsoft Office 4+ as Windows 95 and Windows NT 4.0 Workstation gain popularity. Spreadsheets include the capability to emulate some features of database managers, such as sorting ranges of cells that are defined as a worksheet "database."

One of the principal uses of Access 97 is the conversion of data in large worksheets to tables in a relational database structure. Some of the primary justifications for converting from the familiar worksheet model to an Access 97 database include the following:

- Access queries provide much greater flexibility in selecting and sorting data than is offered by the limited sort and selection criteria of spreadsheet applications.
- You can create Access forms to simulate common business forms, which is difficult or impossible to do with present-day Windows spreadsheet applications. Excel 97 forms (dialog sheets in Excel 95), for example, offer the choice of only a few simple unbound control objects. Data-entry validation is much easier in Access than in worksheets.
- Access offers many more options for printing formatted reports from your data than are available with worksheets.
- Access VBA allows you to write programs in a full-featured language and does not restrict you to using a set of predetermined worksheet functions. (Spreadsheet applications that use VBA or other BASIC-like macro languages, however, don't suffer from this restriction.)

Oracle Ex. 1009, p. 743
Oracle v. DeBellis
IPR2024-00221

- Properly designed Access relational databases minimize the duplication of information, speed data entry, and reduce disk file-storage requirements for large aggregations of data.

In many situations, however, changing a worksheet to a database is impractical. You may need to be able to view, import, or edit data that is contained in a worksheet within an Access application. In this case, linking the spreadsheet as an OLE object is the best method. You also can attach worksheet files in Excel 5+ formats to Access 97 tables with the Excel ODBC driver that is supplied with Microsoft Office.

The limited utility of attached worksheet files is demonstrated in the first sections of this chapter by the conversion process necessary to restructure the data to relational form. The second major topic of this chapter, "Using Excel as an OLE Server," describes how to link or embed worksheet data as OLE objects in bound or unbound object frames.

Note

Microsoft Excel 97 is used in the following examples, but you can use any Windows spreadsheet application that has OLE server capability. The first series of examples that show you how to reorganize spreadsheet data into Access tables doesn't require a spreadsheet application. You only need a suitable file in Excel .xls or Lotus .wk? format to import.

Importing and Reorganizing Worksheet Data

Worksheets created by Excel and other spreadsheet applications often contain data that is suitable for processing with relational database techniques. The worksheet usually is organized for viewing the data quickly; however, this organization seldom is appropriate for data manipulation by an RDBMS.

STK_21_1 (described in the next section) is an example of a worksheet that is formatted for viewing data. Four rows of 22 columns each contain the equivalent of a database record—all available data for a single stock. This type of structure, where a group of individual rows constitute a database record, is common in worksheet design. This structure differs from the design of a proper worksheet database, in which all of the data for a single entity (in this case, a stock) is contained in a single row. The examples that follow in this section illustrate how to import a spreadsheet of multiple-row-per-record type and convert the data it contains to related Access tables. Make-table and update queries assist in the reorganization of the data into relational form.

Note

The example presented in the following sections is based on a specific worksheet format that contains New York Stock Exchange trading data for common stocks. The format of the data in the worksheet is similar to that used by online sources of financial information, many of whose sources

Oracle Ex. 1009, p. 744
Oracle v. DeBellis
IPR2024-00221

organize their transmitted data for automatic or manual importing into worksheets. You're likely to find other time-series statistical data to have been created as worksheets. The following sections are specific to a single data format, but the process of converting almost all column-row, time-series data to database format is similar to that presented throughout this chapter.

Obtaining the Example Worksheet

The file used to create many of the examples in this chapter, Stk_21.xls (Excel 97 format), contains the high price, low price, and trading volume of more than 300 individual stocks for a 21-day period beginning on April 10, 1992. This file was originally created as a Lotus 1-2-3 worksheet file (STK_21.WK1) by Ideas Unlimited.


On the Web

The Stk_21.xls file is included in the example files for this book, available from the Macmillan Publishing USA Information SuperLibrary Web site. Follow the links from this URL to the *Special Edition Using Access 97* page for downloading instructions:

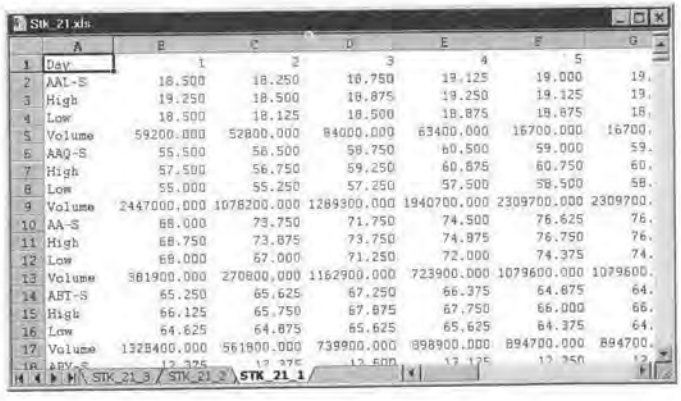
http://www.mcp.com/que/msoffice/


On the Web

The examples files are archived in ZIP format. You can download an evaluation copy of the latest 32-bit version of WinZip, which expands the ZIP files, from:

http://www.winzip.com/

Figure 21.1 shows the Stk_21.xls workbook opened in Excel 97. Stk_21.xls contains three worksheets. The worksheet named STK_21_1 contains the entire set of sample data—more than 300 stocks in more than 1,200 rows. The worksheet named STK_21_2 contains 201 rows and 50 stocks, while STK_21_3 contains 17 rows and 4 stocks. The two smaller worksheets, STK_21_2 and STK_21_3, demonstrate the effect of worksheet size on Access's performance in sections related to OLE later in this chapter. The first row of each worksheet is used to provide field names for the table. The first column of the first row contains the entry "Day," and the remaining columns of the first row are numbered sequentially from 1 to 21.


Stk_21.xls

	A	B	C	D	E	F	G
1	Day	1	2	3	4	5	
2	AAL-S	18.500	18.250	18.750	19.125	19.000	19.
3	High	19.250	18.500	18.875	19.250	19.125	19.
4	Low	18.500	18.125	18.500	18.875	18.875	18.
5	Volume	59200.000	52800.000	84000.000	63400.000	16700.000	16700.
6	AAQ-S	55.500	56.500	58.750	60.500	59.000	59.
7	High	57.500	56.750	59.250	60.875	60.750	60.
8	Low	55.000	55.250	57.250	57.500	58.500	58.
9	Volume	2447000.000	1078200.000	1289300.000	1940700.000	2309700.000	2309700.
10	AA-S	68.000	73.750	71.750	74.500	76.625	76.
11	High	68.750	73.875	73.750	74.875	76.750	76.
12	Low	68.000	67.000	71.250	72.000	74.375	74.
13	Volume	381900.000	270800.000	1162900.000	723900.000	1079600.000	1079600.
14	ABT-S	65.250	65.625	67.250	66.375	64.875	64.
15	High	66.125	65.750	67.875	67.750	66.000	66.
16	Low	64.625	64.875	65.625	65.625	64.375	64.
17	Volume	1328400.000	561800.000	739900.000	898900.000	894700.000	894700.
18	ABY-S	12.375	12.375	12.500	12.125	12.250	12.

STK_21_3 / STK_21_2 / STK_21_1

FIG. 21.1 The Stk_21.xls sample workbook opened in Excel 97.

Oracle Ex. 1009, p. 745
Oracle v. DeBellis
IPR2024-00221

The organization of these worksheets is especially well-suited to the first example in the next section, "Importing the Example Worksheet," which shows you how to reorganize a worksheet to create a properly designed table. Alternatively, you can use for the examples any Excel or .wk? file that contains multiple series of numbers.

Importing the Example Worksheet

The first step in the importing process is to create a new database in which to import the worksheet data. Follow these steps:

◀◀ See "Creating a New Database," p. 112
◀◀ See "Creating a Table by Importing an Excel Worksheet," p. 221

1. Launch Access if it is not open, select Blank Database in the opening dialog, and click OK. Alternatively, if Access is already open, choose File, New Database, and double-click the Blank Database icon. The New Database dialog appears.
2. In the New Database dialog, enter **Stocks.mdb** as the name of your new database, and then click Create.
3. Choose File, Get External Data, Import. The Import dialog opens (see Figure 21.2).



FIG. 21.2 Access 97's Import dialog.

4. Select Microsoft Excel (.xls) from the Files of Type drop-down list. (Support for importing all Excel file versions with a single file-type selection was new with Access 95.)
5. Select Stk_21.xls in the list, and click the Import button to display the first dialog of the Import Spreadsheet Wizard dialog (see Figure 21.3).

Oracle Ex. 1009, p. 746
Oracle v. DeBellis
IPR2024-00221

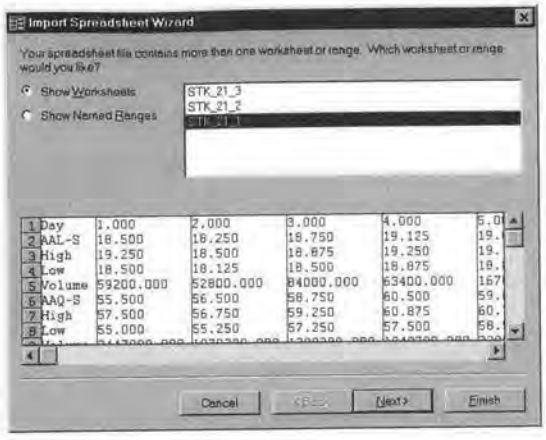


FIG. 21.3 Access 97's Import Spreadsheet Wizard's first dialog for importing worksheet files.

6. Select STK_21_1 from the worksheet name list. (Select one of the smaller sheets if you want to save disk space.) Click Next to display the second wizard dialog.

7. Mark the First Row Contains Column Headings check box (see Figure 21.4). Access displays a message informing you that the first row contains data that can't be used for valid Access field names and that the wizard will assign valid field names for you. (Access field names must begin with an alphabetic character, so the column headings 1–21 can't be converted directly to field names.) Click OK to dismiss the information dialog, and then click Next to display the third wizard dialog (selecting a destination for the imported data).

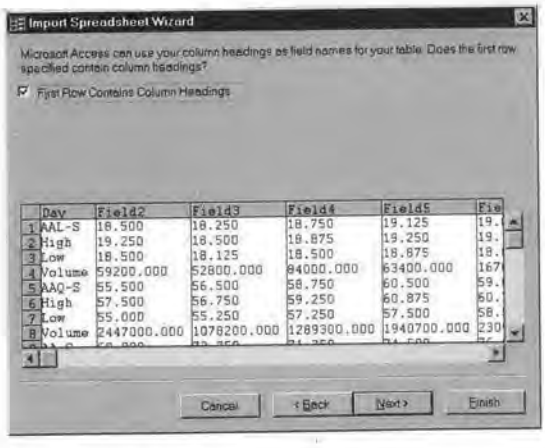


FIG. 21.4 The Import Spreadsheet Wizard's second dialog.

Oracle Ex. 1009, p. 747
Oracle v. DeBellis
IPR2024-00221

8. Select the In a New Table option, and click Next to display the fourth wizard dialog (field options).
9. You don't need to specify field data types in this case, so click Next to display the fifth wizard dialog (primary key options). You don't need a primary key at this point, so click the No Primary Key option button, and click Next to display the final Wizard dialog.
10. Because you specified a new table for the imported data, Access now asks you for a name for the new table. The default name for the new table, STK_21_1, is correct, so click Finish to import the spreadsheet.
11. When the Wizard's message box announces that the import process has completed, click OK.



12. With the STK_21_1 table, click the Open button to display the table in Table view.
13. Select all 22 fields by clicking the Day column header button, holding down the mouse button, and dragging the mouse to the right. Choose Format, Column Width to display the Column Width dialog (see Figure 21.5).
14. Click the Best Fit button to reduce the column widths to suit the entries in the table.



FIG. 21.5 Setting the column width of the STK_21_1 table.

Access has interpreted the field type of the Excel 97 numeric data as double-precision, which is not required for any of the values in the table. Single-precision numbers can accommodate all of the numeric values in the table. Using single-precision, rather than double-precision, values saves about 33 percent of the disk space.

To change the Field Size property of the numeric fields of the STK_21_1 table, follow these steps:

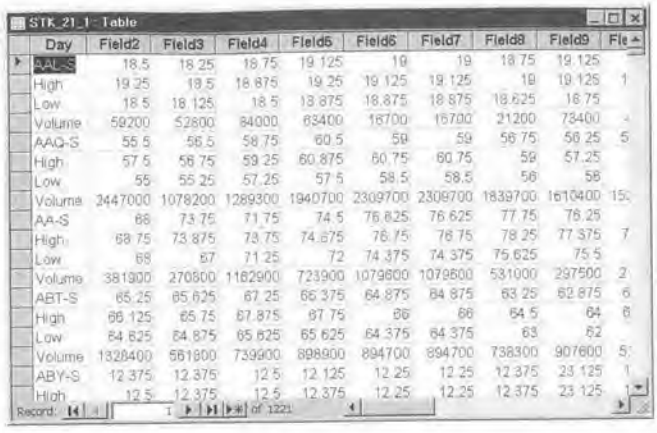


1. Click the Design View button.
2. Select the first Number field—the field named Field2. Change the Field Size value from Double to Single.
3. Repeat step 2 for the remaining 20 Number fields (Field3–Field23).



4. Click the Datasheet button to return to Table view. Click Yes when asked whether you want to save your changes and when notified that some data may be lost. Access converts the field sizes in the STK_21_1 table from Double to Single at this point, and no data is lost. Your table appears as shown in Figure 21.6.

Oracle Ex. 1009, p. 748
Oracle v. DeBellis
IPR2024-00221



STK_21_1 : Table

Day	Field2	Field3	Field4	Field5	Field6	Field7	Field8	Field9	Fie
AAL-S	18.5	18.25	18.75	19.125	19	19	18.75	19.125	
High	19.25	18.5	18.875	19.25	19.125	19.125	19	19.125	1
Low	18.5	18.125	18.5	18.875	18.875	18.875	18.625	18.75	
Volume	59200	52800	84000	83400	16700	16700	21200	73400	4
AAQ-S	55.5	56.5	58.75	60.5	59	59	56.75	56.25	5
High	57.5	58.75	59.25	60.875	60.75	60.75	59	57.25	
Low	55	55.25	57.25	57.5	58.5	58.5	56	56	
Volume	3447000	1078200	1289300	1940700	2309700	2309700	1839700	1610400	15
AA-S	68	73.75	71.75	74.5	76.625	76.625	77.75	78.25	
High	68.75	73.875	73.75	74.875	76.75	76.75	78.25	77.375	7
Low	68	67	71.25	72	74.375	74.375	75.625	75.5	
Volume	381900	270500	1162900	723900	1079600	1079600	531000	297500	2
ABT-S	65.25	65.625	67.25	66.375	64.875	64.875	63.25	62.875	6
High	66.125	65.75	67.875	67.75	66	66	64.5	64	6
Low	64.625	64.875	65.625	65.625	64.375	64.375	63	62	
Volume	1328400	561600	739900	898800	894700	894700	738300	907600	5
ABY-S	12.375	12.375	12.5	12.125	12.25	12.25	12.375	23.125	1
High	12.5	12.375	12.5	12.375	12.25	12.25	12.375	23.125	1

Record: 1 of 1221

FIG. 21.6 The revised version of the STK_21_1 table.

Developing a Conversion Strategy

The second step in the conversion process is to define the tables to be created as a result of normalizing the data in the imported spreadsheet (*normalizing* the data means converting it to a form that follows the rules of relational database design). Very few worksheets can be converted to a single table that meets relational database standards. Designing the tables to contain the data and establishing the relationships of these tables to one another are the most important elements of the conversion process. Following are the two objectives of your design strategy:

▶▶ See "Normalizing Data to the Relational Model," p. 802



- *A design optimized for data display, entry, and editing.* This objective is primary if you plan to use Access for data entry. In the stock prices example, data entry and editing aren't a consideration because the data is supplied in worksheet format by Ideas Unlimited and other purveyors of stock price data.
- *A design allowing you to extract imported data with the fewest steps.* If you obtain periodic updates to your stock price data in worksheet format, for example, ease of importing the data is the principal consideration.

The initial design usually is a compromise between these two objectives. The stock price example does not involve a compromise because data entry and editing are unnecessary.

When planning the initial design of the tables to contain the stock price and volume data, consider these points:

Oracle Ex. 1009, p. 749
Oracle v. DeBellis
IPR2024-00221



▶▶ See "Fifth Normal Form and Combined Entities," p. 807

- *The data for each element of the group—Close, High, Low, and Volume—is incorporated in an individual table.* A table with the 84 fields required to hold all data for a stock is unwieldy at best. Reconstructing a worksheet from a table with such a design results in a cumbersome worksheet. One of the principles of relational database design is that you should be able to reconstruct your original database from the relational data.
- *The key field of each table, Symbol, is the ticker symbol of the stock.* (The ticker symbol is the abbreviation for the name of the stock assigned by the New York Stock Exchange.) This key field allows the tables to be linked in a one-to-one relationship based on the unique values of the ticker symbols. One-to-one relationships are uncommon in relational databases, but in this example a one-to-one relationship is quite useful.
- *Queries organize the data in the tables as required for Access forms and reports.* Because each field in the query is prefaced by the table name, the names of the tables should be short to save keystrokes.
- *When multiple tables are combined in a query, the records in each table must be identified by type.* The second field, Type, is a single-letter abbreviation of the type of data: C(lose), H(igh), L(ow), and V(olume).

Now you need to develop the tactics to create the required tables in accordance with your strategy. The conversion plan involves these elements:

- The records to be included in the individual tables are extracted from the STK_21_1 table by make-table queries with criteria based on the values in the Day field.
- The Symbol field from the Close table must be added to the High, Low, and Volume tables.
- In the query used to create the final table, adding the Symbol field to a table requires a unique key to link the Close table and the tables without symbol values. An AutoNumber field added to the Close, High, Low, and Volume tables can serve as a temporary key field.
- Creating the new tables is a two-step process. First, the data is extracted to a temporary set of tables (Hi, Lo, and Vol). The second step combines these three tables with the Symbol field of another temporary table—Close—to create the final High, Low, and Volume tables. You need temporary tables in this case because make-table queries should never alter tables on which they are based.

Oracle Ex. 1009, p. 750
Oracle v. DeBellis
IPR2024-00221

- Because the Close table lacks a type identifier field, a Type field must be added to this table. The remaining tables have a type identifier word that can be replaced by the corresponding code letter.

Now that you have a conversion strategy and have decided the tactics for carrying it out, you are ready to test the merits of both, as described in the next section.

Extracting Data

Make-table queries are designed specifically for creating new tables from data in existing tables that meet a specified set of criteria. For STK_21_1, you use make-table queries to create one final table (Close) and three temporary tables (Hi, Low, and Vol). To create these four tables with make-table queries, follow these steps:

◀◀ See "Creating Action Queries to Append Records to a Table," p. 378



1. With the STK_21_1 table open, click the arrow of the New Object button of the toolbar and select New Query from the list to create a new query based on STK_21_1.



2. With Design View selected in the New Query dialog's list, click OK. Access creates a new query and automatically adds the STK_21_1 table to the query.
3. Click the Day field, and drag the field symbol to the first column of the query.
4. Click the asterisk (*, all fields), and drag the field symbol to the second column of the query.
5. Click the Show check box in the Day field to clear the box. (Day is included in the fields in the second column of the query. You cannot include two fields of the same name in a make-table query.)
6. In the Criteria row of the Day field, enter **High** to include only records for the high price of the stock in the temporary Hi table. Your query appears, as shown in Figure 21.7.
7. Choose Query, Make Table Query to display the Make Table dialog.
8. Enter **Hi** in the Table Name text box and then click OK (see Figure 21.8). The default values for the remaining elements in the Make Table dialog are satisfactory.
9. Click the Run button of the toolbar to create the temporary Hi table. Access displays a message box that indicates the number of records to be added to the new table (see Figure 21.9). Click Yes to complete the make-table query.



Oracle Ex. 1009, p. 751
Oracle v. DeBellis
IPR2024-00221

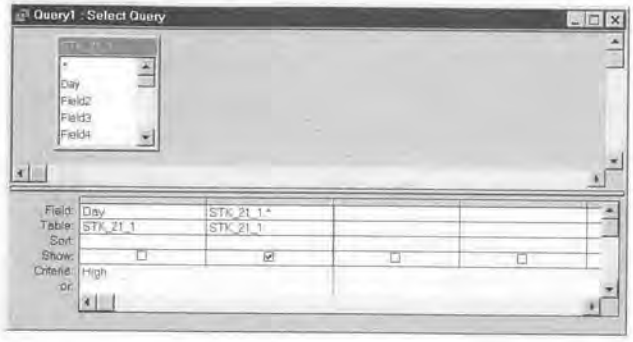


FIG. 21.7 Creating the temporary Hi table with a make-table query.



FIG. 21.8 Entering the temporary table name in the Make Table dialog.

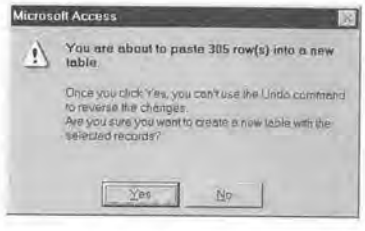


FIG. 21.9 The message box that confirms the number of rows added to the new table.

Tables

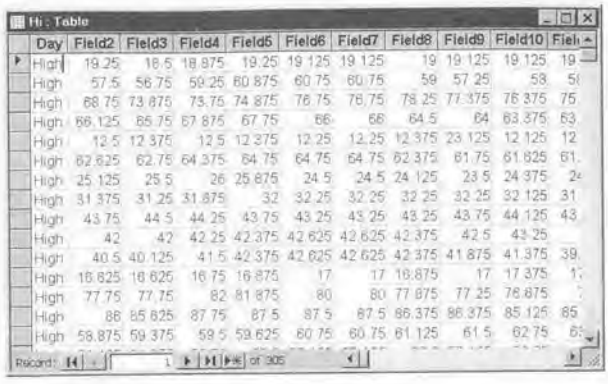
10. Open the Database window, click the Tables tab, and double-click the Hi entry in the list to view the table (see Figure 21.10).

11. Change the Criteria value of your make-table query to **Low** and repeat steps 6–10, substituting **Lo** for the table name in steps 6 and 10.

12. Change the Criteria value to **Volume** and repeat steps 6–10, using **Vol** for the table name.

13. The closing prices are in the row with the ticker symbol. Each of the symbols has a hyphen followed by a character that identifies the type of security; -S represents a common stock. Change the Criteria value to **Like** "*-*" to select these rows and repeat steps 6 through 10, using **Close** for the table name. The expression *-* selects all records containing a hyphen. You don't need to create a temporary table in this case because the Close table includes the ticker symbols, as illustrated by Figure 21.11.

Oracle Ex. 1009, p. 752
Oracle v. DeBellis
IPR2024-00221

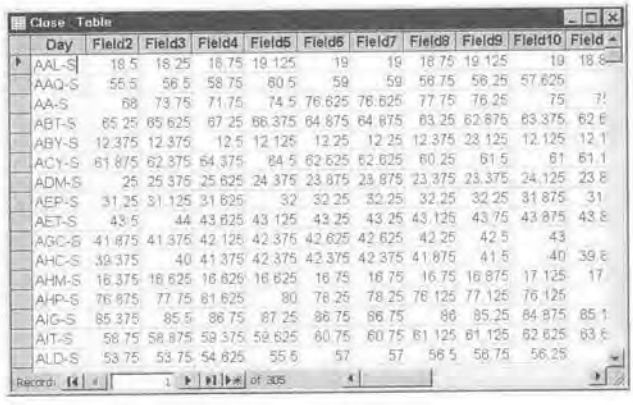


Hi : Table

Day	Field2	Field3	Field4	Field5	Field6	Field7	Field8	Field9	Field10	Fiel
High	19.25	18.5	18.875	19.25	19.125	19.125	19	19.125	19.125	19
High	57.5	56.75	59.25	60.875	60.75	60.75	59	57.25	58	5
High	68.75	73.875	73.75	74.875	76.75	76.75	78.25	77.375	76.375	75
High	66.125	65.75	67.875	67.75	66	66	64.5	64	63.375	63
High	12.5	12.375	12.5	12.375	12.25	12.25	12.375	23.125	12.125	12
High	62.625	62.75	64.375	64.75	64.75	64.75	62.375	61.75	61.625	61
High	25.125	25.5	26	25.875	24.5	24.5	24.125	23.5	24.375	2
High	31.375	31.25	31.875	32	32.25	32.25	32.25	32.25	32.125	31
High	43.75	44.5	44.25	43.75	43.25	43.25	43.25	43.75	44.125	43
High	42	42	42.25	42.375	42.625	42.625	42.375	42.5	43.25	
High	40.5	40.125	41.5	42.375	42.625	42.625	42.375	41.875	41.375	39
High	16.625	16.625	16.75	16.875	17	17	16.875	17	17.375	1
High	77.75	77.75	82	81.875	80	80	77.875	77.25	76.875	
High	86	85.625	87.75	87.5	87.5	87.5	86.375	86.375	85.125	85
High	58.875	59.375	59.5	59.625	60.75	60.75	61.125	61.5	62.75	6

Record: 1 of 305

FIG. 21.10 The temporary Hi table created by the make-table query.



Close : Table

Day	Field2	Field3	Field4	Field5	Field6	Field7	Field8	Field9	Field10	Field
AAL-S	18.5	18.25	18.75	19.125	19	19	18.75	19.125	19	18.8
AAQ-S	56.5	56.5	58.75	60.5	59	59	58.75	58.25	57.625	
AA-S	68	73.75	71.75	74.5	76.625	76.625	77.75	76.25	75	7
ABT-S	65.25	65.625	67.25	66.375	64.875	64.875	63.25	62.875	63.375	62.6
ABY-S	12.375	12.375	12.5	12.125	12.25	12.25	12.375	23.125	12.125	12.1
ACY-S	61.875	62.375	64.375	64.5	62.625	62.625	60.25	61.5	61	61.1
ADM-S	25	25.375	25.625	24.375	23.875	23.875	23.375	23.375	24.125	23.8
AEP-S	31.25	31.125	31.625	32	32.25	32.25	32.25	32.25	31.875	31
AET-S	43.5	44	43.625	43.125	43.25	43.25	43.125	43.75	43.875	43.8
AGC-S	41.875	41.375	42.125	42.375	42.625	42.625	42.25	42.5	43	
AHC-S	39.375	40	41.375	42.375	42.375	42.375	41.875	41.5	40	39.6
AHM-S	16.375	16.625	16.625	16.625	16.75	16.75	16.75	16.875	17.125	17
AHP-S	76.875	77.75	81.625	80	78.25	78.25	76.125	77.125	76.125	
AIG-S	85.375	85.5	86.75	87.25	86.75	86.75	86	85.25	84.875	85.1
AIT-S	58.75	58.875	59.375	59.625	60.75	60.75	61.125	61.125	62.625	63.6
ALD-S	53.75	53.75	54.625	55.5	57	57	56.5	56.75	56.25	

Record: 1 of 305

FIG. 21.11 The Close table with the ticker symbol of the stocks.

Modifying the Table Structure

Each table you created in the preceding section requires additional fields so that you can design an append query to create a properly structured table. An AutoNumber field is used as a temporary key field for each of the four tables. The Close table needs a Type field added, and the Day field of each temporary table needs the name changed to Type. Follow these steps to make the changes:

Tip

AutoNumber fields are the fastest way of adding a primary-key field to a table that does not contain unique values on which to base a primary key.

1. Click the Tables tab in the Database window, select the Hi table, and click the Design button.



Oracle Ex. 1009, p. 753
Oracle v. DeBellis
IPR2024-00221

2. Change the name of the first field from Day to **Type** and set the Field Size property to **1**, as shown in Figure 21.12. This action truncates the Type field value, High, to the required single-letter code H when you save your design changes.

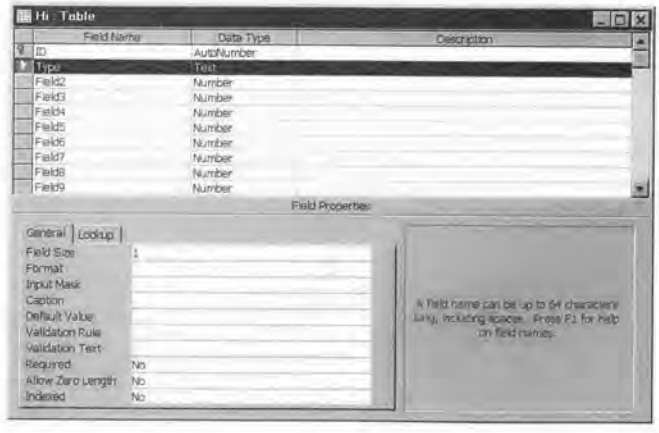


FIG. 21.12 Modifications to the design of the temporary Hi table.



3. Click the field selector button of the Type field, and press Insert to add a field at the beginning of the table. Type **ID** in the Field Name column, and select AutoNumber in the Data Type column. Click the Primary Key button of the toolbar to make the ID field the primary-key field.



4. Click the Table View button on the toolbar. Save the changes to the table. A message box advises you that some data may be lost as a result of truncating the Type field length (see Figure 21.13). Click Yes to approve the change. The table appears as shown in Figure 21.14.



FIG. 21.13 The warning that some data may be lost as a result of truncating the Type field.



5. Repeat steps 1 through 4 for the Lo and Vol tables.

6. Open the Close table in Design mode. Change the Day field name to **Symbol**, and change its Field Size property to **10** to accommodate longer ticker symbols.



7. Click the Selection button for the Symbol field, and press Insert to add an **ID** AutoNumber field. Click the Primary Key button of the toolbar to create the primary-key index.

Oracle Ex. 1009, p. 754
Oracle v. DeBellis
IPR2024-00221

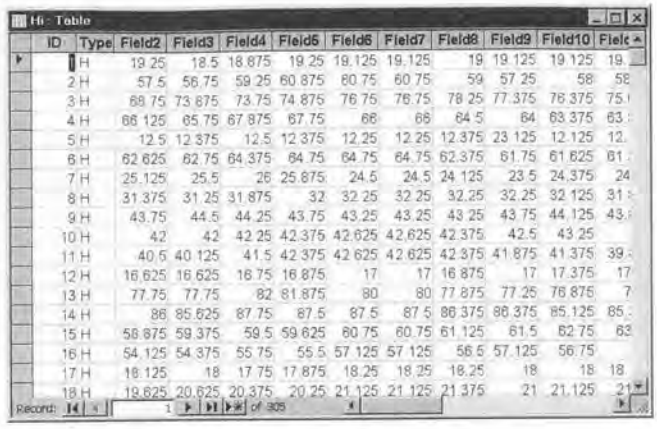


FIG. 21.14 The temporary Hi table with the ID key field added.

8. Select the Field2 field (the first of the Number fields), and then press Insert to add a new field. Enter **Type** as the Field Name of the new field (added fields default to the Text data type), and set its Field Size property to **1**. Figure 21.15 shows the design of the Close table.

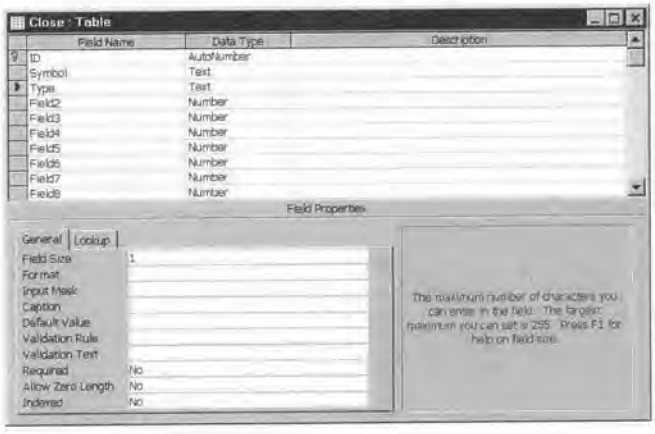


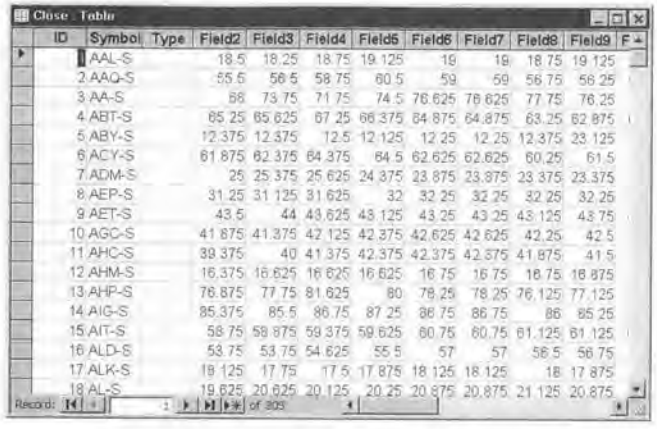
FIG. 21.15 Modifications made to the design of the Close table.

9. Click the Table View button of the toolbar, save your changes, and accept the truncated field. Figure 21.16 shows the resulting Close table.



The extraction of data from STK_21_1 to the required tables is a relatively simple process because the labels used to identify the data are consistent throughout the worksheet. You may need to write a worksheet macro that creates consistent labels for rows to be included in a specific table if the labels don't exist or aren't consistent in the original version of the worksheet.

Oracle Ex. 1009, p. 755
Oracle v. DeBellis
IPR2024-00221



Close : Table

ID	Symbol	Type	Field2	Field3	Field4	Field5	Field6	Field7	Field8	Field9	F
1	AAL-S		18.5	18.25	18.75	19.125	19	19	18.75	19.125	
2	AAQ-S		55.5	56.5	58.75	60.5	59	59	56.75	56.25	
3	AA-S		68	73.75	71.75	74.5	76.625	76.625	77.75	76.25	
4	ABT-S		65.25	65.625	67.25	66.375	64.875	64.875	63.25	62.875	
5	ABY-S		12.375	12.375	12.5	12.125	12.25	12.25	12.375	23.125	
6	ACY-S		61.875	62.375	64.375	64.5	62.625	62.625	60.25	61.5	
7	ADM-S		25	25.375	25.625	24.375	23.875	23.875	23.375	23.375	
8	AEP-S		31.25	31.125	31.625	32	32.25	32.25	32.25	32.25	
9	AET-S		43.5	44	43.625	43.125	43.25	43.25	43.125	43.75	
10	AGC-S		41.875	41.375	42.125	42.375	42.625	42.625	42.25	42.5	
11	AHC-S		39.375	40	41.375	42.375	42.375	42.375	41.875	41.5	
12	AHM-S		16.375	16.625	16.625	16.625	16.75	16.75	16.75	16.875	
13	AHP-S		76.875	77.75	81.625	80	78.25	78.25	76.125	77.125	
14	AIG-S		85.375	85.5	86.75	87.25	86.75	86.75	86	85.25	
15	AIT-S		58.75	58.875	59.375	59.625	60.75	60.75	61.125	61.125	
16	ALD-S		53.75	53.75	54.625	55.5	57	57	56.5	56.75	
17	ALK-S		19.125	17.75	17.5	17.875	18.125	18.125	18	17.875	
18	AL-S		19.625	20.625	20.125	20.25	20.875	20.875	21.125	20.875	

Record: 1 of 305

FIG. 21.16 The Close table with the ID and Type fields added.

Adding the Type Value to the Close Table

When you need to replace data in a field containing text values, the Replace command on the Edit menu is usually a faster process than creating an UPDATE query. Because the Replace What text box in the Replace in Field dialog doesn't accept a Null value, however, you cannot use the Replace command on the Edit menu to add the C code (for "Close") to the Type field of the Close table. Instead, you must use an update query to change the Type field value. Follow these steps:

◀◀ See "Updating Values of Multiple Records in a Table," p. 383

New Object: Query

1. Make the Close table active. Click the arrow of the New Object button of the toolbar, and choose New Query from the menu list.
2. Select Design View in the New Query dialog's list, and then click OK to create a new query with the Close table added.
3. Choose _Query, Update Query to change the query type from select to update.
4. Select the Type field and drag the field symbol to the first column of the query.
5. Type **C** in the Update To row. This action changes the Type field value in all records from the default Null to the single letter C. Figure 21.17 shows the Update Query design at this point.
6. Click the Run button on the toolbar to update the Type field of the Close table. When the message box shown in Figure 21.18 appears, click Yes.

Oracle Ex. 1009, p. 756
Oracle v. DeBellis
IPR2024-00221

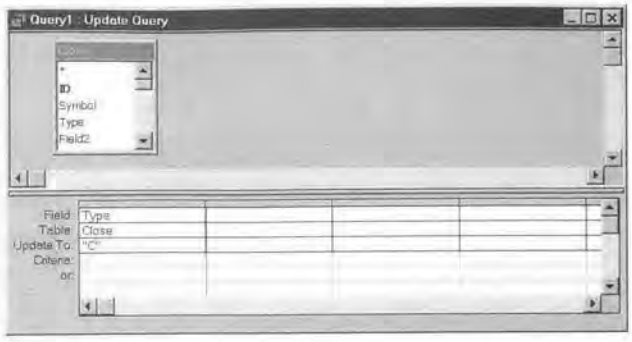


FIG. 21.17 The Update Query used to add C to the Type field of each record in the Close table.

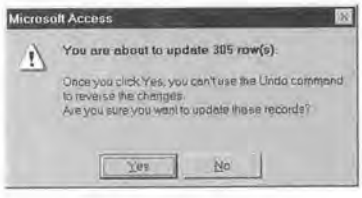


FIG. 21.18 The message box that indicates the number of records to be updated.

Creating the Final Tables

Now, you need to combine the Symbol field of the Close table with the data in the Hi, Lo, and Vol tables to create the final High, Low, and Volume tables. In this example, you use the same query each time, changing the temporary table name to create the three final tables.

Tip

If you are creating an application that uses Access VBA code to automate the conversion process, save each query with a unique name so that the query can be run by a VBA procedure.

To create your final High, Low, and Volume tables with make-table queries, follow these steps:

1. With the Close table active, click the arrow of the New Object button of the toolbar and select Design View. Then, click the OK button of the New Query dialog to create a new query based on the Close table. New Object: Query
2. Choose Query, Make Table Query to open the Make Table dialog. Enter **High** in the Table Name text box as the name of the final table to create, and click OK.
3. Select the Symbol field of the Close table, and drag it to the first column of the make-table query.

Oracle Ex. 1009, p. 757
Oracle v. DeBellis
IPR2024-00221



4. Click the Show Table button of the toolbar or choose Query, Show Table to open the Show Table dialog. Select Hi from the Tables list and click Add. Then choose the Close button. Access automatically creates the join between the ID fields of the Close and Hi tables for you.

Caution

This step is very important. If Access fails to establish this relationship and you don't add the relationship, you create a *Cartesian product* instead of the result you want. A Cartesian product is all possible combinations of all field values contained in the two tables. In this case, the Cartesian product contains about 95,000 (305 * 305) rows.

5. Click the asterisk (*, all fields) of the Hi table, and drag the field symbol to the second column of the query. The make-table query appears as shown in Figure 21.19.

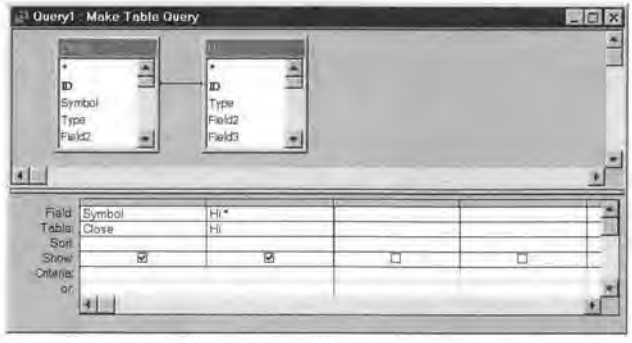


FIG. 21.19 The make-table query that creates the final High table.



6. Click the Run button of the toolbar to create the final High table. Click OK in the message box that appears to confirm creating the new table.
7. In the query, select the Hi table field list, and press Delete to remove the field list from the query. Repeat steps 4 through 6 for the Lo table, substituting **Low** as the name for the new final table in the Make Table dialog.
8. Repeat step 7 for the Vol table, entering **Volume** as the name of the final table in the Make Table dialog.
9. Close the query, and don't save the changes.

You must remove the temporary key field, ID, from the Close table so that you can make Symbol the key field. Follow these steps:



1. Open the Database window, click the Tables tab, and select the Close table. Click the Design button.



Oracle Ex. 1009, p. 758
Oracle v. DeBellis
IPR2024-00221

2. Click the field selection button of the ID field, and then press Delete. When Access displays message boxes asking you to confirm deletion of the field and key field, click OK.
3. Select the Symbol field; then click the Primary Key button of the toolbar to make the Symbol field the primary-key field. Figure 21.20 shows the Design view of the Close table.



4. Click the Table View button on the toolbar to display the contents of the table. Click OK when asked to confirm your changes.



5. Select all columns of the table by dragging the mouse pointer over the column's field name buttons.
6. Choose Format, Column Width to open the Column Width dialog, and then click Best Fit to change the column sizes and close the Column Width dialog. Figure 21.21 shows the resulting Close table.

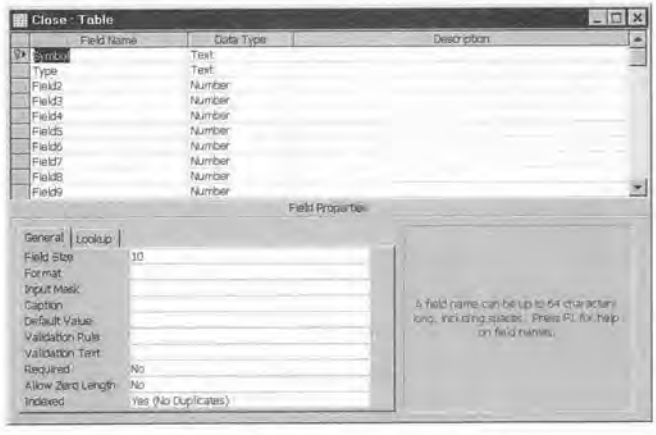


FIG. 21.20 The Close table with Symbol as the primary-key field.

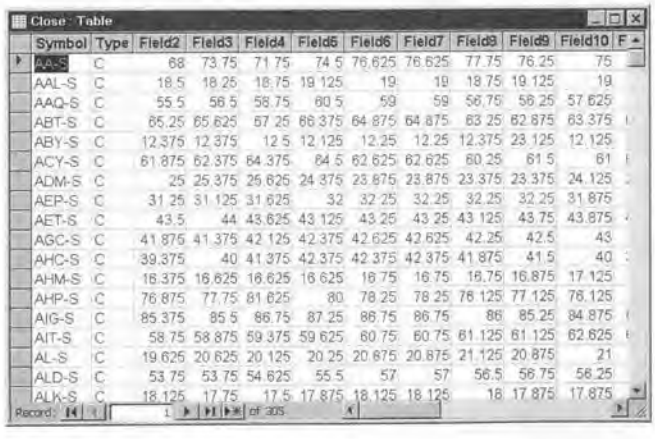


FIG. 21.21 The final version of the Close table.

Oracle Ex. 1009, p. 759
Oracle v. DeBellis
IPR2024-00221

7. Choose File, Save to save the column width change.

At this point, you can delete the Hi, Lo, and Vol tables. Deleting temporary tables conserves disk space, but only after you compact the database.

Verifying the Tables by Re-Creating the Worksheet

An important step when creating tables from external data sources is to verify that the tables contain the correct information. In most cases, the best method of testing is to use the tables to re-create the data in its original format—or as close to the original format as possible. This strategy allows you to make a direct comparison of the source data and the data contained in the tables.

◀◀ See "Copying and Pasting Tables," p. 159

To create a replica of the original STK_21_1 worksheet, follow these steps:

Tables

1. Make the Database window active, click the Tables tab, and select the Close table.
2. Press Ctrl+C to copy the table to the Clipboard, and then press Ctrl+V to create a copy of the table. The Paste Table As dialog appears.
3. Type **tblStockPrices** as the name of the new table in the Paste Table As dialog (see Figure 21.22). Make sure that the default Structure and Data options button is selected, and then click OK. Access creates a new tblStockPrices table.

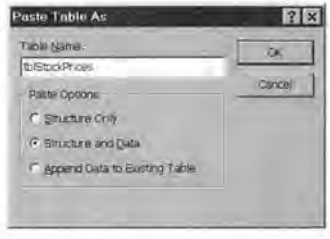


FIG. 21.22 Entering the table name in the Paste Table As dialog.

Design

4. Select the tblStockPrices table in the Tables list of the Database window, and then click the Design button.

5. Click the Indexes button of the toolbar to display the Indexes dialog.

6. Select both the Symbol and Type fields by holding down the Shift key and clicking the two field selection buttons. Then click the Primary Key button of the toolbar.
7. Select the Symbol field and click the Indexed text box in the Field Properties section of the Design window. Choose Yes (Duplicates OK) from the Indexed drop-down list. If you choose No (No Duplicates), you cannot append records. Indexing the Symbol field can speed up queries based on a specific symbol or set of symbols. Figure 21.23 shows the design for the tblStockPrices table.

Oracle Ex. 1009, p. 760
Oracle v. DeBellis
IPR2024-00221

8. Click the Table View button of the toolbar, and click OK when asked whether you want to save your changes. Review the data in the table, and then close the tblStockPrices window.

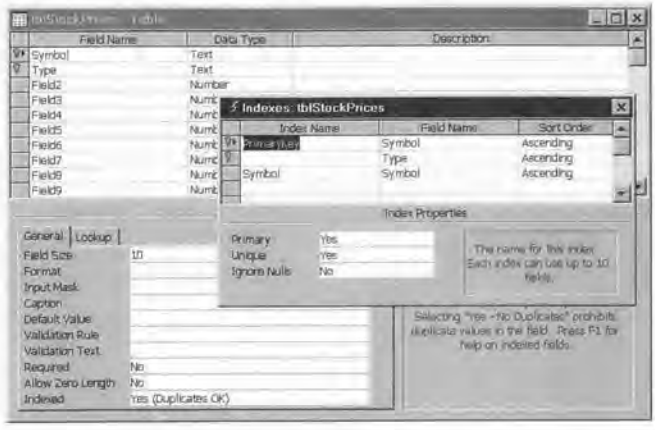


FIG. 21.23 Modifying the key field and indexing properties of the tblStockPrices table.

In preparation for appending the data in the other tables to the tblStockPrices table, you need to remove the primary key (ID) field. In prior versions of Access, you could append data from tables without an identical set of fields. Access 97 issues an `INSERT INTO statement contains unknown field name 'FieldName'` message if the field names don't match in the source and destination tables. Open the High, Low, and Volume tables in Design view, and delete the ID field of each of these three tables.

> **Note**
>
> An alternative to deleting the ID fields is to add all fields, except the ID field, of the High, Low, and Volume fields to the queries you create in the following series of steps. If you do delete the ID fields from the tables, you should make backup copies of the High, Low, and Volume tables, because you'll use these tables again in the next section of this chapter, "Using the Tables You Created."

To add the data in the High, Low, and Volume tables to the Stock Data table, follow these steps:

1. Select the High table in the Database window; then click the New Object button of the toolbar, and select New Query from the menu list to display the New Query dialog.



2. With Design View selected, click OK to open a new query based on the High table.
3. Choose Query, Append to display the Append dialog. Open the Table Name drop-down list and select tblStockPrices (see Figure 21.24). The tblStockPrices table is the table to which you want to append the records. Click OK.

Oracle Ex. 1009, p. 761
Oracle v. DeBellis
IPR2024-00221



FIG. 21.24 Entering the name of the table to append records to in the Append dialog.

4. Select the asterisk (*) field of the High table field list and drag the field list symbol to the first column of the query. Figure 21.25 shows the append query design.

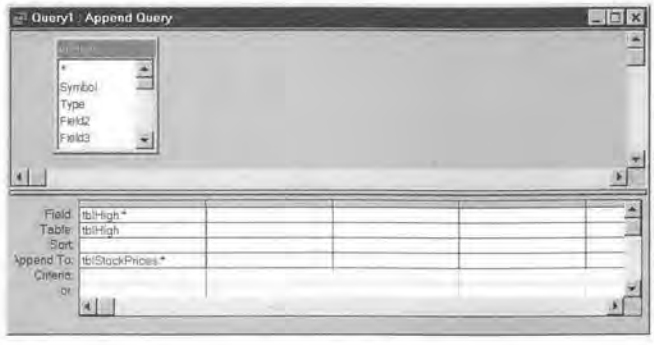


FIG. 21.25 The append query design for adding records to tblStockPrices from the High table.



5. Click the Run button on the toolbar to append the High records to Stock Data. The message box shown in Figure 21.26 indicates the number of records to be appended. Click Yes.

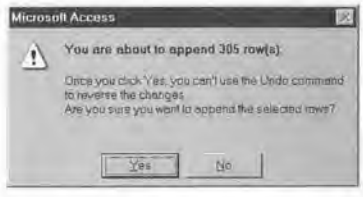


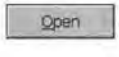
FIG. 21.26 The message box confirming the append operation.



6. Delete the High table's field list from the query, and then click the Show Table button of the toolbar. Repeat steps 4 and 5 to append the Low and Volume table data to tblStockPrices.







7. After you have appended the data from the High, Low, and Volume tables to tblStockPrices, open the tblStockPrices table from the Database window. The table appears as shown in Figure 21.27. The data for Close, High, Low, and trading Volume appears in the same sequence as in the worksheet because the table is indexed on the combination of the Symbol and the Type fields, the primary key. Access creates a no-duplicates index on the key field(s).

Oracle Ex. 1009, p. 762
Oracle v. DeBellis
IPR2024-00221

8. Drag the tblStockPrices table to the lower right of your display. Open the STK_21_1 table; then click the exposed surface of the tblStockPrices window to compare it with the original version (see Figure 21.28).

9. Indexing tblStockPrices has changed the order of some of the entries. Confirm that the data for a few stocks in both tables are the same.

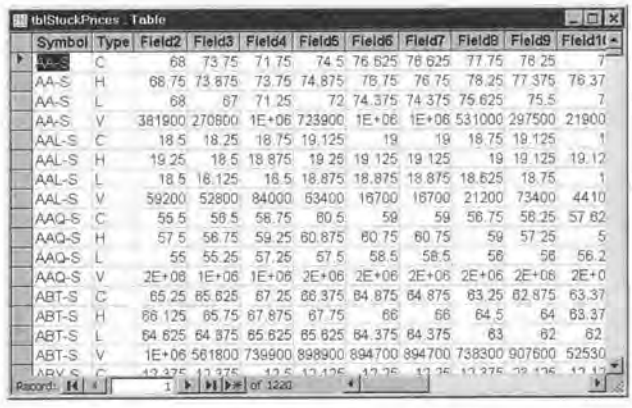


FIG. 21.27 The final version of the tblStockPrices table with all records appended.

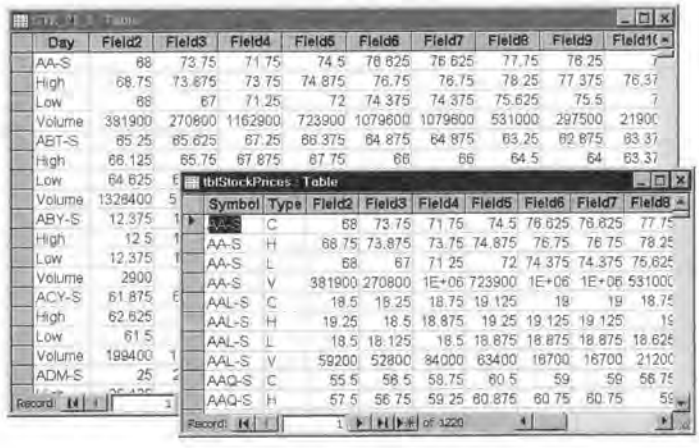


FIG. 21.28 Comparing the tblStockPrices table with the original version.

You can add stock price data for later dates by consecutively numbering the fields in successive Excel tables. As an example, the next 21 days of prices and trading volumes would use field names 23–44 for the numeric values. You could name the tables you create Close2, High2, Low2, and Volume2, and then add these tables to your queries to extend the range of dates to be included. Alternatively, you could create tables that have the high, low, close, and volume data for a given stock on a specific date. Making this type of transformation, however, requires that you write Access VBA code to restructure the tables.

Oracle Ex. 1009, p. 763
Oracle v. DeBellis
IPR2024-00221

> **Note**
>
> The structure of the tblStockPrices table does not conform to the rules of relational databases, although its structure is an improvement over the original Excel worksheet. Using individual fields for dates constitutes *repeating groups*, a violation of first normal form for relational tables. A fully normalized table would consist of a single record for the high, low, close, and volume data for a single stock on a single date. The composite primary key of such a table would be the stock ticker symbol and the date. Creating a fully normalized table from STK_21.xls requires the use of Automation and a substantial amount of Access VBA code.

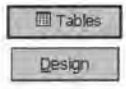
Using the Tables You Created

In this section, queries combine the data in the tables to create forms and reports that display the data in tabular or graph form. The relationships between the tables should be established automatically for the queries you create. In addition, you need to maintain referential integrity; that is, you shouldn't be able to delete a closing price for a stock for which you have high, low, and volume data. This obligation requires that you include relationships as properties of the tables.



◀◀ See "Establishing Relationships Between Tables," p. 146
◀◀ See "Enforcing Referential Integrity," p. 157

To establish the relationships between the tables, follow these steps:



1. Open the High table in Design view, select the Symbol field, and click the Primary Key button of the toolbar to make the Symbol field the primary key field. Repeat the process for the Low and Volume tables.

   Access 95 and 97 require a primary key to determine whether the relationship between tables is one-to-one or one-to-many. (Access 2.0 lets you specify the type of relationship.)



2. Make the Database window active, and then click the Relationships button of the toolbar or choose Tools, Relationships. The empty Relationships window opens with the Show Table dialog active.
3. Select the High table in the Tables list box, and click the Add button.
4. Repeat step 2 for the Low, Close, and Volume tables, and then click the Close button. Your Relationships window appears as shown in Figure 21.29.
5. Drag the Symbol field from the High table field list to the Symbol field of the Low table field list to create a join. Access opens the Relationships dialog.
6. Mark the Enforce Referential Integrity check box to enable cascading updates and deletions. Access automatically determines the relationship between each table as one-to-one by testing the joined fields.

Oracle Ex. 1009, p. 764
Oracle v. DeBellis
IPR2024-00221

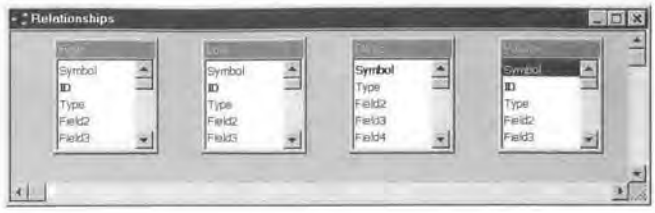


FIG. 21.29 The Relationships window with the four tables added.

7. Mark the Cascade Update Related Fields check box. This action lets you change the Symbol value for a stock if its NYSE ticker symbol should change, an unlikely (but conceivable) event. If you change one Symbol value, the Symbol values for all of the related tables change in unison.

8. Mark the Cascade Delete Related Records check box. This action lets you remove all of the records for a stock that is delisted from the NYSE, a more likely event. Your Relationships dialog appears as shown in Figure 21.30.

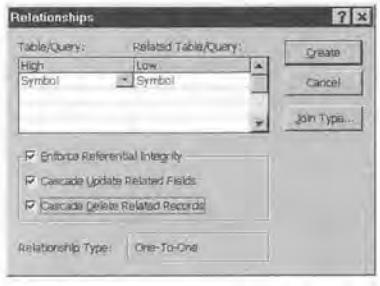


FIG. 21.30 Setting referential integrity enforcement rules in the Relationships dialog.

9. Click the Create button to create the join and close the Relationships dialog.

10. Repeat steps 4 through 8 for the join between the Low and Close tables and the join between the Close and Volume tables. The Relationships window now appears as shown in Figure 21.31.



FIG. 21.31 The Relationships window with one-to-one relationships between each table.

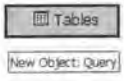
11. Close the Relationships window and save your changes.

Oracle Ex. 1009, p. 765
Oracle v. DeBellis
IPR2024-00221

Note

Always establish default relationships between the tables of your database so that these relationships are added automatically to the queries you create. If you don't establish default relationships and then Access doesn't create or you forget to add relationships to your query, you may obtain the Cartesian product (described earlier in this section) instead of the result you want. The Cartesian product of large tables can be large and take several minutes to create. Pressing Ctrl+Break may not halt the process, and Windows may exhaust its resources in creating the Cartesian product, creating an out-of-memory error.

To create a test query that includes all of the data in your tables, follow these steps:



1. Select the High table in the Database window, click the arrow of the New Object button of the toolbar, and select New Query from the button menu to display the New Query dialog.
2. With Design View selected in the list, click OK to create a query with the High table added.



3. Click the Show Table button of the toolbar to add the Low, Close, and Volume tables. The joins between the tables are established automatically (by the relationships you established in the preceding series of steps) as you add each table, as shown by the lines connecting the Symbols fields in Figure 21.32. Click the Close button to close the Show Table dialog.

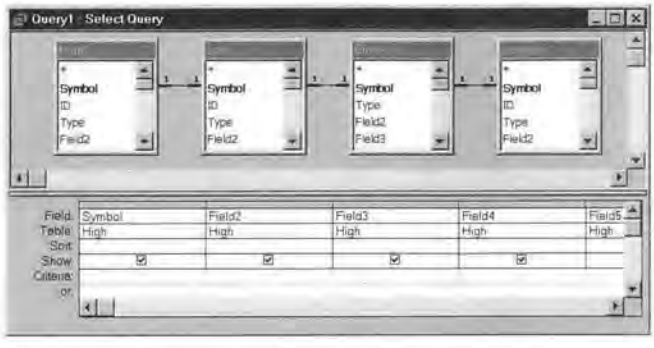


FIG. 21.32 Part of the query design for the 21-day stock prices query.

4. Double-click the header of the High table's field list to select all of the fields, and drag the multiple field column to the first column of the query.
5. Select and delete the Type and ID columns. Then use the horizontal scroll bar slider to display the first empty column after the 22 columns devoted to the High fields.
6. Repeat steps 4 and 5 for the Low, Close, and Volume tables, but also delete the Symbol field for these three tables.

Oracle Ex. 1009, p. 766
Oracle v. DeBellis
IPR2024-00221

7. Click the Run button of the toolbar to display the query. You can drag the columns of the query to reorder the columns in a logical sequence, as shown in Figure 21.33.



8. Choose View, SQL to display the SQL statement for the query (see Figure 21.34).

Query1 : Select Query

Symbol	High.Field2	Low.Field2	Close.Field2	Volume.Field2	High.Field3	L
AAL-S	19.25	18.5	18.5	59200	18.5	
AAQ-S	57.5	55	55.5	2447000	56.75	
AA-S	68.75	68	68	581900	73.875	
ABT-S	66.125	64.625	65.25	1329400	65.75	
ABY-S	12.5	12.375	12.375	2900	12.375	
ACY-S	62.625	61.5	61.875	189400	62.75	
ADM-S	25.125	24.5	25	1819500	25.5	
AEP-S	31.375	31	31.25	214300	31.25	
AET-S	43.75	42.5	43.5	160700	44.5	
AGC-S	42	41.75	41.875	66400	42	
AHC-S	40.5	39.25	39.375	213300	40.125	
AHM-S	16.825	16.125	16.375	266000	16.625	
AHP-S	77.75	76.75	76.875	561800	77.75	

Record: 1 of 305

FIG. 21.33 The query result set with the columns reordered.

Query1 : Select Query

```
SELECT High.Symbol, High.Field2, High.Field3, High.Field4, High.Field5, High.Field6,
High.Field7, High.Field8, High.Field9, High.Field10, High.Field11, High.Field12,
High.Field13, High.Field14, High.Field15, High.Field16, High.Field17, High.Field18,
High.Field19, High.Field20, High.Field21, High.Field22, Low.Field2, Low.Field3, Low.Field4,
Low.Field5, Low.Field6, Low.Field7, Low.Field8, Low.Field9, Low.Field10, Low.Field11,
Low.Field12, Low.Field13, Low.Field14, Low.Field15, Low.Field16, Low.Field17,
Low.Field18, Low.Field19, Low.Field20, Low.Field21, Low.Field22, Close.Field2,
Close.Field3, Close.Field4, Close.Field5, Close.Field6, Close.Field7, Close.Field8,
Close.Field9, Close.Field10, Close.Field11, Close.Field12, Close.Field13, Close.Field14,
Close.Field15, Close.Field16, Close.Field17, Close.Field18, Close.Field19, Close.Field20,
Close.Field21, Close.Field22, Volume.Field2, Volume.Field3, Volume.Field4, Volume.Field5,
Volume.Field6, Volume.Field7, Volume.Field8, Volume.Field9, Volume.Field10,
Volume.Field11, Volume.Field12, Volume.Field13, Volume.Field14, Volume.Field15,
Volume.Field16, Volume.Field17, Volume.Field18, Volume.Field19, Volume.Field20,
Volume.Field21, Volume.Field22
FROM ((High INNER JOIN Low ON High.Symbol = Low.Symbol) INNER JOIN Close ON
(High.Symbol = Close.Symbol) AND (Low.Symbol = Close.Symbol)) INNER JOIN Volume
ON (High.Symbol = Volume.Symbol) AND (Close.Symbol = Volume.Symbol);
```

FIG. 21.34 The SQL statement for the queries of Figures 21.32 and 21.33.

You can use queries of the preceding type to display or edit any data in the four tables by using a form, or these queries can be used to print part or all of the stock data with an Access report. You might design other queries that display only stocks which meet a specific criterion, such as minimum trading volume or a range of stock closing prices. The ability to rearrange data in almost any desired row and column sequence quickly, using simple queries, demonstrates that Access query datasheets are at least as flexible as worksheets.

Oracle Ex. 1009, p. 767
Oracle v. DeBellis
IPR2024-00221

Exporting Stock Data as a Worksheet

When you are collecting data on the same set of common stocks, you can simplify the conversion process by modifying the design of the Excel worksheet to correspond with the design of the tables. You can avoid making a large number of changes to the worksheet in Excel by exporting the Access tblStockPrices table back to Excel in Excel 97 format.

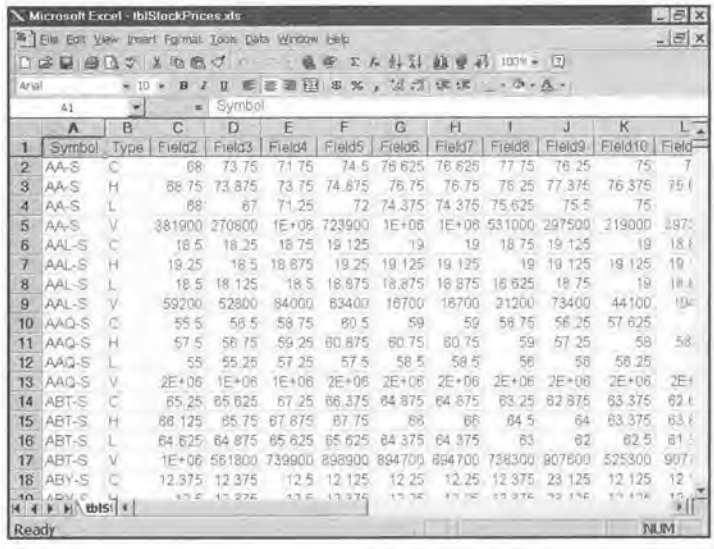
To export the Stock Data table to an .XLS workbook file, follow these steps:



1. Activate the Database window, and select the tblStockPrices table.



2. Click the arrow of the Office Links button of the toolbar, and choose Analyze It with Excel to export the table data and launch Excel with the tblStockPrices.xls worksheet active (see Figure 21.35).



	A	B	C	D	E	F	G	H	I	J	K	L
1	Symbol	Type	Field2	Field3	Field4	Field5	Field6	Field7	Field8	Field9	Field10	Field
2	AA-S	C	68	73.75	71.75	74.5	76.625	76.625	77.75	76.25	75	7
3	AA-S	H	68.75	73.875	73.75	74.875	76.75	76.75	76.25	77.375	76.375	75
4	AA-S	L	68	67	71.25	72	74.375	74.375	75.625	75.5	75	
5	AA-S	V	381900	270800	1E+06	723900	1E+06	1E+06	531000	297500	219000	297
6	AAL-S	C	18.5	18.25	18.75	19.125	19	19	18.75	19.125	19	18
7	AAL-S	H	19.25	18.5	18.875	19.25	19.125	19.125	19	19.125	19.125	19
8	AAL-S	L	18.5	18.125	18.5	18.875	18.875	18.875	18.625	18.75	19	18
9	AAL-S	V	59200	52800	84000	83400	16700	16700	21200	73400	44100	19
10	AAQ-S	C	55.5	56.5	58.75	60.5	59	59	58.75	56.25	57.625	
11	AAQ-S	H	57.5	56.75	59.25	60.875	60.75	60.75	59	57.25	58	58
12	AAQ-S	L	55	55.25	57.25	57.5	58.5	58.5	56	58	56.25	
13	AAQ-S	V	2E+06	1E+06	1E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E+06	2E
14	ABT-S	C	65.25	65.625	67.25	66.375	64.875	64.875	63.25	62.875	63.375	62
15	ABT-S	H	66.125	65.75	67.875	67.75	66	66	64.5	64	63.375	63
16	ABT-S	L	64.625	64.875	65.625	65.625	64.375	64.375	63	62	62.5	61
17	ABT-S	V	1E+06	561800	739900	898900	894700	894700	738300	907800	525300	907
18	ABY-S	C	12.375	12.375	12.5	12.125	12.25	12.25	12.375	23.125	12.125	12

FIG. 21.35 The stock price data appears in the order of the table's primary-key fields.

3. Choose File, Save to save the sorted worksheet and then choose File, Exit to close Excel.

You can use the new worksheet as a template for the entry or import of stock price and trading volume data for other ranges of dates. Using the new format eliminates the necessity of creating temporary tables during the conversion process.

Using Excel as an OLE Server

This section describes methods of creating links between data cells in OLE Object fields and worksheets created with Excel 97. You can duplicate some examples with Excel 3.0 and 4.0, but version 5 and later versions—with their OLE 2+ compatibility—have several added features that simplify the process. The step-by-step examples in this section are based on the assumption that you are familiar with the use of Excel 5+.

Oracle Ex. 1009, p. 768
Oracle v. DeBellis
IPR2024-00221

You can embed or link an Excel worksheet or graph as a bound or unbound OLE object. You can copy data from the OLE worksheet object to the Clipboard and then paste the data into a bound or unbound text box. The following sections provide examples of these techniques.

Embedding an Excel Worksheet in a Form

You can embed an entire Excel worksheet in an unbound object frame using the simple process that follows. In this case, the presentation of an Excel worksheet (or what you can display of the worksheet in a bound object frame) is the entire content of the worksheet. Large embedded worksheets occupy considerable disk space and may require a substantial period of time to display their presentation; you may want to use a small worksheet (a file size of 5K or less, such as a workbook file created from the STK_21_3 worksheet) if you are short on disk space.

To embed an Excel worksheet in an unbound object frame of a new form, follow these steps:

1. Open the Stocks.mdb database, if necessary, and create a new blank form. Initially, size the form to about 3.33 inches high by 6 inches wide.
2. Click the Toolbox button of the toolbar and select the unbound object frame tool. Create a frame with the left corner at about .5 inch from the form's left edge and about 0.3 inch from the top. (The dimensions of these margins are important; see the troubleshooting note at the end of this section.) The Insert Object dialog appears.
3. Click the Create from File option button, then click the Browse button to open the Browse Dialog and locate the folder containing the tblStockPrices.xls file. Select tblStockPrices.xls (or the name of any other Excel 5+ workbook file you want to embed) in the file list, and then click Open to close the Browse dialog. Then click OK to close the Insert Object dialog (see Figure 21.36).

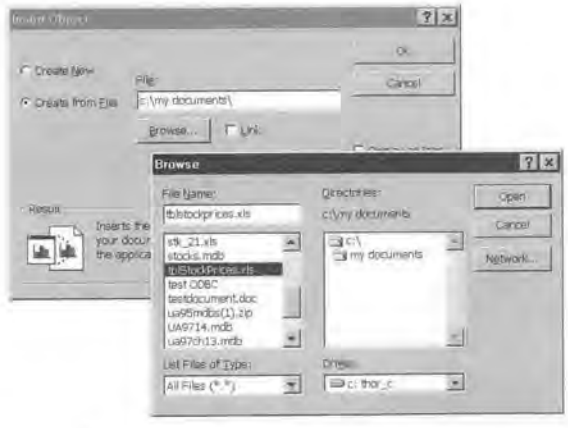


FIG. 21.36 Inserting an unbound object from an Excel 97 file.

Oracle Ex. 1009, p. 769
Oracle v. DeBellis
IPR2024-00221

Troubleshooting

A message box appears, indicating that the Excel worksheet is corrupted, that Excel is not properly registered, or that not enough available memory exists to open Excel.

Close other open applications and try again. If the procedure continues to display error messages, exit Windows and start over. (An application has failed to release its global memory blocks when closing.) If you continue to receive error messages that refer to the Registry, you need to verify that the Registry entries for Excel are correct. See Chapter 19, "Using 32-Bit OLE Components," for detailed instructions on the Registry entries required for local servers.

4. Select the unbound object frame and drag the right border to about 5.8 inches. Drag the bottom border to about 3.1 inches.

5. Click the Properties button of the toolbar. Click the Data tab and set the value of the Enabled property to Yes and the Locked property to No. If you don't enable and unlock the object frame, you can't activate the object (see Figure 21.37).



FIG. 21.37 Setting the Enabled and Locked properties of the object to allow in-place activation.



6. Click the Form View button of the toolbar to display the presentation of your worksheet (see Figure 21.38). If only part of a column displays, return to Design view and adjust the width of your object frame so that a full column is displayed with a 0.25-inch border at the right. You also may need to adjust the height of the form to provide a 0.25-inch border at the bottom. (The borders inside the object frame are necessary; see the following troubleshooting note.)

7. Double-click the object frame to activate the worksheet in place. If you have set the left and top positions of your object frame and provided the proper internal borders, your activated worksheet appears as shown in Figure 21.39. Excel 97's Edit and View menus replace Access's Edit and View menus, and Excel's Insert, Format, Tools, and Data menus are added to Access's menu. The toolbars that normally appear when you open Excel are added to Access 97's toolbars. Activated mode is indicated by a hashed border around the object frame.

Oracle Ex. 1009, p. 770
Oracle v. DeBellis
IPR2024-00221

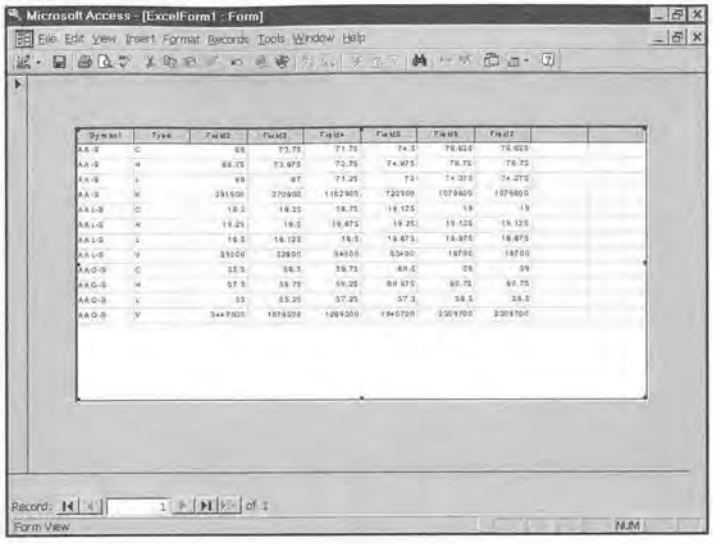


FIG. 21.38 The presentation of a worksheet in an unbound object frame.

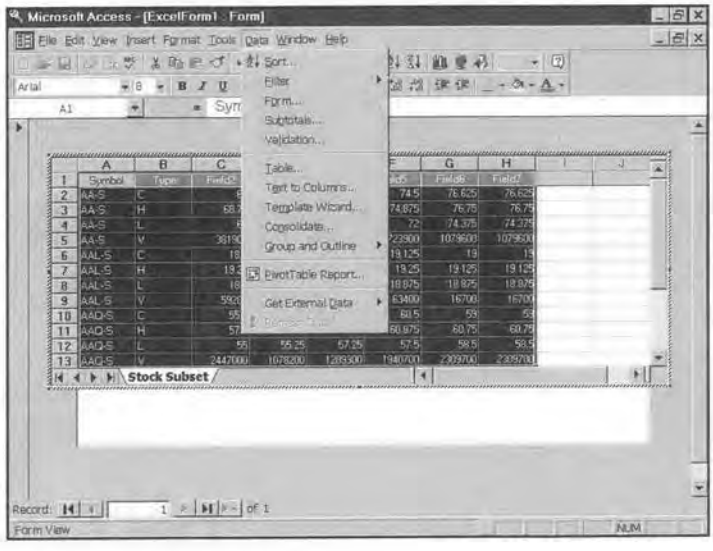


FIG. 21.39 The worksheet activated in place.

You can perform any operation that is possible in Excel 97 when the object is activated, except operations that require use of Excel's File menu—such as saving the workbook to a file or printing the worksheet.

8. Click outside the bound object frame to deactivate the object, and return to presentation mode so that Access's menubar is active.

Oracle Ex. 1009, p. 771
Oracle v. DeBellis
IPR2024-00221

9. Choose Edit, Worksheet Object, Open to open Excel 97's window with the embedded data as the source of Excel's current workbook (see Figure 21.40). When you open Excel, you can print the worksheet or save the embedded workbook to a file.

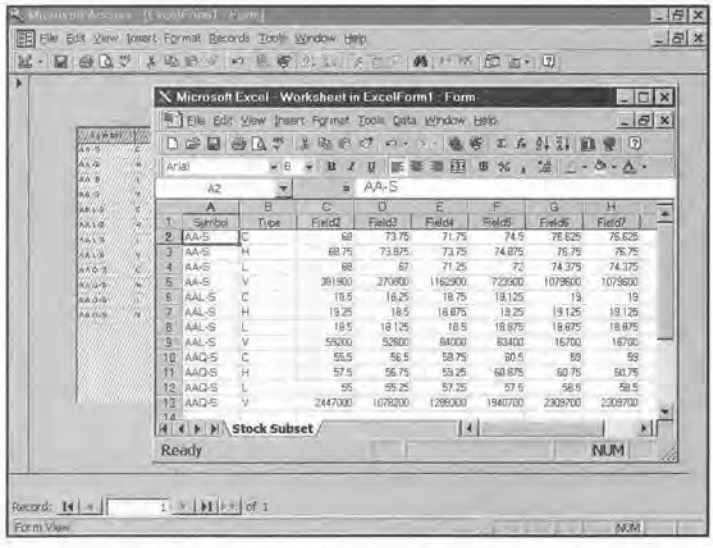


FIG. 21.40 Opening Excel 97 in its own window (from Presentation mode).

10. Choose File, Exit to close the instance of Excel and return to presentation mode.

11. Close your form and save the changes.

Troubleshooting

Excel's column selection buttons, row selection buttons, worksheet tabs, horizontal scroll bar, or the vertical scroll bar don't appear when the object is activated.

Two sets of critical factors determine the visibility of *adornments*, as the preceding objects are called, when you activate an object in place. The top position of the object frame must provide room for the column selector buttons in the form area (0.25 inch), and the left position of the frame must provide space for the row selector buttons (0.5 inch). The visibility of the horizontal and vertical scroll bars is determined by the internal margin of the object, the space between the edge of the worksheet presentation, and the bottom and right edge of the object frame (0.25 inch each).

If, after setting these values, all of the adornments shown in Figure 21.39 do not appear, position the mouse pointer on the upper-left corner of the activation border and drag the corner diagonally downward to reduce the size of the activation frame. You may need to make repeated adjustments to the size of the object frame and the activation frame to ensure that all adornments appear.

Finally, you may need to set the Size Mode property of the unbound object frame to Zoom. If the embedded spreadsheet is very large, the default Clip property may cause the adornments at the right and bottom edges of the activated workbook object to be clipped off.

Oracle Ex. 1009, p. 772
Oracle v. DeBellis
IPR2024-00221

Extracting Values from an OLE Object

You can copy individual numeric or text values from a linked or embedded Excel OLE object and place the values in a text box. To add the close, high, and low values of the AAL-S stock to a multiline text box added to the form you created in the preceding section, follow these steps:

1. In Design view, reduce the size of your unbound object frame containing the Excel worksheet object, and move the frame down to make room for a multiline text box at the top of the form.
2. Add a text box to the form, and set the value of the Scrollbars property to Vertical.
3. In Form view, double-click the unbound object frame to activate the workbook object.

> **Note**
>
> You must copy and paste the selected values while the form is in Form view; otherwise, the pasted text becomes the text box's Control Source property, resulting in the display of the `#Name?` error when you display the form in Form view.

4. Select the cell or range of cells you want to import to a text control object. In this case, select A2:H5 to return seven days of price data for the AA-S stock.
5. Choose Edit, Copy, or press Ctrl+C to copy the A2:H5 range to the Clipboard.
6. Select the text box, which deactivates the object frame, and then choose Edit, Paste Special to display the Paste Special dialog. When you paste data items from an embedded object into a text box, you only have the option of pasting them as text (see Figure 21.41). Click OK to paste the text into the text box.



FIG. 21.41 Pasting a selection from an embedded Excel worksheet as text.

7. The pasted data items appear as shown in Figure 21.42. The vertical bars between the values in the text box represent the tab characters that Excel uses to separate data columns in a row.

Oracle Ex. 1009, p. 773
Oracle v. DeBellis
IPR2024-00221

If you create one or more bound text boxes with a numeric data type, you can paste a number from a selected single cell to each text box and then use the values to update the fields of the table to which the text box is bound. A more efficient method, however, is to use Automation to update values in tables with data from another application.

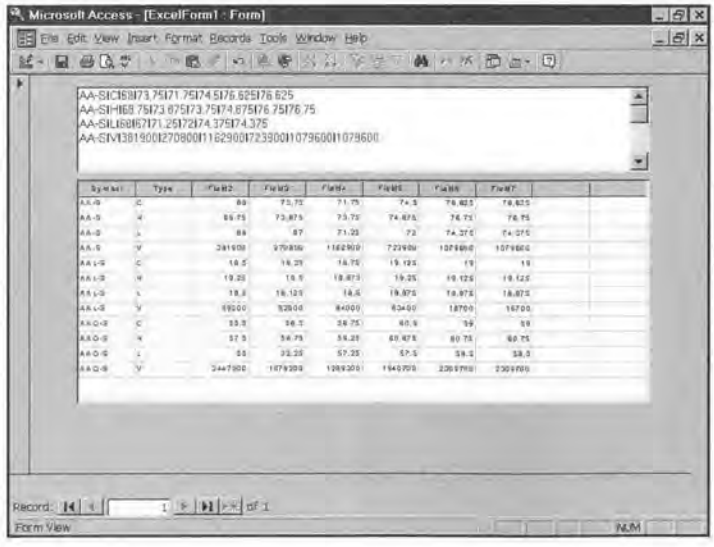


FIG. 21.42 A selection from an embedded Excel worksheet pasted into an unbound text box.

Linking to a Range of Cells in an Excel Worksheet

Embedding Excel objects is useful if you want to use Automation to transfer data from a `Recordset` object to embedded worksheet cells. In most cases, however, creating an OLE link to all or a range of cells in a worksheet is a more common practice. Linking allows you to display or edit the most recent version of the worksheet's data from its source file. Any changes you make in Access are reflected when you close Excel if you save the changes.



▶▶ See "Manipulating an Excel 97 Workbook Object," p. 1102

The conventional process of linking a file in Excel is similar to that for using OLE 1.0 to link graphics files; in-place activation is not available with linked objects. To create a link with a range of cells in an Excel file, perform the following steps:

1. Open a new, blank Access form.
2. Launch Excel independently of Access.

Oracle Ex. 1009, p. 774
Oracle v. DeBellis
IPR2024-00221

3. Choose File, Open, and in Excel's Open dialog select the file you want to link. This example uses tblStockPrices.xls.
4. Select the cells of the worksheet to be included in your Access table, and then copy the selected cells to the Clipboard with Ctrl+C. Cell range A1:E17 of the tblStockPrices.xls worksheet is used in this example.
5. In Access, choose Edit, Paste Special to display the Paste Special dialog. Click the Paste Link option button. Your only choices for a linked object are the object or text contained in the selected data items. The data source, tblStockPrices!R1C1:R17C5, appears in the Source label (see Figure 21.43).



FIG. 21.43 The Paste Special dialog for a linked range of cells in an Excel worksheet.

6. Click OK to create the unbound object frame containing the presentation of the selected cells. You don't need to provide for margins in the design because in-place activation is not available with linked objects.
7. Open the Properties window and set the value of the Enabled property of the object frame to Yes and the Locked property of the frame to No.
8. Click the Form View button of the toolbar. The presentation of your linked object appears, as shown in Figure 21.44.
9. Double-click the presentation of the worksheet to launch Excel in its own window with the linked cells selected (see Figure 21.45). As noted earlier in this section, in-place activation does not apply to linked objects.
10. In Excel, choose File, Exit to return the focus to Access. If you have made changes to the data, you can elect to save the changes at this point.

Oracle Ex. 1009, p. 775
Oracle v. DeBellis
IPR2024-00221

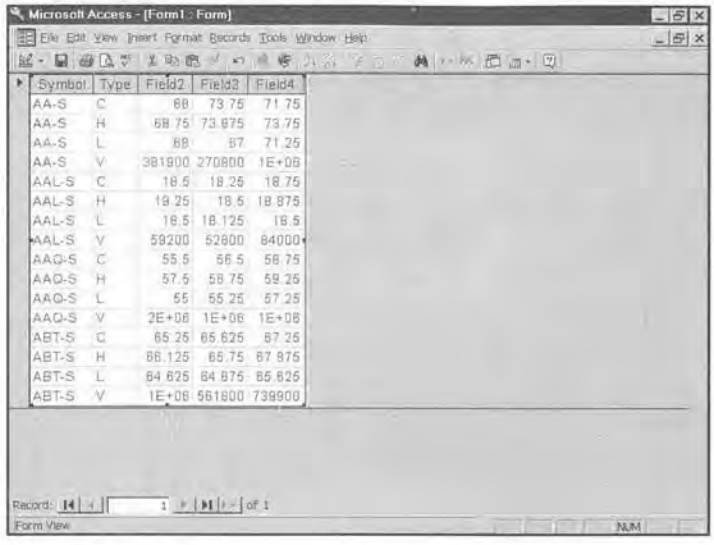


FIG. 21.44 The presentation of a linked range of worksheet cells.

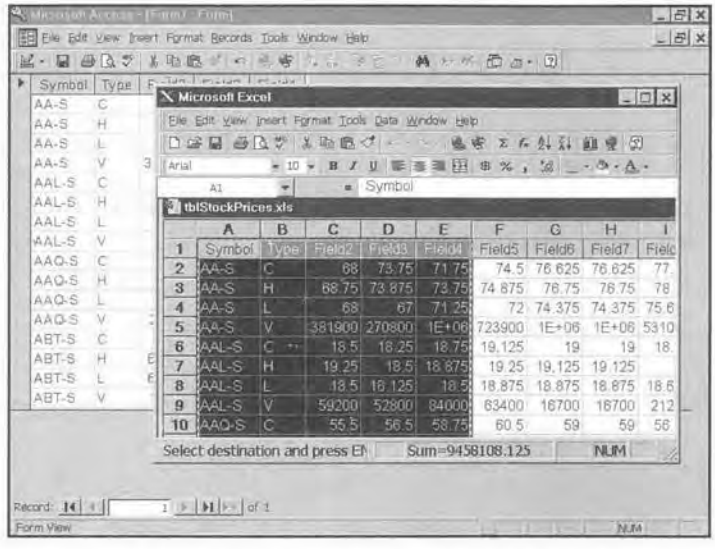


FIG. 21.45 The instance of Excel launched by double-clicking the linked cells' presentation.

Oracle Ex. 1009, p. 776
Oracle v. DeBellis
IPR2024-00221

Chapter 22

Using Access with Microsoft Word and Mail Merge

Members of the Microsoft Office 97 software suite are specifically designed to make constructing cooperative applications easy. *Cooperative applications* use two or more Windows productivity applications to perform a specified task. One of the principle uses for database applications is creating mailing lists for use in conjunction with form letters. Thus, Access 97—a member of the Professional Edition of Microsoft Office 97—includes a Mail Merge Wizard that not only automates the process of creating Word 97 Merge data files but also helps you create new form letters.

You also can use the reverse process and create form letters using Word 97's new mail merge process. Creating form letters from Word 97 accommodates users who do not have retail Access 97 on their computers. Word 97 uses 32-bit Microsoft Query (Msqry32.exe) and the new 32-bit Open Database Connectivity (ODBC) application programming interface (API) version 2.5 to connect to Access 97 and earlier .mdb files, as well as to a variety of other desktop database types.

As with Excel worksheets, you can embed or link Word documents in bound or unbound object frames and add a complete word-processing system to your Access application. If you embed the Word document in the object frame, you can take advantage of OLE 2.1's in-place activation to make the operating environment of Access almost identical to that of Word. Word's menu supplements the Access menu, and Word's toolbars appear as docked or floating toolbars on your display. Word's document editing window, in Page view, appears within the confines of your object frame.

Using the Access Mail Merge Wizard

Access 97's Mail Merge Wizard can help you create a new main merge document or employ an existing main merge document from which to create form letters. The Mail Merge Wizard uses a table or a query as the data source for the merge data file. The sections that follow describe the following two methods of creating a form letter:

Oracle Ex. 1009, p. 777
Oracle v. DeBellis
IPR2024-00221

- Using the Mail Merge Wizard to create a new main merge document whose merge data source is an Access table
- Using an existing main merge document with a merge data source from an Access select query

Creating and Previewing a New Form Letter

When you first try a new wizard, it's customary to create a new object rather than use the wizard to modify an existing object, such as a main merge document. The following steps use the Mail Merge Wizard to create a new main merge document from records in the Customers table of Northwind.mdb.

> **Note**
>
> Using the Mail Merge Wizard consumes an extraordinary amount of Windows resources because Word and two instances of Access run simultaneously. If you have less than 16M of RAM, close all running applications except Word and Access before starting the following process.



1. Open Northwind.mdb, if necessary, and select the Customers table in the Database window.



2. Click the arrow of the Office Links button of the toolbar, and select Merge-It from the drop-down menu to launch the Microsoft Mail Merge Wizard. The wizard's first and only dialog appears as shown in Figure 22.1.

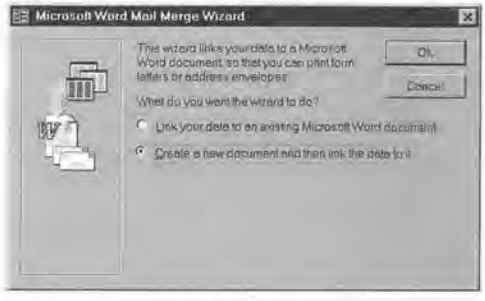


FIG. 22.1 The sole dialog of the Microsoft Word Mail Merge Wizard.

3. Select the Create a New Document and Then Link the Data To It option to create a new main merge document using fields from the Customers table.
4. Click OK to launch Word 97 if it is not running, and open a new mail merge main document. The Mail Merge Wizard uses dynamic data exchange (DDE) to communicate with Word and opens a new instance of Access 97.



5. Click the Insert Merge Field button to verify the fields from the Customers table in the drop-down list, as shown in Figure 22.2.

Oracle Ex. 1009, p. 778
Oracle v. DeBellis
IPR2024-00221

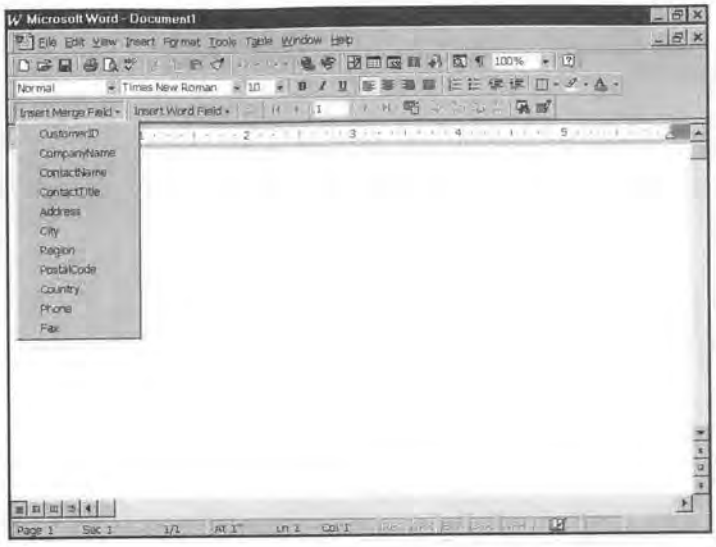


FIG. 22.2 Displaying the available merge fields in Word 97's Mail Merge window.

6. With the caret at the top of the document, choose Insert, Date and Time to display the Date and Time dialog; choose any date format you want, and then click OK to add a date field to the main document.

Tip

In Word 97, click the Show/Hide button on Word's toolbar to display end-of-paragraph marks, space characters, tab characters, and other document symbols that are usually hidden. All of the figures of Word 97 in this chapter were taken with the Show/Hide button in its down position.



7. Add a blank line, click the Insert Merge Field button to display the drop-down list, and insert the fields from the Customers table to create the address section of the main document (see Figure 22.3).



Note

Spaces and other punctuation in merge data field names are not permitted by Word. The Mail Merge Wizard substitutes underscores (_) for spaces and other illegal characters in Access field names, when present.

8. Click the View Merged Data button of the Mail Merge toolbar to preview the appearance of your form letter.



Oracle Ex. 1009, p. 779
Oracle v. DeBellis
IPR2024-00221

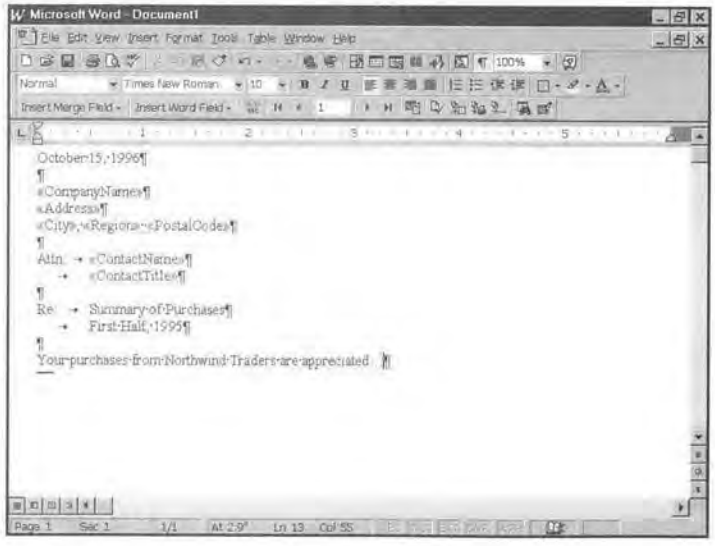


FIG. 22.3 Adding the merge fields to the main merge document.

Tip

Click the Find Record button of the toolbar, type **USA** in the Find What text box, and select Country from the In Field drop-down list. Click OK to find the first U.S. record.

9. The form letters go only to customers in the United States, so repeatedly click the Next Record button of the Mail Merge toolbar to find the first U.S. record. Alternatively, type **32** in the text box of the toolbar. The preview of the form letter for Great Lakes Food Market appears as shown in Figure 22.4.

10. You want to send letters to U.S. customers, so you need to create a query that returns only records whose Country column has the value "USA." Close Word, and save your main merge document with an appropriate file name, such as **PurchaseSummary1995H1.doc**. This file is used in the next section, as well as later in the chapter when you open the Access data source from Word.

Note

The Mail Merge Wizard uses DDE to communicate with Word, so you cannot use Word 97's query features to select and sort the merge data. If you attempt to do so, you break the DDE link between Word and Access. Thus, you need to base your final mail merge document on an Access query if you want to select or sort your records.

Oracle Ex. 1009, p. 780
Oracle v. DeBellis
IPR2024-00221

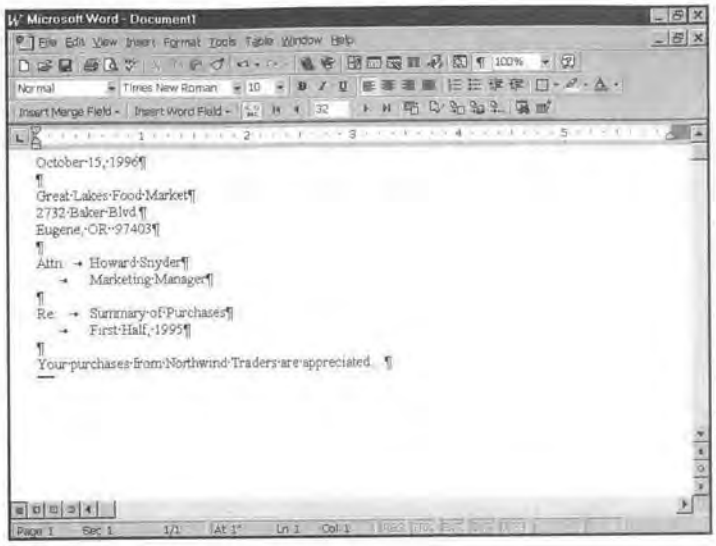


FIG. 22.4 Displaying a preview of a form letter to a U.S. customer.

Using an Existing Main Merge Document with a New Data Source

Once you've created a standard main merge document, the most common practice is to use differing data sources to create form letters by addressee category. Take the following steps to use the main mail merge document you created in the preceding section, PurchaseSummary1995H1.doc, with a new data source based on a simple Access query:

1. Open a new query, and add the Customers table.



2. Add the CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, and Country fields to the query.
3. Enter **USA** as the criterion for the Country field, and clear the Show check box to prevent Country from appearing in the query. Add an ascending sort to the PostalCode field. Your query design appears as shown in Figure 22.5.

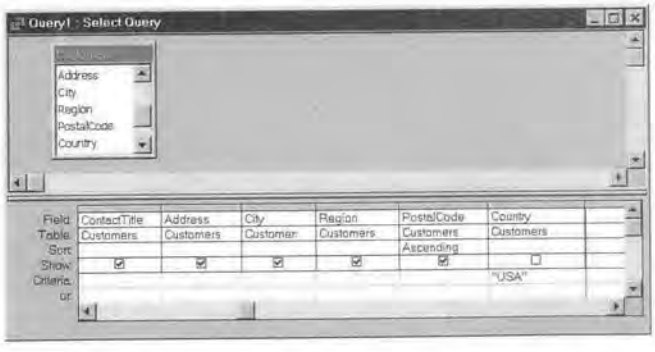


FIG. 22.5 The query design for the U.S. customers mailing list.

Oracle Ex. 1009, p. 781
Oracle v. DeBellis
IPR2024-00221

4. Click the Run button of the toolbar to verify the query result set (see Figure 22.6). Choose File, Save or Save As, and save the query with an appropriate name, such as **qryPurchaseSummaryUSA**.



FIG. 22.6 The query result set for U.S.-based customers.

5. With the query open, choose Tools, Office Links, Merge It with MS Word to launch the Mail Merge Wizard. With the Link Your Data to an Existing Microsoft Word Document option button marked (the default), click OK to display the Select Microsoft Word Document dialog (see Figure 22.7).



FIG. 22.7 Selecting the main merge document.

6. Select your main merge document in the file list and click Open. A message box, shown in Figure 22.8, appears when you change the data source for a merge document. Click Yes to change to the new data source.



7. Confirm that your query is the new merge data source by clicking the Insert Merge Field button and checking the field list. (The Country field shouldn't appear.)



Alternatively, you can click the Edit Data Source button of Word's Mail Merge toolbar to display the query in Access, as shown in Figure 22.9. (Click the Minimize button of Access's Query window, and then click the Word document to return the focus to Word.)

Oracle Ex. 1009, p. 782
Oracle v. DeBellis
IPR2024-00221

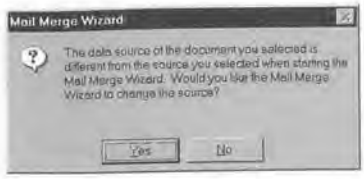


FIG. 22.8 The message box that appears when you change the merge data source.



FIG. 22.9 Displaying the query result set from Word.

8. You can merge the main document and data source directly to the printer or create a series of form letters in a new document. The latter choice lets you inspect the letters before you print them. Click the Merge To New Document button to create the new form letter. The top of the first form letter appears—in ZIP code sequence—as shown in Figure 22.10.

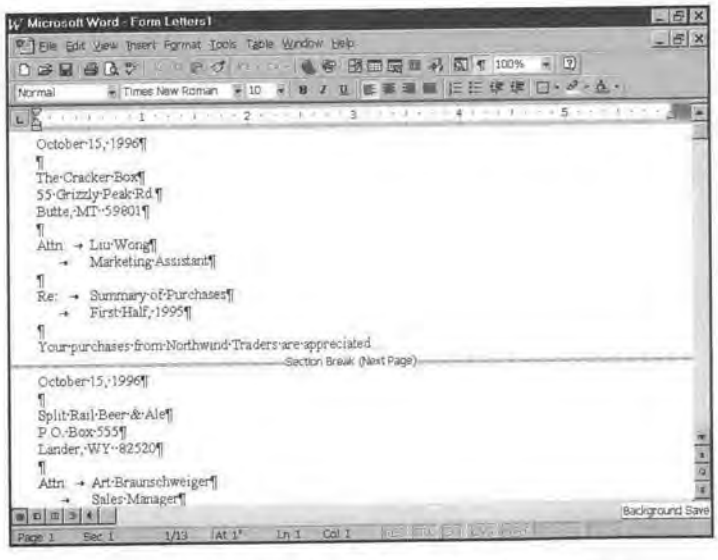


FIG. 22.10 The final version of the form letter addressed to U.S. customers.

Oracle Ex. 1009, p. 783
Oracle v. DeBellis
IPR2024-00221

If you close Word at this point, make sure you save your changes to PurchaseSummary1995H1.doc. This file is used as the main merge document in the sections that follow.

Using Word 97's Mail Merge Feature with Access Databases

In many cases, Access 97 isn't available to Word users who need to create form letters from data contained in Access .mdb files. Office 97 includes Microsoft Query and the necessary 32-bit ODBC drivers to connect to Access .mdb files (all versions) and Microsoft SQL Server databases, plus dBASE, FoxPro, and Paradox 3+ table files. (ODBC drivers for Excel worksheets and text files also are included.) Microsoft Query is modeled on Access's query design window, but Microsoft Query displays the query result set automatically in a separate pane below the design pane as you construct the query. Office 97 applications launch and control Microsoft Query with DDE.



▶▶ See "Understanding ODBC Drivers," p. 918

Note

You need Microsoft Office 97 installed on your PC to create the examples in this section. Access 97 and Jet 3.5 require the 32-bit ODBC 3.0 drivers. Office 97 includes the 32-bit ODBC driver for Access 97 databases. The prior 16-bit Access drivers do not support 32-bit Access 97 .mdb files.

Word 97 includes the Mail Merge Helper, which is similar in concept to an Access wizard. The following three sections use the Mail Merge Helper to create a new Microsoft Query (MSQuery) data source and to use an existing MSQuery data source.

Creating a New Mail Merge Data Source with Microsoft Query and an ODBC Data Source

To use Microsoft Query (MSQuery) to create a merge data source from a Microsoft Access database, follow these steps:

1. Launch Word 97, if necessary, and open the PurchaseSummary1995H1.doc main merge document you created earlier in this chapter in the section "Creating and Previewing a New Form Letter."

Note

ODBC and Microsoft Query must be installed on your computer system before you can use the procedures described in this and the following sections of this chapter. If ODBC and MSQuery are not installed on your computer, rerun the Microsoft Office 97 Setup program and add these items to your system configuration.

Oracle Ex. 1009, p. 784
Oracle v. DeBellis
IPR2024-00221

2. Click the Mail Merge Helper button of the mail merge toolbar to open the Mail Merge Helper dialog (see Figure 22.11). The entry in the Data label of the Data Source section is Northwind.mdb!Query qryPurchaseSummaryUSA. This syntax is used for specifying the topic of a DDE conversation when you use Access as a DDE server.

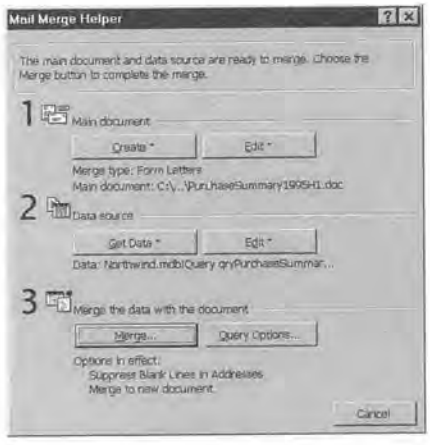


FIG. 22.11 The Mail Merge Helper dialog with an Access DDE merge data source specified.

3. Click the Get Data button, and select Create Data Source from the drop-down list to open the Create Data Source dialog. Word includes a set of default field names you can use to create merge data files (see Figure 22.12).

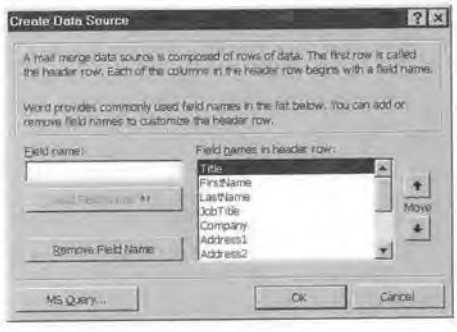


FIG. 22.12 Word 97's Create Data Source dialog.

4. This example uses MSQuery to create the data source, so click the MS Query button to launch MSQuery. MSQuery opens with the Choose Data Source dialog active (see Figure 22.13).

Oracle Ex. 1009, p. 785
Oracle v. DeBellis
IPR2024-00221



FIG. 22.13 MSQuery's Choose Data Source dialog.

5. Click the Databases tab to bring the Databases list to the front of the dialog (if necessary); select <New Data Source> in the Databases list, and then click OK. MSQuery displays the Create New Data Source dialog (see Figure 22.14).

   The Create New Data Source dialog contains four numbered controls: a text box, a drop-down list, a command button, and another drop-down list. When MSQuery first displays the Create New Data Source dialog, only the first text box is enabled. Each successive control is enabled as you complete each item. Figure 22.14 shows the Create New Data Source dialog after filling in all options for this data source. Each numbered control corresponds to an item of information that you must supply for MSQuery to create the data source: the data source's name, the driver for the data source, connection information for connecting to the data source, and an optional default table for the data source.



FIG. 22.14 MSQuery's Create New Data Source dialog.

6. In the first text box, type **PurchaseSummaryUSA** as the name of the new data source. As you enter the data source name, MSQuery enables the drop-down list below it.

7. In the drop-down list (item 2 in the Create New Data Source dialog), select Microsoft Access Driver (*.mdb) as the driver for this data source.

8. Click the Connect command button to display the ODBC Microsoft Access 97 Setup dialog (see Figure 22.15). Click the Select button in the Database frame to display the Select Database dialog.

Oracle Ex. 1009, p. 786
Oracle v. DeBellis
IPR2024-00221

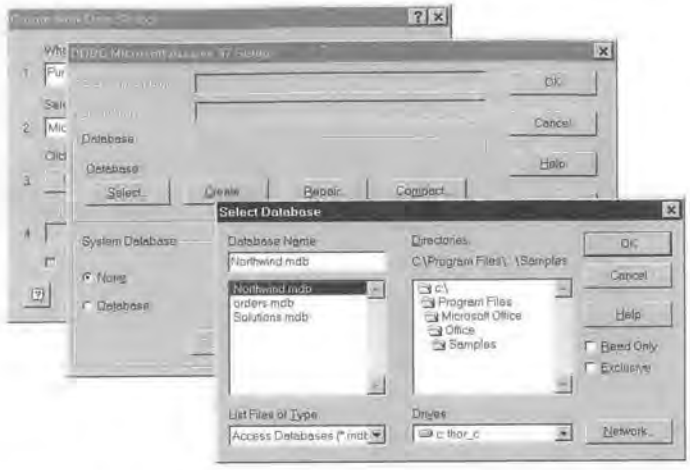


FIG. 22.15 Choosing the Connection for a new MSQuery data source.

9. Maneuver to the folder containing Northwind.mdb (usually C:\Program Files\Microsoft Office\Office\Samples) and select Northwind.mdb in the Database Name list (see Figure 22.15). Click OK to close the Select Database dialog.

10. If you have secured Access or Northwind.mdb, click the Database option button in the System Database frame, and then click the System Database button to open the System Database dialog. Select the System Database you're using, and then click OK to close the dialog. Click the Advanced button to open the Advanced dialog, and enter your login name and password in the text boxes. Then click OK to close the Advanced dialog.

11. Click OK to close the ODBC Microsoft Access 97 Setup dialog. The Create New Data Source dialog now displays the connected database's folder path and file name (or as much of it as will fit) next to the Connect button.

12. Optionally, you can select a table or query from the connected database as the default table for queries created from this data source. In the final drop-down list (numbered 4 in the Create New Data Source dialog), select Customers (refer to Figure 22.14).

13. Click OK to close the Create New Data Source dialog. MSQuery adds the newly created data source to the Databases list in the Choose Data Source dialog.

14. Select PurchaseSummaryUSA in the Databases list of the Choose Data Source dialog and click OK. MSQuery automatically starts its Query Wizard; the first dialog of the Query Wizard is shown in Figure 22.16.

    The Choose Columns dialog of the Query Wizard displays an expandable tree list of all tables and queries in the connected database, with the default table or query's branch selected and expanded for you as shown in Figure 22.16.

Oracle Ex. 1009, p. 787
Oracle v. DeBellis
IPR2024-00221

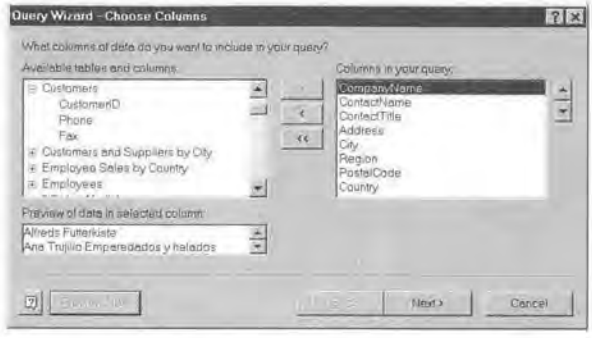


FIG. 22.16 MSQuery's Query Wizard dialog, in which you select the columns for the new query.

15. Select the CompanyName field in the Available Tables and Columns list, and then click the > button to copy the CompanyName field to the Columns in your query list. If you want, you may preview the data in any field by selecting that field in either list and clicking the Preview Now button.

16. Repeat step 15 to add the ContactName, ContactTitle, Address, City, Region, PostalCode, and Country fields to the Columns in your query list (see Figure 22.16). The specific order of the columns is unimportant. Click Next to continue with the second Query Wizard dialog.

17. In the second step of the Query Wizard, you enter criteria to restrict the data retrieved by MSQuery. Select the Country field in the Column To Filter list, then select **equals** in the first drop-down list in the Only Include Rows Where frame, and select **USA** in the second drop-down list (see Figure 22.17). Click Next to continue with the third Query Wizard dialog.

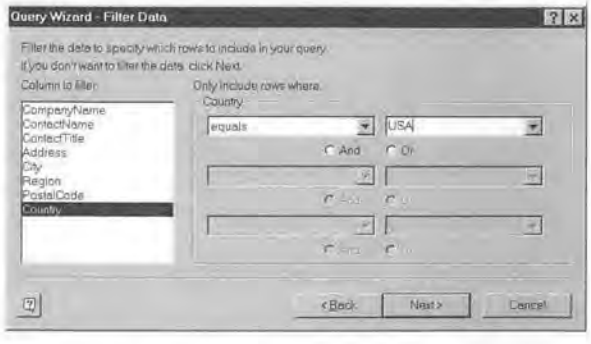


FIG. 22.17 The Filter Data dialog of MSQuery's Query Wizard.

18. In the Query Wizard's Sort Order dialog, you select how you want the retrieved data to be sorted. Select **PostalCode** in the first drop-down list; MSQuery automatically selects the Ascending option button (see Figure 22.18). Click Next to display the final Query Wizard dialog.

Oracle Ex. 1009, p. 788
Oracle v. DeBellis
IPR2024-00221

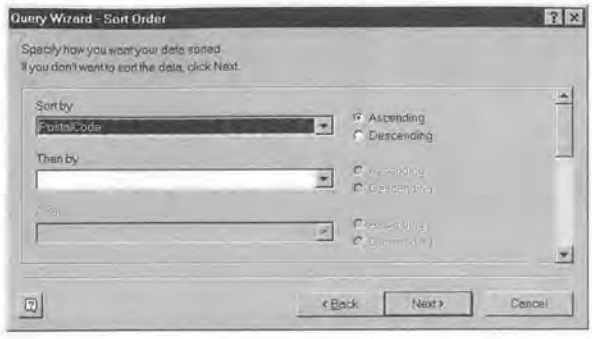


FIG. 22.18 The Sort Order dialog of MSQuery's Query Wizard.

19. Select the View Data or Edit Query in Microsoft Query option, and then click Finish to complete the query (see Figure 22.19). Although you can immediately return data to Word or click the Save Query button to save your query, you should usually take a look at the finished query to make sure that it produces the results you desire.

The completed query is shown in Figure 22.20; notice that the MSQuery query design grid is highly similar to the query design grid in Access, except that the query's results are shown in a table underneath the criteria rows. Use the scroll bars to view the data returned by the query. You can add or edit criteria in MSQuery much the same way you add or edit selection criteria in an Access query. MSQuery, however, uses single quotation marks (') for literal strings (as shown in Figure 22.20) instead of the double quotation marks (") used by Access. Also, MSQuery handles sorting differently than Access—there is no Sort row in the query design grid. Instead, you sort rows by selecting a column in the query's result and then clicking one of the sort order buttons on the toolbar. Essentially, the process is the same as sorting a table's view in Access.



FIG. 22.19 The final dialog of the MSQuery Query Wizard.

Oracle Ex. 1009, p. 789
Oracle v. DeBellis
IPR2024-00221

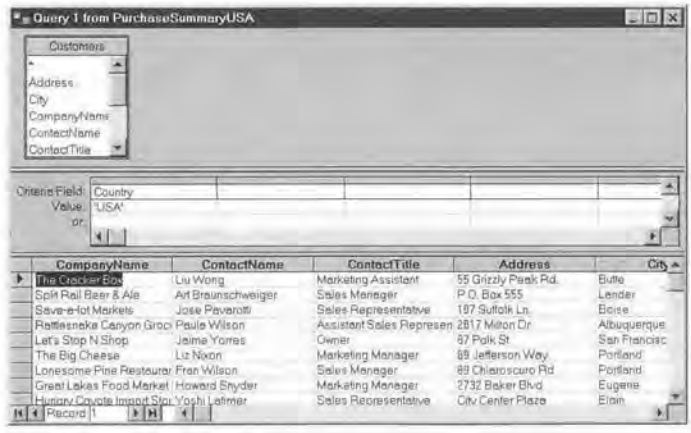


FIG. 22.20 The completed query in MSQuery.

20. Choose File, Save to open the Save dialog. Assign your query a name, such as **PurchaseSummary.dqy**, and click Save. By default, queries are saved in the \Program Files\Microsoft Office\Queries folder. You use the saved query in the section that follows, "Creating Form Letters from an Existing Query."

21. Choose File, Return Data to Microsoft Word to close MSQuery and return to the Mail Merge Helper. The entry in the Data label of the Data Source section is now C:\Program Files\Microsoft Office\Office\Samples\Northwind.mdb. Click the Merge button of Mail Merge Helper to open the Merge dialog.

22. Accept the default New Document selection in the Merge To drop-down list. Click the Check Errors button of the Merge dialog to display the Checking and Reporting Errors dialog. Click the Complete the Merge, Pausing to Report Each Error As It Occurs option button (see Figure 22.21).

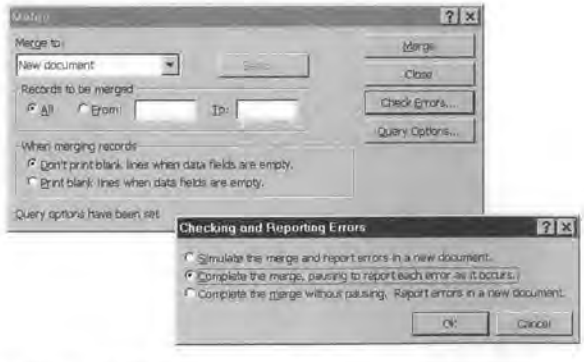


FIG. 22.21 The Merge dialog and the Checking and Reporting Errors dialog.

Oracle Ex. 1009, p. 790
Oracle v. DeBellis
IPR2024-00221

23. Click OK to close the Checking and Reporting Errors dialog, and click Merge to perform the merge. Word finishes merging the documents and displays the final form letter. After reviewing the form letters, close the form letter document; you don't need to save changes.

Creating Form Letters from an Existing Query

Once you've created and saved a query with MSQuery, you can use the saved query to create another set of form letters. MSQuery's saved queries are similar to Access queries saved as `QueryDef` objects in .mdb files. To use an existing .dqy file as the data source for a merge document, follow these steps:

> **Note**
>
> Versions of MSQuery prior to Office 97 saved queries in files with the .qry file extension. MSQuery in Office 97 uses the .dqy file extension, instead.

1. In Word 97, click the Mail Merge Helper button to display the dialog.
2. Click the Get Data button, and select Open Data Source from the drop-down list to display the Open Data Source dialog.
3. Choose the MS Query Files (*.dqy) item in the Files of Type drop-down list.
4. Select the PurchaseSummary.dqy file you saved in the preceding section, and click Open (see Figure 22.22). The message box shown in Figure 22.23 appears.



FIG. 22.22 Choosing an existing MSQuery .dqy file in the Open Data Source dialog.

5. To make PurchaseSummary.dqy the permanent source of data for the PurchaseSummary1995H1.doc main merge document, click the Yes button of the message box to return to the Mail Merge Helper dialog. The path to and the name of your query file appear in the Data label of the Data Source section.

Oracle Ex. 1009, p. 791
Oracle v. DeBellis
IPR2024-00221



FIG. 22.23 Making the .dqy file the permanent source of data for the main merge document.

Embedding or Linking Word Documents in Access Tables

Many word-processing documents are a collection of individual paragraphs, each of which can change depending on the purpose of the document. If the document is a contract, many of the paragraphs are likely to be *boilerplate*: standard paragraphs that are added based on the jurisdiction and purpose of the contract and relationship between the parties. Similarly, books are collections of chapters; when an author is writing a book, each chapter may go through several editing stages. Keeping track of boilerplate files and maintaining collections of book chapter files in various editing stages can be a daunting project. Even if you establish a workable DOS filenaming convention, you can easily lose track of the relationship between the file name and the content of the file.

Applications that track documents and maintain revision records for documents fall into the category of *document management systems*. Document management systems differ from image management systems; the latter systems handle static bitmapped images (usually created by scanners) rather than dynamic document content (editable data). With its OLE 2.1 capability, Access 97 is a logical candidate for the creation of document management applications.

You can create a simple document management system by designing a table with one or more fields of the OLE Object data type to contain embedded documents or links to individual document files. You need a minimum of two other fields: one to identify the source file name of the document and the other to provide a document description. Additional fields can be added to indicate document ownership, track document status, hold key terms, and control who can modify the document. Figure 22.24 shows the design of a simple table that is devised to store the manuscript of this edition in the form of individual chapters in an OLE Object field.

After you define the fields for your document table, you need to determine whether you want to embed the document's data in the table or link the documents to their source files. Make your choice based on the following criteria:

- Embedding the document lets you use in-place activation to review the document within Access. In-place activation is a less-intrusive process.
- Activating a linked document launches Word 97 in its own window.
- Embedding the document provides an independent copy of the document that can serve as an archive. You can set the value of the Locked property of the object to Yes to allow the object to be activated but not altered.

Oracle Ex. 1009, p. 792
Oracle v. DeBellis
IPR2024-00221

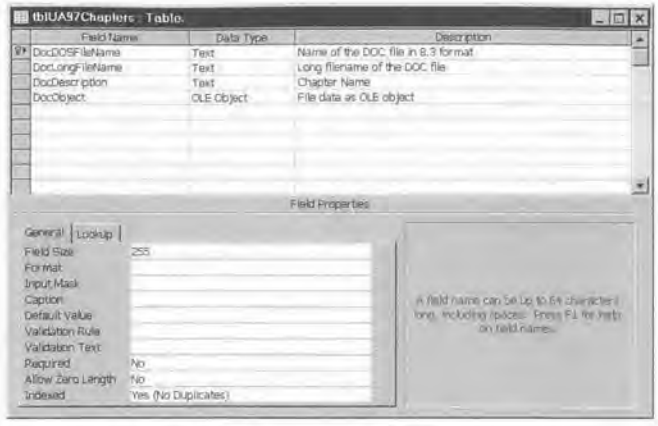


FIG. 22.24 The design of the table for a simple document management system.

- Linking the document allows you to view changes to the document as they occur.
- Linking requires that the document remain in the same location. In most cases, moving the document to another drive or directory breaks the link.
- You cannot save an embedded Word 97 or Excel 97 document to a file or print the embedded document using File menu choices in the in-place activated mode. The File menus of these applications do not replace the File menu of Access 97 when the embedded objects are activated. However, you can open Word's window to make the Word File menu accessible.

> **Note**
>
> You can use Automation instructions in Access VBA modules to save an embedded Word 97 document to a file or to print the document. The Object property of the object frame lets you manipulate embedded or linked objects with Access VBA code. Chapter 31, "Exchanging Data with Automation and ActiveX Controls," describes how to apply Automation methods to Word documents.

Embedding or Linking a Word 97 Document in a Table

To embed or link a Word 97 document in an OLE Object field of a table with a design similar to that shown in Figure 22.24, follow these steps:

1. Place the caret in the OLE Object field, and choose Insert, Object to display the Insert Object dialog (see Figure 22.25).
2. You can create an empty Word document by accepting the default, Create New, and then clicking OK. To link or embed an existing document, click the Create from File option; the Object Type list changes to the File text box. (You don't need to select Microsoft Word Document when you insert an object from a file.)

Oracle Ex. 1009, p. 793
Oracle v. DeBellis
IPR2024-00221

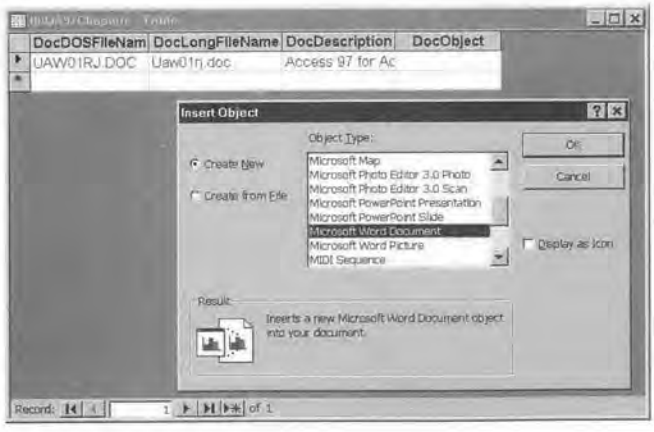


FIG. 22.25 The Insert Object dialog.

3. You can type the path and file name in the File text box or click the Browse button to display the Browse dialog (see Figure 22.26). Select the file you want to use in the File Name list, and then click Open to close the Browse dialog and return to the Insert Object dialog.

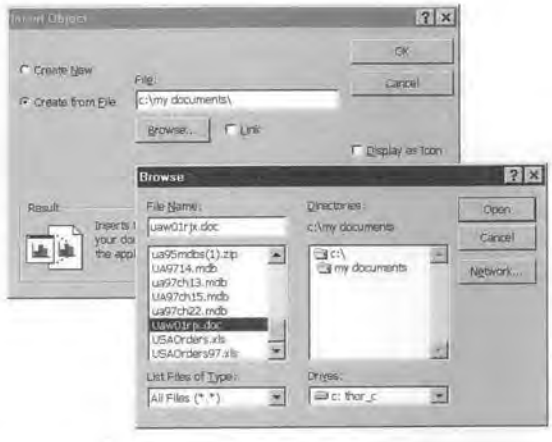


FIG. 22.26 Selecting a source document file in the Browse dialog.

4. The file you selected in the preceding step appears in the File text box. At this point, you can choose between linking and embedding the file. The example that follows uses embedded objects to demonstrate in-place activation (see Figure 22.27). If you want to link the file, mark the Link text box. Click OK.

Oracle Ex. 1009, p. 794
Oracle v. DeBellis
IPR2024-00221



FIG. 22.27 Embedding a Word 97 document object from a file.

5. Position the record selector of the table to a different record to save the embedded object or the link to the object's file in your table, together with its OLE presentation.

Repeat the preceding steps for each document you want to add to the table. You can activate the document object in Word 97's window by double-clicking the OLE Object cell. Viewing the documents you insert in the file lets you verify that their contents correspond to their description.

Troubleshooting

The Microsoft Word 97 Document entry does not appear in the Insert Object dialog's Object Type list, or attempting to insert a Word 97 document results in a message box stating that the registration database entry is invalid or corrupted.

The Registry entries for Word 97 are missing or invalid. If the Word 97 entry is missing, Word's Setup program probably did not complete its operation. (The last step of Setup adds entries to the Registry.) If the "corrupted" message appears, it is likely that you moved the Word files from the original directory in which Setup installed the files into a different directory. In either case, you need to use the Registry Editor (RegEdit.exe) to correct the problem. See the instructions for using the Registry editor in Chapter 19, "Using 32-Bit OLE Components."

Creating a Form to Display the Document

If your table contains only a few fields, you can use the AutoForm feature to create a simple form to display and edit your linked or embedded object. To create the document display form, follow these steps:

1. With the table containing your Word objects, open with the focus in Datasheet view, click the arrow of the New Object button of the toolbar and select AutoForm from the drop-down menu. The Form Wizard automatically creates a standard form.

New Object: AutoForm

Oracle Ex. 1009, p. 795
Oracle v. DeBellis
IPR2024-00221