

# CHAPTER 25



## CONVERTING ACCESS OBJECTS TO DATA ACCESS PAGES

### In this chapter

- Understanding the Limitations of the DAP Conversion Process 1066
- Saving Tables as Data Entry Pages 1067
- Saving Queries as Read-Only Pages 1075
- Saving Forms as DAP 1082
- Saving Reports as DAP 1084
- Saving the Inventory by Category Report to a Page 1091
- Troubleshooting 1097
- In the Real World—Enhancing Data Access Page Design 1098

## UNDERSTANDING THE LIMITATIONS OF THE DAP CONVERSION PROCESS

Access 2003's Save As Data Access Page feature is available for table, query, form, and report objects in conventional Jet .mdb files and ADP. Be forewarned—Save As DAP has a limited repertoire and stumbles when you try to save most Northwind.mdb or Northwind.adp sample form and report objects as pages. On the other hand, Save As DAP is the quickest and easiest way to create simple data entry pages for tables and to display query result sets in a browser. Saving PivotTable and PivotChart views of aggregate queries as pages is the most expeditious method of delivering data analysis pages to intranet users.

### NOTE

SharePoint Team Services 2.0 lists that you import from or link to Jet or SQL server tables and queries are similar in function to data objects saved as DAP. Chapter 26, "Collaborating with SharePoint Team Services 2.0," describes how to create static and updatable lists from Jet and SQL Server 2000 tables and queries.

Like the Save As XML feature described in Chapter 23, "Exporting and Importing Data with XML," Save As DAP relies on ReportML to extract the property values of Access objects. The Save As DAP feature substitutes RPT2DAP.xml for RPT2HTML4.xml and transforms the intermediate XML data in ReportML format to the combination of XML and DHTML code used by DAP. RPT2DAP.xml shares most of RPT2HTML4.xml's limitations on support for native Access controls. The most important of the excluded objects are subform, subreport, tab, and bound and unbound object frame controls. You can use HTML inline frames (<IFRAME> elements) and VBScript code to emulate form/subform combinations, but transforming an existing Access form object to a main page and the subform to a page in an embedded frame usually doesn't succeed. Few conventional bound Access forms are suited to saving as pages directly. If you can adapt the page to a banded design, subordinate sections with navigation controls sometimes can take the place of subforms.

→ To download a Wizard that generates DAP with inline frames for subforms and subreports, see "In the Real World—Enhancing Data Access Page Design," p. 1098.

### TIP

The "Embed an Existing Page in a Data Access Page" article at [http://msdn.microsoft.com/library/en-us/dnacc2k2/html/odc\\_embedpage.asp](http://msdn.microsoft.com/library/en-us/dnacc2k2/html/odc_embedpage.asp) explains how to add an inline HTML frame (<IFRAME>) and use a server filter to emulate a form/subform combination.

Saving reports as pages is a mixed bag. Even simple reports require design tweaks to generate attractive, usable pages. An exception is Northwind.mdb's Invoice report; it transforms to an almost-identical representation of the original. The only missing element is the

DatePrinted value in the Page Header section, which involves a very simple fix. For more complex reports that include subtotals and grand totals, plan on spending at least an hour or two making page design changes. If your reports print calculated values other than standard aggregates of field values, be prepared to write VBScript event-handling code to generate the values.

#### NOTE

The chapter requires familiarity with basic HTML authoring techniques, but you don't need to be a professional Web page designer. Some of the examples include simple VBScript procedures for page navigation and passing parameters from one page to another. You don't need to be an experienced VBScript programmer to follow this chapter's examples.

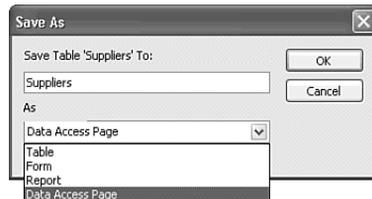
## SAVING TABLES AS DATA ENTRY PAGES

Saving a table as a page is the simplest Save As DAP operation. For example, do the following to save the Suppliers table as a Suppliers.htm page:

1. In the Tables page of Northwind.mdb's Database window, right-click the Suppliers table and choose Save **A**s to open the Save As dialog.
2. Change Copy of Suppliers to **Suppliers**, and select Data Access Page (see Figure 25.1). Click OK to open the New Data Access Page dialog.

**Figure 25.1**

Name the page for the table, and choose Data Access Page in the Save As dialog.



3. Accept Suppliers.htm as the file name, and click OK to create the file in the current folder.
4. After a second or two, the Suppliers page opens in Page view with a caption, data-entry, standard page navigation section (see Figure 25.2).

Opening HTML elements;table cell values> a page in IE 6+ or Page view generates an ADO Recordset object of the updateable snapshot type over the entire table. Specially formatted text boxes (HTML <TEXTAREA> elements) contain cell values of the table.

Following are the default display and editing characteristics of a table saved as a page:

- The datasheet layout determines the column widths of the page. If the table has a lookup field, the column width of the field is expanded, but the foreign key value—not the lookup field value—appears in the column.

- Clicking a field caption sorts the underlying Recordset in ascending order of the field values. For example, clicking Company Name sorts the records in alphabetic order. To restore the original sequence, click the Supplier ID column.
- The default record navigation control displays 10 records at a time.
- Recordset navigation progresses by 10-record groups and uses the First, Previous, Next, and Last buttons.
- Record-selector buttons are bitmap images stored in the Office Web Components (OWC) Data Source control (MSODSC). When you select a record by clicking the record-selector button, the black arrow changes to white and the background changes to a darker shade of gray.
- The Tab key navigates from left to right by column and the top to bottom by row. Fields with locked values—such as Jet AutoNumber fields—aren't tab stops. When you reach the last column of the last visible row, the focus changes to the navigation control buttons. Pressing Tab when the focus is on the last navigation control moves to the first editable column of the first row.
- The ↓ and ↑ keys don't navigate rows in the current columns. You can assign key names as the value of the AccessKey property of a text box to enable navigation by Alt+Key down—but not up—the rows of the specified column. The later “Streamlining Data Entry” section describes how to set AccessKey values for columns.
- The Tab key positions the insertion point after the last character of an editable cell. You must manually select the entire field (press Shift+Home or Shift+↑) if you want the text that you type to replace—rather than extend—the existing value.
- Jet Hyperlink fields display the text but not the link; thus, Jet hyperlinks aren't operable. This is the opposite of the behavior of Hyperlink fields that you add to DAP from the Field List.
- Field- and table-level validation rules for Jet tables and CHECK CONSTRAINTS for SQL Server tables work as expected, and referential integrity rules are enforced. For example, attempting to delete a Supplier record that has dependent records in the Products table displays the messages shown in Figure 25.3.

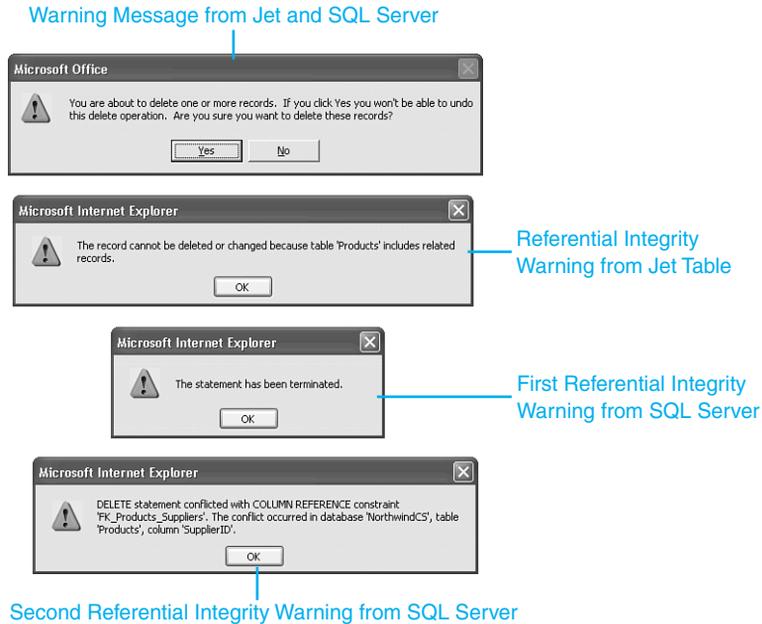
**Figure 25.2**

Page view of a table saved as a page is a reasonably close rendition of datasheet view, considering the limitations of HTML. The Suppliers datasheet uses default formatting, which displays a plain-vanilla page.

Supplier ID	Company Name	Contact Name	Contact Title
1	Exotic Liquids	Charlotte Cooper	Purchasing Manager
2	New Orleans Cajun Delights	Shelley Burke	Order Administrator
3	Grandma Kelly's Homestead	Regina Murphy	Sales Representative
4	Tokyo Traders	Yoshi Nagase	Marketing Manager
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Export Administrator
6	Mayumi's	Mayumi Ohno	Marketing Represent
7	Pavlova, Ltd.	Ian Dewling	Marketing Manager
8	Specialty Biscuits, Ltd.	Peter Wilson	Sales Representative
9	PB Knäckebröd AB	Lars Peterson	Sales Agent
10	Refrescos Americanas LTDA	Carlos Diaz	Marketing Manager

**Figure 25.3**

Users receive a warning message before deleting a record. If deleting the record violates referential integrity rules, a single message from Jet or two messages from SQL server explain the reason for denying the deletion operation.



- The Input Mask property of fields is ignored. You must write VBScript event-handling code to implement after-entry input masks, such as the >LLLLL mask for the CustomerID field of the Customers table. For this example, the event handler must apply the ucase function to the entry, test for the correct number of characters, and use the asc function to verify that only letters are present.

**TIP**

For a VBScript workaround for lack of Input Mask capability in pages, see the "ACC2002: Data Access Page Ignores Underlying Table's Input Mask" Knowledge Base article at <http://support.microsoft.com/support/kb/articles/Q299/0/09.ASP>.

Web pages aren't as effective as Access datasheets for bulk data-entry operations because keyboard shortcuts for editing operations—such as copying field values from the preceding row—aren't available in IE.

## ENHANCING THE PAGE'S APPEARANCE

A few simple design changes can dress up the table-based page and improve the usefulness of the display. The following sections offer suggestions for making table-based pages resemble the style of pages that you create in Page Design view or that the Layout Wizard creates for you.

## ADDING HEADING ELEMENTS WITH THE SCRIPT EDITOR

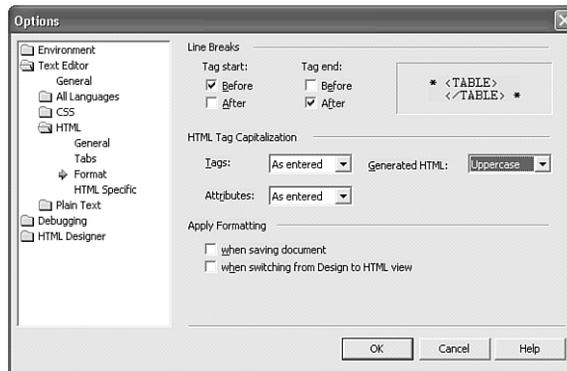
Unlike pages that you create from scratch, pages saved from datasheets don't have empty `<H1>` and `<H2>` elements for adding a heading and explanatory text to the table. Adding elements to table-based pages requires adding HTML code with the Script Editor.

→ To review use of the Microsoft Script Editor, see “The Microsoft Script Editor,” p. 1012.

To add `<H1>` and, optionally, `<H2>` elements, do the following:

1. With the Suppliers page open, change to Page Design view and click the Microsoft Script Editor button to open the source window for your page.
2. By default, the Script Editor adds HTML tag names in lower case, which conforms to XHTML standards; RPT2DAP.xsl generates upper-case tag names. To change the tag name case, choose **T**ools, **O**ptions, to open the Options dialog. Expand the Text Editor and HTML nodes, and select Format.
3. Select Uppercase in the Generated HTML list. If you want to make your changes more evident, mark the Tag Start: Before and Tag End: After check boxes (see Figure 25.4). Click OK to close the Options dialog.

**Figure 25.4**  
The Script Editor defaults to lowercase tags. You change to uppercase tags in the Option Dialog's HTML Format page. You also can specify line breaks in the code before and after tag pairs.



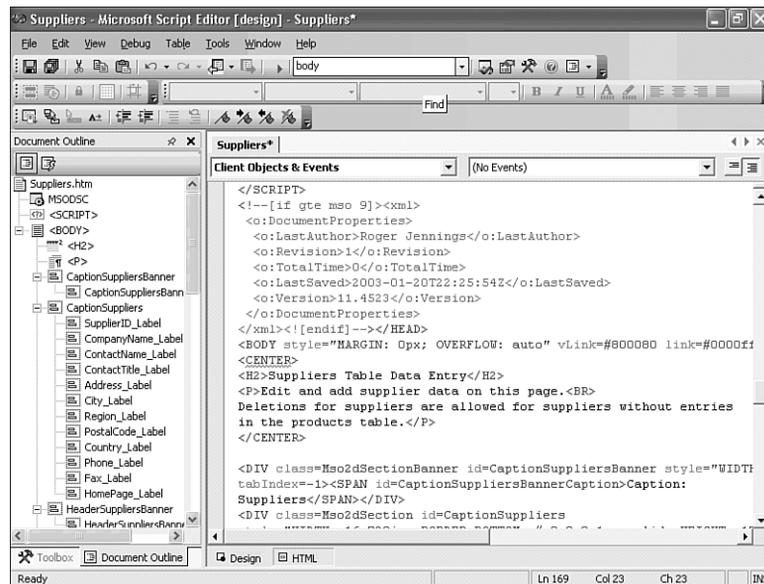
### TIP

You can reduce toolbar clutter by choosing **V**iew, **T**oolbars, and toggling the Design, Formatting, and Text Editor toolbars off. Only the Standard toolbar is useful for the HTML editing examples in this chapter.

4. After the `<TITLE>` tag, change Suppliers to **Suppliers Table Data Entry**.
5. Press Ctrl+F to open the Find dialog, type **body**, and press Alt+F twice to locate the `<BODY>` tag that follows the `</HEAD>` tag.
6. Move to the end of the line, and press Enter to create a new line.

7. Type `<h2>`, which opens the HTML statement-completion window and selects the H2 item. Double-click H2 and type `>` to create an `<H2></H2>` tag pair. Type the title that you added in step 4 between the tag pair as the heading for the form. As you add the HTML content, Page Design view displays the changes.
8. Optionally, add a `<P></P>` tag element;adding to table-based pages> element;adding to table-based pages> pair with explanatory text after the heading. To center the lines, type a `<CENTER>` tag before the `<H2>` element, and type `</CENTER>` after the `</P>` element. Remove the extra `</CENTER>` and `<CENTER>` tags that the statement completion feature adds (see Figure 25.5), and close the Script Editor.

**Figure 25.5**  
The HTML code between the `<CENTER>` and `</CENTER>` tags adds an 18-point heading and a text paragraph to the page.



### TIP

Close the Script Editor before making further changes in Page Design view. If you leave the Script Editor open it's difficult—and sometimes impossible—to alter control positions or sizes.

- B** 9. Return to Access, select all captions, and click the Bold button. Press Ctrl+S to save your changes.
- E** 10. Change element;adding to table-based pages> element;adding to table-based pages> to Page view to check your work so far (see Figure 25.6).

**Figure 25.6**

The added HTML code of Figure 25.5 provides the heading and text shown here. After you add the elements, you can edit their text in Page Design view.

Supplier ID	Company Name	Contact Name	Contact Title
1	Exotic Liquids	Charlotte Cooper	Purchasing Manager
2	New Orleans Cajun Delights	Shelley Burke	Order Administrator
3	Grandma Kelly's Homestead	Regina Murphy	Sales Representative
4	Tokyo Traders	Yoshi Nagase	Marketing Manager
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Export Administrator
6	Mayumi's	Mayumi Ohno	Marketing Representatir
7	Pavlova, Ltd.	Ian Dewling	Marketing Manager
8	Specialty Biscuits, Ltd.	Peter Wilson	Sales Representative
9	PB Knäckebröd AB	Lars Peterson	Sales Agent
10	Refrescos Americanas LTDA	Carlos Diaz	Marketing Manager

**NOTE**

If you've specified a default theme, the styles of the theme determine the color, font, and size attributes of the elements that you add. Otherwise, the attributes default to the standard style for DAP.

**MODIFYING FIXED PAGE LAYOUT AND DESIGN ELEMENTS**

A few additional design changes make the page more attractive and make the appearance conform to the DAP that you created from scratch in Chapter 25, “Designing and Deploying Data Access Pages.” The sections of DAP that you save from Access objects have fixed positions and dimensions, so you must change section property values to alter the overall design of the page.

To make the form-based design of saved datasheets resemble the standard format of the other pages you've created, using the Suppliers.htm page as an example, do the following:

-  1. Return to Page Design view, and double-click the Caption: Suppliers section heading to open its properties window.
2. Change the BackgroundColor property value to **steelblue** and the Height to **25px**.
-   3. Select all the captions, open the Fore/Font Color picker, and click the white square.
4. With all captions selected, drag the captions down one grid dot.
5. Change the Supplier ID caption to ID and decrease its width to fit the text.
6. Decrease the width of the Supplier ID text box to match the width of the ID caption, and select and drag the remaining text boxes to the left. Dragging the text boxes also drags the labels.
7. The Home Page hyperlink field isn't functional in the Suppliers page, because the page link isn't present. Delete the field and its caption.

8. The width of the active area of the page is set by the fixed `Width` property of the record navigation section. Double-click the `NavigationSuppliers` section, and set the `Width` property value to **13.75in**.
9. To add a small margin around the page elements, open the Script Editor and change the `<BODY style="MARGIN: 0px` attribute to **17px**.
10. Press `Ctrl+S` to save the edits, close the Script Editor, and change to Page view to review your work (see Figure 25.7).

**Figure 25.7**  
Adding a background color to the caption, reducing the width of the Supplier ID column, and adding a margin improves the appearance of the page.

ID	Company Name	Contact Name	Contact Title
1	Exotic Liquids	Charlotte Cooper	Purchasing Manager
2	New Orleans Cajun Delights	Shelley Burke	Order Administrator
3	Grandma Kelly's Homestead	Regina Murphy	Sales Representative
4	Tokyo Traders	Yoshi Nagase	Marketing Manager
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Export Administrator
6	Mayumi's	Mayumi Ohno	Marketing Representative
7	Pavlova, Ltd.	Ian Devling	Marketing Manager
8	Specialty Biscuits, Ltd.	Peter Wilson	Sales Representative
9	PB Knäckebröd AB	Lars Peterson	Sales Agent
10	Refrescos Americanas LTDA	Carlos Diaz	Marketing Manager

## STREAMLINING DATA ENTRY

The width of the sections exceeds the visible page widths on ordinary monitors, so editing values in more than one row of columns that aren't visible is a painfully slow process. Tabbing through columns and then rows causes the display to flash, which leads to data-entry operator fatigue. To navigate by rows of a particular column, you add `AccessKey` property values to each updateable field.

Access key column navigation is limited to the number of rows displayed by the Header section. The 10-row default creates a page that resembles a form more than a Web page. Forms usually have about a 4:3 width-to-height aspect ratio to avoid scrolling the form, but users are accustomed to scrolling Web pages vertically. Displaying more rows of the table in a group aids navigation, especially when combined with column access keys.

To add and test `AccessKey` values for the columns of the sample `Suppliers.htm` page and increase the number of rows on the page, do this:

1. In Page Design view, double-click the Company Name text box to open the Text Box: `CompanyName` window.
2. Click the Other tab and type **1** as the `AccessKey` value. Alternatively, type a letter that's easier to reach with a thumb or finger on the `Alt` key.

3. Select each text box in sequence, and specify a different `AccessKey` value. This example uses 1, 2, 3, ... 0 for the 10 active fields of the Suppliers table. (SupplierID is an AutoNumber or identity field, so it isn't active.)
4. Right-click the page and choose Group Level Properties to open the Group Level: Suppliers window, and set the `DataPageSize` property value to a larger number. Select All if you want to scroll the entire Recordset.
5. Press Ctrl+S to save your changes, and then open the page in Web Page Preview. Click the Company Name header to sort the page by company.
6. Press Alt+5 (or the letter that you chose) to move the insertion point to the end of the City name in the first row (see Figure 25.8). Pressing the same key combination repeatedly scrolls the City column. When you reach the last row of a list that's less deep than IE's window, the insertion point returns to the first row of the City column. Otherwise the insertion point returns to the first row of the page.

**Figure 25.8**

You can sort the page by clicking a column heading or selecting a column and clicking the A-Z or Z-A buttons. This view is sorted by city. Pressing Alt+AccessKey navigates horizontally to the designated column and then vertically through the rows.

Company Name	Contact Name	Contact Title	Address	City
Homestead	Regina Murphy	Sales Representative	707 Oxford Rd.	Ann Arbor
	Eliane Noz	Sales Representative	Bat. B	Anney
	Cheryl Saylor	Regional Account Rep.	3400 - 8th Avenue	Bend
mbH & Co. KG	Petra Winkler	Sales Manager	Tiergartenstraße 5	Berlin
Food Cannery	Robb Merchant	Wholesale Account Agent	Order Processing Dept.	Boston
Handelsgesellschaft mbH	Sven Petersen	Coordinator Foreign Markets	Frahmredder 112a	Cuxhaven
ITB	Martin Bein	International Marketing Mgr.	Bogenallee 51	Frankfurt
KB	Lars Peterson	Sales Agent	Kaloadaqatan 13	Göteborg
	Anne Heikkonen	Product Manager	Valtakatu 12	Lappeenranta
	Charlotte Cooper	Purchasing Manager	49 Gilbert St.	London
	Niels Petersen	Sales Manager	Lynqbyssid	Lynqby
s, Ltd.	Peter Wilson	Sales Representative	29 King's Way	Manchester
	Ian Devling	Marketing Manager	74 Rose St.	Melbourne
aux	Marie Delamare	Sales Manager	22, rue H. Voiron	Montceau
	Jean-Guy Lauzon	Marketing Manager	2960 Rue St. Laurent	Montréal
un Delights	Shelley Burke	Order Administrator	P. O. Box 78934	New Orleans
	Mayumi Ohno	Marketing Representative	92 Setsuko	Osaka
uesos 'Las Cabras'	Antonio del Valle Saavedra	Export Administrator	Calle del Rosal 4	Oviedo

Unfortunately, you can't freeze fields to make the Supplier ID and Company Name field visible while you edit other fields.

It's a good practice to delete buttons in the `PageNameNavigation` section that aren't necessary for data entry. For example, the Help button opens the "About Data Access Pages" help topic; installing the Office Web Components (OWC) 11 runtime on client PCs adds this help file. The help file contains more information than most data-entry operators need. Depending on your application, you might want to remove the filter buttons. To remove a button, in Page Design view, select the button and press Delete. Removing unneeded buttons also improves page performance.

→ To make the data entry page available to users, see “Delivering Pages on an Intranet or the Internet,” p. 1057.



If you encounter problems opening the page from a networked computer other than the one on which you're designing the page, see the “Data Source Problems” topic of the “Troubleshooting” section near the end of the chapter.

## SAVING QUERIES AS READ-ONLY PAGES

Saving query datasheets as DAP is an alternative to creating the live Web reports described in Chapter 23. If the query is updateable, you can edit the result set by the methods described in the preceding section. In most cases, however, pages based on queries are read-only, even if the result set is updateable.

Datasheets with special formatting preserve the format by changing the Cascading Style Sheet (CSS) style for text boxes. For an example, save the Customers and Suppliers by City UNION query to a page that closely resembles the query's Datasheet view. To make the entire text box visible, you must increase the depth of the Header section from 17 to about 21 pixels and then expand the depth of text boxes, which appear as command buttons in Page Design view. On the whole, special effects applied to datasheets are less than special in pages.

### DESIGNING THE QUERY LAYOUT FOR DAP

If you intend to save a query as a page, it's faster to change the column captions, when possible, and the column widths in Datasheet view. Changing the default Medium Date to Short Date format makes date columns consistent with other pages that use MM/DD/YYYY format. Figure 25.9 shows a sample Jet qryOrdersPage query in Datasheet view; column widths are minimized to ensure full display within IE 6+ maximized 800×600 monitor resolution. Applying a descending sort on the OrderID column displays latest orders first.

**Figure 25.9**  
Adjusting column widths, captions, and formatting of queries before you save the page minimizes redesign effort. For orders, it's usually preferable to set the default sort order to descending.

OrderID	Company Name	Order	Ship	Amount	Freight	Company Name
11077	Rattlesnake Canyon Grocery	05/06/1998		\$1,256.72	\$8.63	United Package
11076	Bon app'	05/06/1998		\$792.75	\$38.28	United Package
11075	Richter Supermarkt	05/06/1998		\$498.10	\$6.19	United Package
11074	Simons bistro	05/06/1998		\$232.08	\$18.44	United Package
11073	Pericles Comidas clásicas	05/05/1998		\$300.00	\$24.95	United Package
11072	Ernst Handel	05/05/1998		\$5,218.00	\$268.64	United Package
11071	LILA-Supermercado	05/05/1998		\$484.50	\$0.93	Speedy Express
11070	Lehmanns Marktstand	05/05/1998		\$1,629.97	\$136.00	Speedy Express
11069	Tortuga Restaurante	05/04/1998	05/06/1998	\$360.00	\$15.67	United Package
11068	Queen Cozinha	05/04/1998		\$2,027.08	\$81.75	United Package
11067	Drachenblut Delikatessen	05/04/1998	05/06/1998	\$86.85	\$7.98	United Package
11066	White Clover Markets	05/01/1998	05/04/1998	\$928.75	\$44.72	United Package
11065	LILA-Supermercado	05/01/1998		\$189.42	\$12.91	Speedy Express
11064	Save-a-lot Markets	05/01/1998	05/04/1998	\$4,330.40	\$30.09	Speedy Express
11063	Hungry Owl All-Night Grocers	04/30/1998	05/06/1998	\$1,342.95	\$81.73	United Package
11062	Reggiani Caseifici	04/30/1998		\$406.40	\$29.93	United Package
11061	Great Lakes Food Market	04/30/1998		\$510.00	\$14.01	Federal Shipping
11060	Franchi S.p.A.	04/30/1998	05/04/1998	\$266.00	\$10.98	United Package

When you save the query as a page, RPT2DAP.xsl might transform the design incorrectly. In this case, the Order ID caption doesn't fit the allotted space, so the caption and all rows have a two-line depth (see Figure 25.10).

**Figure 25.10**  
RPT2DAP.xsl interprets the insufficient width for the Order ID caption as requiring an increase in the depth of the rows.

Order ID	Company Name	Order	Ship	Amount	Freight	Company Name
11077	Rattlesnake Canyon Grocery	05/06/1998		\$1,255.72	\$8.53	United Package
11076	Bon app'	05/06/1998		\$792.75	\$38.28	United Package
11075	Richter Supermarkt	05/06/1998		\$498.10	\$6.19	United Package
11074	Simons bistro	05/06/1998		\$232.08	\$18.44	United Package
11073	Pericles Comidas clásicas	05/05/1998		\$300.00	\$24.95	United Package
11072	Ernst Handel	05/05/1998		\$5,218.00	\$258.64	United Package
11071	LILA-Supermercado	05/05/1998		\$484.50	\$0.93	Speedy Express
11070	Lehmans Marktstand	05/05/1998		\$1,629.97	\$136.00	Speedy Express
11069	Tortuga Restaurante	05/04/1998	05/06/1998	\$360.00	\$15.67	United Package

The solution to this problem is to change Order ID to **ID** in the caption, and delete the Order ID text in the first text box of the header. (Field names aren't necessary in Page Design view of text boxes.) You don't need record-selector buttons for a read-only query, so right-click the page, choose Group Level Properties, and set the RecordSelector property value to False. After making design changes similar to those recommended in the preceding two sections, your page appears as shown in Figure 25.11. The New, Save, Undo, and Help buttons are deleted because the query result set is read-only.

**Figure 25.11**  
Design changes for table-based pages described earlier have been applied to the query-based page shown here. When you remove the record-selector buttons, the left border of the text box for the first column is missing.

ID	Customer	Order	Ship	Amount	Freight	Ship By
11077	Rattlesnake Canyon Grocery	05/06/1998		\$1,255.72	\$8.53	United Package
11076	Bon app'	05/06/1998		\$792.75	\$38.28	United Package
11075	Richter Supermarkt	05/06/1998		\$498.10	\$6.19	United Package
11074	Simons bistro	05/06/1998		\$232.08	\$18.44	United Package
11073	Pericles Comidas clásicas	05/05/1998		\$300.00	\$24.95	United Package
11072	Ernst Handel	05/05/1998		\$5,218.00	\$258.64	United Package
11071	LILA-Supermercado	05/05/1998		\$484.50	\$0.93	Speedy Express
11070	Lehmans Marktstand	05/05/1998		\$1,629.97	\$136.00	Speedy Express
11069	Tortuga Restaurante	05/04/1998	05/06/1998	\$360.00	\$15.67	United Package
11068	Queen Cozinha	05/04/1998		\$2,027.08	\$81.75	United Package
11067	Drachenblut Delikatessen	05/04/1998	05/06/1998	\$86.85	\$7.98	United Package
11066	White Clover Markets	05/01/1998	05/04/1998	\$928.75	\$44.72	United Package
11065	LILA-Supermercado	05/01/1998		\$189.42	\$12.91	Speedy Express
11064	Save-a-lot Markets	05/01/1998	05/04/1998	\$4,330.40	\$30.09	Speedy Express
11063	Hungry Owl All-Night Grocers	04/30/1998	05/06/1998	\$1,342.95	\$81.73	United Package
11062	Reggiani Caseifici	04/30/1998		\$406.40	\$29.93	United Package
11061	Great Lakes Food Market	04/30/1998		\$510.00	\$14.01	Federal Shipping
11060	Franchi S.p.A.	04/30/1998	05/04/1998	\$266.00	\$10.98	United Package
11059	Ricardo Adocicados	04/29/1998		\$1,838.00	\$85.80	United Package

#### TIP

As an alternative to setting all margins at once by changing the `<BODY style="MARGIN: 0px` attribute value, you can change individual margins on the Format page of the Page: *PageName* window. Set the `MarginLeft` and `MarginTop` property values to **17px** (or more).

To add a left border to the text boxes in the Header section, double-click the text box with the missing border—in this case, ID—to open its properties window. Delete the last #000000 value of the `BorderColor` property and delete the last none value of `BorderStyle`.

**TIP**

To minimize active connections to the database, open the Page: *PageName* properties sheet, click the Data tab, and use the `RecordsetType` property's list box to change the default `dscUpdateableSnapshot` to `dscSnapshot`.



The `qryOrdersPage` query and a Current Orders link to the `CurrentOrders.htm` page are in the `Data25.mdb` file located in the `\Seu11\Chaptr25` folder of the accompanying CD-ROM.

## WORKING WITH PARAMETERIZED QUERIES

Users opening pages based on queries download the entire query result set to a local `Recordset`, unless you add a `TOP n` modifier or a `WHERE` clause to limit the number of rows. Alternatively, you can add a parameter to the query to limit the `Recordset` size. For this example, the parameter is the earliest order date to view. To avoid errors that result from not typing the parameter value in the exact date format, add a hidden `OrderDate` column and type `>=[Enter Earliest Order Date (M/D/YYYY)]` or the like in the Criteria row of the query. (Keep your parameter captions short for pages.)

### Jet SQL

The Jet SQL statement for the `qryOrdersPageParam` parameter query used in the following sections is:

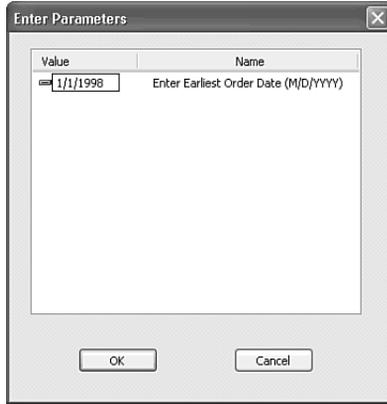
```
PARAMETERS [Enter Earliest Order Date (M/D/YYYY)] DateTime;
SELECT Orders.OrderID, Customers.CompanyName,
       Format([OrderDate], "mm/dd/yyyy") AS [Order],
       Format([ShippedDate], "mm/dd/yyyy") AS Ship,
       [Order Subtotals].Subtotal AS Amount,
       Orders.Freight, Shippers.CompanyName
FROM Shippers
     INNER JOIN (Customers
                INNER JOIN (Orders
                           INNER JOIN [Order Subtotals]
                                ON Orders.OrderID = [Order Subtotals].OrderID)
                ON Customers.CustomerID = Orders.CustomerID)
     ON Shippers.ShipperID = Orders.ShipVia
WHERE (((Orders.OrderDate)>=[Enter Earliest Order Date]))
ORDER BY Orders.OrderID DESC;
```

### SUPPLYING PARAMETER VALUES IN THE ENTER PARAMETERS DIALOG

When you open the page in Page View or Web Page Preview, a nonstandard Enter Parameters dialog appears first. The dialog can accommodate about 12 parameter values, but the Name list doesn't expand to accommodate fully descriptive parameter captions. Figure 25.12 shows the Enter Parameters dialog for the `CurrentOrdersParam.htm` page.

**Figure 25.12**

A single parameters dialog replaces one or more Access Enter Parameter Value dialogs. The first Microsoft mystery of this chapter is why the parameter names are centered instead of left-justified in the list.



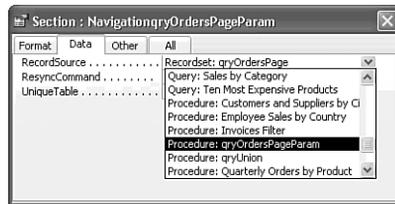
Typing the required value and clicking OK (or pressing Enter twice) sends the parameter value(s) to Jet and opens the page. If you don't enter a value and click OK, no rows appear. If you click Cancel, the page opens with #Name? values in a single row. The parameter entry procedure is the same for SQL Server parameterized stored procedures and table-returning functions.

**TIP**

If you copy the parameterized query with a different name—such as qryOrdersParam—and use Save As to save the link and page with a different link and .htm file name, you must change the RecordSource property value on the Data page of the Section: NavigationQueryName window to point to the parameterized query. Open the RecordSource list, and scroll to the Procedure: items—not Query: items—to find the new parameterized query (see Figure 25.13). The Procedure: prefix applies to Jet parameterized queries and SQL Server stored procedures with parameters.

**Figure 25.13**

When you change the RecordSource property from a conventional query to a parameterized query, select the query name from the items with the Procedure: prefix.



The qryOrdersPageParam query and a Current Orders (Param) link to the CurrentOrdersParam.htm page are in the Data25.mdb file located in the \Seua11\Chaptr25 folder of the accompanying CD-ROM.

## PASSING FILTER CRITERIA FROM ONE PAGE TO ANOTHER WITH COOKIES

The Enter Parameters dialog isn't user-friendly, and it limits the explanation of the required parameter entry to a few words. A better alternative is an initial page in which users type one or more parameter values in text box(es), and then click a button or a label formatted as a hyperlink to open a second page that displays the data. The `onClick` event handler of the button or label writes a cookie with the parameter names and values, and then opens the other page. The second page retrieves the cookie and passes the values as query parameter values or the criterion value for the `DataSourceControl`'s `ServerFilter` property. One of the advantages of using an initial page is that you can provide a default value for the parameter in a text box.

### NOTE

The `DataSourceControl`'s `ServerFilter` property is similar to the `Filter` property of a table or query. The `Filter` property downloads the entire `Recordset` and then applies the filter on the client. Applying a `ServerFilter` downloads only those records that meet the filter criterion, which minimizes network traffic and speeds response time.

Cookies are a means for passing variable values between conventional HTML forms. Cookies are `Name=Value` pairs—called *crumbs*—separated by semicolons. Cookies can contain up to 20 crumbs. Unless you add an `expires=datetime` crumb, a cookie ordinarily stays in memory only while the browser is open. In Access, the cookie stays in memory while the page is open in Page or Page Design view. Cookies are properties of the browser's document object; you create a cookie with a VBScript `document.cookie = "Name=Value"` statement and retrieve it with a `strVariable = document.cookie` statement. Obtaining the value of a crumb requires parsing the cookie with VBScript string-manipulation functions. VBScript doesn't support all Jet/VBA string-manipulation functions; those that VBScript does support—with a few exceptions—use the same syntax as their Jet/VBA counterpart.

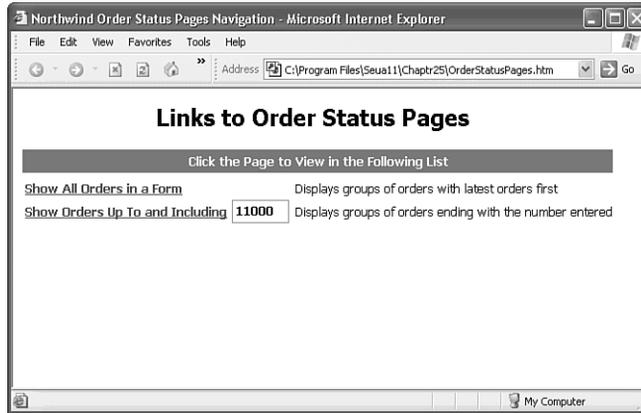
### NOTE

Like VBA, VBScript isn't case sensitive. Web programmers use a variety of capitalization styles when writing VBScript code. A common style is to use lower case for VBScript key words and object, property, and method names defined by the Document Object Model. Mixed case is used for variable names and other object names—such as `MSODSC.RecordsetDefs` for the `RecordsetDefs` collection of the Microsoft Office Data Source Control. This book uses mixed case for VBScript.

VBScript has only one data type—`Variant`. The VBScript examples of this chapter use variable-type prefixes—such as `dat` for date and `str` for character variables—for consistency with the VBA code that you encounter in Part VII, "Programming and Converting Access Applications."

You can create the initial page in the Access page designer or any other HTML editor, such as FrontPage 2003. The initial page might include navigation buttons or labels for several related pages. Figure 25.14 shows a simple navigation form with links to the `CurrentOrders.htm` and `CurrentOrdersCookie.htm` pages.

**Figure 25.14** OrderStatusPages.htm is an unbound navigation page that has a link to CurrentOrders.htm and a text box for entering the earliest order parameter required when clicking the link to the CurrentOrdersCookie.htm page.

**TIP**

When you deploy a navigation page with links to other pages, create a virtual directory for it and all related pages in Internet Information Server (IIS). Rename the navigation page to Default.htm. Users enter `http://servername/directoryname/` to open Default.htm; `filename.htm` isn't required.

The hyperlinks of the navigation page of Figure 25.14 are formatted labels (blue color, underlined). The Show All Orders in a Form link's `onclick` event handler contains the following VBScript code to open the CurrentOrders.htm page:

```
<SCRIPT language=vbscript event=onclick for=lnkAllOrders>
<!--
    dim strURL
    strURL = "CurrentOrders.htm"
    window.navigate(strURL)
-->
</SCRIPT>
```

**NOTE**

The preceding event-handler code is typical for links or buttons on pages that perform navigation functions similar to that of Northwind.mdb's Main Switchboard form.

The Show Orders Up To and Including... link (`lnkLatestOrders`) points to CurrentOrdersCookie.htm and has a text box (`txtEndDate`) that contains a default value appropriate for the range of the latest orders in the Northwind.mdb Orders table. The following code illustrates how to create the parameter cookie (`LastOrder=11000`) before opening the CurrentOrdersCookie.htm page.

```
<SCRIPT language=vbscript event=onclick for=lnkLatestOrders>
<!--
    Dim strURL
```

```

Dim lngLastOrder
Const strParam1 = "LastOrder"

strURL = "CurrentOrdersCookie.htm"
lngLastOrder = txtLastOrder.value
document.cookie = strParam1 & "=" & lngLastOrder
window.navigate(strURL)
-->
</SCRIPT>

```

The following VBScript added to the CurrentOrdersCookie.htm page reads the cookie, extracts the date from the cookie text, and adds a ParameterValues object to the MSODSC.RecordsetDefs collection.

```

<SCRIPT language=vbscript event=BeforeInitialBind(varParams) for=MSODSC>
<!--
Dim lngLastOrder
Dim strCookie

strCookie = document.cookie
If Len(strCookie) > 0 Then
    'Extract the date from the cookie
    lngLastOrder = Mid(strCookie, InStr(strCookie, "LastOrder=") + 10)
    If InStr(lngLastOrder, ";") Then
        'Remove trailing cookie name-value pairs
        lngLastOrder = Left(lngLastOrder, InStr(lngLastOrder, ";") - 1)
    End If
    MSODSC.RecordsetDefs("qryOrdersPage").ServerFilter = "OrderID >= " &
lngLastOrder
Else
    MSODSC.RecordsetDefs("qryOrdersPage").ServerFilter = "OrderID >= 11050"
End If
-->
</SCRIPT>

```

→ To review the syntax of the Mid and InStr functions, which is the same in VBA and VBScript, see “Text-Manipulation Functions,” p. 373.



For an alternative and more complex VBScript example for passing query parameters between pages, search for “Passing Parameters to a Data Access Page” (include the quotes) at <http://msdn.microsoft.com>.



The OrderStatusPages.htm, CurrentOrders.htm, and CurrentOrdersCookie.htm files are in the \Seual1\Chaptr25 folder of the accompanying CD-ROM. To test passing a parameter with a cookie, open OrderStatusPages.htm in IE and then click the Display Latest Orders... link.

#### TIP

Don't attempt to pass a cookie value to another page in Page view; both pages must run in IE to share the cookie. The preceding code substitutes a constant OrderID value (11050) to prevent an error or downloading the entire Recordset.



If your navigation page's link opens an empty page whose data source is a parameterized query, see the "Parameter Naming and Cookie Parsing Problems" topic of the "Troubleshooting" section near the end of the chapter.

## SAVING FORMS AS DAP

The ReportML transform isn't named FormML for a good reason—saving forms as DAP has severe form design limitations. Most forms have subforms, tab controls, or other objects that ReportML ignores. For example, saving Northwind.mdb's sample Employees form as a page results in colorful—but empty—Caption: Employees and Header: Employees sections. Saving the Customer Phone List form creates a page with elements for the form header, detail, and footer section. The Caption section has an extra line of field-name captions, and the 26 command buttons to filter the records are missing from the option group below the navigation control (see Figure 25.15). Loss of the command buttons isn't a major issue because the form relies on Access macros to filter the detail records.

**Figure 25.15** Saving to DAP forms that generate simple lists usually works, but you must tweak the page design in most cases.

Company Name:	Contact:	Phone:	Fax:
Customer Phone List			
Company Name:	Contact:	Phone:	Fax:
Alfreds Futterkiste	Maria Anders	030-0074321	030-0076545
Ana Trujillo Emparedados y helados	Ana Trujillo	(5) 555-4729	(5) 555-3745
Antonio Moreno Taquería	Antonio Moreno	(5) 555-3932	
Around the Horn	Thomas Hardy	(171) 555-7788	(171) 555-6750
Berglunds snabbköp	Christina Berglund	0921-12 34 65	0921-12 34 67
Blauer See Delikatessen	Hanna Moos	0621-08460	0621-08924
Blondel père et fils	Frédérique Citeaux	88.60.15.31	88.60.15.32
Bólido Comidas preparadas	Martin Sommer	(91) 555 22 82	(91) 555 91 99
Bon app'	Laurence Lebihan	91.24.45.40	91.24.45.41
Bottom-Dollar Markets	Elizabeth Lincoln	(604) 555-4729	(604) 555-3745
B's Beverages	Victoria Ashworth	(171) 555-1212	

Saving a form (or a report) that has a Class Module saves the VBA code as a comment block between BEGIN VBA CODE and END VBA CODE comments (see Figure 25.16). The purpose of saving the code in the script is to aid you when rewriting the code in VBScript, if possible.

**NOTE**

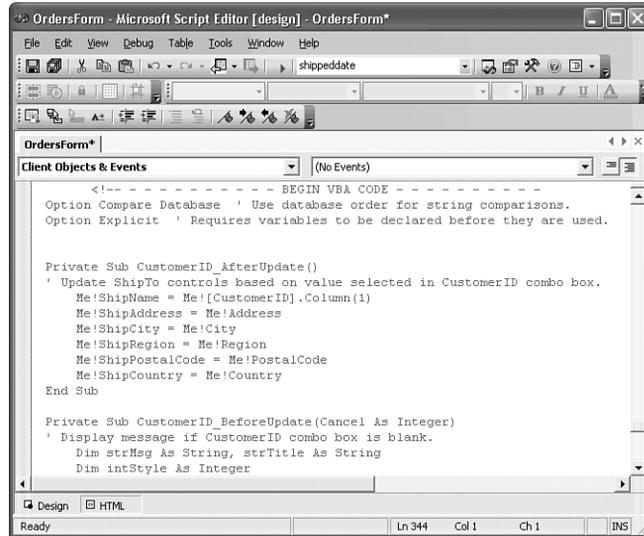


Access 2003 doesn't let you save unbound forms as DAP. Access 2002 could save unbound forms, such as Main Switchboard, to non-functional replicas.

Following are a few types of forms that you can save as DAP with varying degrees of success:

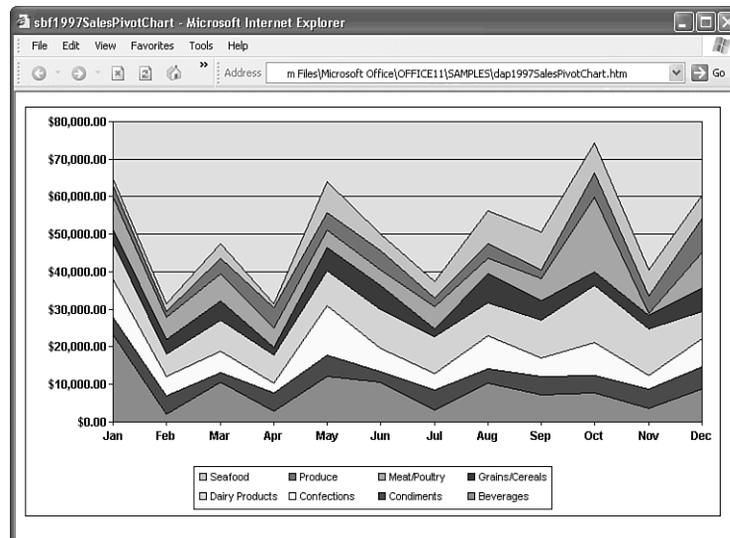
1. Simple forms without subforms, such as the Customer Phone List. Bound combo boxes used for lookup and other operations are empty. Bound subforms display all records from the underlying table or query.

**Figure 25.16**  
The Script Editor displays the VBA code that you can use as a reference when attempting to emulate the form's actions with VBScript.



- Subforms that contain bound PivotTable or PivotChart objects. For example, the sbf1997SalesPivotChart subform that you created in Chapter 18, “Adding Graphs, PivotCharts, and PivotTables,” retains all its properties when saved as a page. You must specify the Default View property as PivotTable or PivotChart to create the correct version of the page. Replace the missing caption supplied by the parent form with a title (see Figure 25.17).

**Figure 25.17**  
PivotChart views of forms or subforms save correctly as pages. A page saved from Chapter 18's sbf1997SalesPCLinked graph displays eight lines instead of the single line specified by the link to the frm1997SalesPCLinked form.



3. Bound pop-up forms and dialogs that don't contain unsupported controls. Pop-up forms and dialogs are better displayed as Web Page Dialogs—such as the Employees.htm page opened by the Review Orders.htm page.

## SAVING REPORTS AS DAP

RPT2DAP.xsl does a better job saving simple reports than saving forms as pages. Grouping levels in reports transform to `GroupLevelx` page sections with a navigation control; detail sections also gain a navigation control. Grouped sections are collapsed by default.

Following is a list of the most commonly used features of reports that don't survive the Save As process:

- Subreports—like subforms—are ignored. You must modify the resulting page—such as by adding an embedded inline frame containing a page—to emulate subreports.
- VBA code in the report's Class Module appears in the script as a comment block.
- You can't group by expressions that aren't supported by the Group On property of the report's group. The Alphabetic List of Products report handles the report's `=Left([ProductName],1)` grouping expression correctly because the report's Group On property value is set to Prefix Characters in the Sorting and Grouping dialog.
- Aggregate values in the outermost grouping section move below the navigation control in the navigation section. Text boxes lose their `ControlSource` property value when moved to this section.
- Aggregate value expressions in section footers don't work. You receive an error message if your report includes text boxes to display the result of aggregate expressions, such as `=Sum([FieldName])`.
- Expressions in the report's group headers and footers aren't supported.
- You can't add bound controls to a caption.
- The Can Grow and Can Shrink properties are ignored.

The restriction on subreports and loss of VBA code in Class Modules makes it difficult or impossible to save complex reports as useful DAP. Only 3 of the 11 sample Northwind reports—Alphabetical List of Products, Invoice, and Products By Category—save as readable pages.

## MODIFYING THE ALPHABETICAL LIST OF PRODUCTS PAGE

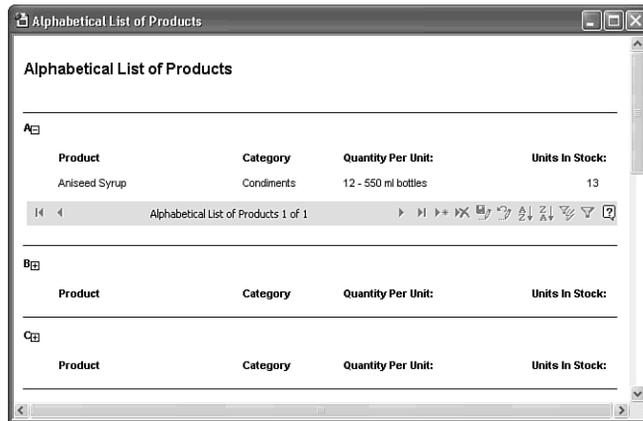
The Alphabetical List of Products report is simple enough for RPT2DAP.xsl to convert without modifying the report before the transformation. The resulting page, however, requires many modifications to improve its appearance and usefulness. The design changes that you make to this report are typical for many simple reports saved as DAP.

To save the Alphabetical List of Products to a page, do the following:

1. In the Database window's Reports page, right-click Alphabetical List of Products and choose Save As to open the Save As dialog.
2. Delete Copy Of in the text box, select Data Access Page in the As list, and click OK to open the New Data Access Page dialog.
3. Shorten the file name of the page to **ProductList.htm**, click Yes to enable scripts, and click OK to open Page view of the transformed report. Expand one or more of the sections to verify the presence of detail records (see Figure 25.18).

**Figure 25.18**

The Alphabetical List of Products report creates a page with a navigation control for each group. It's obvious that this page needs a makeover for viewing in a browser.



4. Change to Page Design view, and start redesign by dragging the empty text box below the title to the right of the title. Drag the caption labels from the Header: GroupLevel0 to the Caption: GroupLevel0 section to prevent repetition in every group level, and delete the colon suffix. Drag the line to under the captions, and drag the bottom of the Caption section to just below the line. Look ahead to Figure 25.19 for control positioning.

**TIP**

As you make changes to the page, switch to Web Page Preview to check your work. Opening the page in IE 6+ doesn't clear the undo stack when you save changes.

5. Navigation controls aren't appropriate unless the page has many rows in each section, which isn't the case with this page. Right-click in the Header: GroupLevel0 section, and clear the Record Navigation check box. Do the same for the Header: Alphabetical List of Products section.

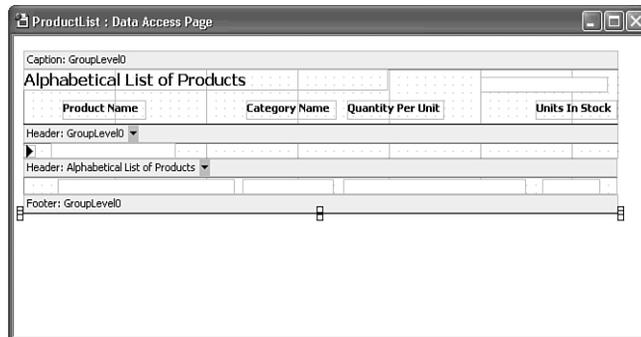
**TIP**

Use **Ctrl+ArrowKey** or **Ctrl+Shift+ArrowKey** to make fine adjustments to the position of the controls.

- 6. In the Header: GroupLevel0 section, move the text box to the right of the Expand control. Double-click the Expand control, click the Other tab, and change the Src property value to Black Arrow. Drag the bottom of the section to the bottom of the text box.
- 7. Drag the bottom of the Header: Alphabetical List of Products to the bottom of the text boxes.
- 8. In the Footer: GroupLevel0 section, move the line up a few pixels, delete the empty text box in the footer, and drag the bottom of the section up.
- 9. Optionally, change the font of the labels and text boxes to match the font used in other DAP—Tahoma, for the examples in this book. The design at this point appears as shown in Figure 25.19.

**Figure 25.19**

The page design changes shown here make the transformed report appear more like a conventional page.



- 10. To add a date to the empty text box in the Caption section, double-click the text box, click the Data tab of its properties window, and type **=Date()** in the DefaultValue text box. Right-align the text, add the bold, and remove the italic attribute.

**NOTE**

Setting the Format property value has no effect on the date display in Page view.

- 11. The width of the QuantityPerUnit field is larger than required, so you can add the Unit Price field to the left of the Units in Stock field. Decrease the width of the text box to two grid dots wider than the Quantity per Unit label. Open the Field List and the Toolbox, select the Bound Span control, and drag the UnitPrice field to the right of the shortened text box. Move the **Price** label to above the new field.

**TIP**

Use bound span controls instead of text boxes wherever possible to improve page performance. If you add a text box, you must change the format of the text box to prevent its borders from appearing on the page.

- Change to **Web Page Preview** to display the redesigned report in IE (see Figure 25.20).

**Figure 25.20**  
Page view of the redesigned page resembles the sample Review Products page but doesn't repeat captions.

Alphabetical List of Products					5/3/2003
Product Name	Category Name	Quantity Per Unit	Price	Units In Stock	
▼ A					
Aniseed Syrup	Condiments	12 - 550 ml bottles	\$10.00	13	
▼ B					
Boston Crab Meat	Seafood	24 - 4 oz tins	\$18.40	123	
▼ C					
Camembert Pierrot	Dairy Products	15 - 300 g rounds	\$34.00	19	
Carnarvon Tigers	Seafood	16 kg pkg.	\$62.50	42	
Chai	Beverages	10 boxes x 20 bags	\$18.00	39	
Chang	Beverages	24 - 12 oz bottles	\$19.00	17	
Chartreuse verte	Beverages	750 cc. per bottle	\$18.00	69	
Chef Anton's Cajun Seasoning	Condiments	48 - 6 oz jars	\$22.00	53	
Chocolade	Confections	10 pkgs.	\$12.75	15	
Côte de Blaye	Beverages	12 - 75 cl bottles	\$263.50	17	
▶ E					



The Alphabetical List of Products link to the ProductList.htm page is in the Data25.mdb file located in the \Seua11\Chaptr25 folder of the accompanying CD-ROM.

## USING A ServerFilter TO DISPLAY A SINGLE INVOICE PAGE

The Invoice report is an example of a report that you can save to a page without a major design change. An Invoice page also is a candidate for addition to the OrderStatusPages.htm page with the order number—the same as the invoice number—passed by a cookie.

### CREATING AND MODIFYING THE INVOICE PAGE DESIGN

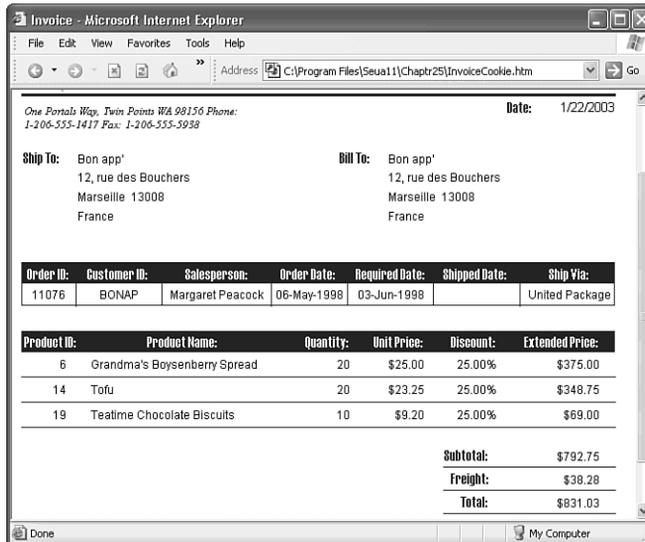
To transform the Invoice report to an Invoice.htm page and modify its design, do this:

- Right-click the Invoice report item in the Database window, choose **Save As**, and change the name to **Invoice**. Then select **Save As Data Access Page**, click **OK**, and click **Yes** to dismiss the warning message, if it appears, to save the page as Invoice.htm.
- Click the **Expand** control above the Ship To label to display the Order Details items. Verify that the Subtotal and Total amounts calculate correctly.
- Change to **Page Design** view, and double-click the DatePrinted text box. Click the **Data** tab of the properties window, and type **=Date()** as the temporary DefaultValue property value.

5. You don't need navigation controls for this page, so right-click Header: GroupLevel0 and clear the Record Navigation check box. Do the same for the Header: Invoices Filter.
6. Right-click Header: GroupLevel0 again and select Group Level Properties. Double-click the ExpandByDefault property to change its value from False to True. Delete the Expand control under the Header: GroupLevel0. Save your changes.
7. Open the page in Web Page Preview to verify the design changes in IE (see Figure 25.21) and then close IE.

**Figure 25.21**

The modified Invoice page has the Expand and record navigation controls removed and the missing Date value added.



### ADDING THE VBSCRIPT CODE TO ACCEPT THE INVOICE NUMBER COOKIE

If you created the OrderStatusPages.htm navigation page in the earlier “Passing Filter Criteria from One Page to Another with Cookies” section, you can add a line to the page to navigate to a specific invoice. This section demonstrates how to reuse VBScript code that you added to the CurrentOrdersCookie.htm page in the Invoices.htm page.

To add the VBScript required to accept a cookie value from the OrderStatusPages.htm page, do the following:



1. In the Database window, select the Invoice page and save it as **Invoice (Cookie)** with **InvoiceCookie.htm** as the page file name.
2. Select the Current Orders (Cookie) item and click the Microsoft Script Editor button to open the Script Editor for the CurrentOrdersCookie.htm page.
3. Scroll to and select the `<SCRIPT>...</SCRIPT>` block for the `BeforeInitialBind` event, copy it to the Clipboard, close the Script Editor, and close the page.



4. In Page Design view of the InvoiceCookie.htm page, click the Microsoft Script Editor button, go to the same location in page (after the `<![endif]-->` tag, press Enter, and paste the script block.
5. Change all instances of `lngLastOrder` to **lngInvoiceNum**, + 10 to +11, two instances of `qryOrdersPage` to `Invoices` and, `>=` to `=` (see Figure 25.22).

**Figure 25.22**  
Reusing VBScript code from another page saves time. Only minor modifications are required to `CurrentOrdersCookie.htm` code to display a single invoice.

```

Invoice (Cookie) - Microsoft Script Editor [design] - Invoice (Cookie)*
File Edit View Debug Table Tools Window Help
qryOrdersPage
Client Objects & Events (No Events)
</xml><![endif]-->
<SCRIPT language=vbscript event=BeforeInitialBind(info) for=MSODSC>
<!--
Dim lngInvoiceNum
Dim strCookie

strCookie = document.cookie
'Extract the invoice number from the cookie
If Len(strCookie) > 0 Then
lngInvoiceNum = Mid(strCookie, InStr(strCookie, "InvoiceNum=") + 11)
If Instr(lngInvoiceNum, ";") Then
'Remove trailing cookie name-value pairs
lngInvoiceNum = Left(lngInvoiceNum, Instr(lngInvoiceNum, ";") - 1)
End If
MSODSC.RecordsetDefs("Invoices").ServerFilter = "OrderID = " + lngInvoiceNum
Else
MSODSC.RecordsetDefs("Invoices").ServerFilter = "OrderID = 11077"
End If
-->
</SCRIPT>
Design HTML
Ready Ln 208 Col 1 Ch 1 INS
  
```

6. Save your script changes.

You can verify that the `ServerFilter` works by opening `InvoiceCookie.htm` in IE and verifying display of the default order 11077.

### CHANGING THE DATEPRINTED VALUE TO THE SHIPPING DATE

The Invoice report displays the system date in the `DatePrinted` span. Ordinarily, the invoice date is the date shipped, although some firms have been known to issue invoices for products not shipped or services not rendered. To comply with generally accepted accounting practices (GAAP), the invoice date should be blank if the goods haven't been shipped. Failure to use the shipping date as the invoice date is another Microsoft mystery.

The solution to this problem is to set the `innerText` property of the page's `DatePrinted` span to the `innerText` property of the `ShippedDate` text box. The VBScript code to make the change is simple and illustrates how to write event handlers for the Date Source Control. MSODSC has many more events than the ActiveX Data Object (ADO) Recordset; most of the events have equivalents in bound Access forms.

To make the change to the `ShippedDate` span, do this:



1. With Script Editor open with the `Invoice.htm` page, navigate to a position above or below the cookie script that you added in the preceding section. Add an empty line between the `</SCRIPT>` and `</BODY>` tags.

2. With the insertion point on the empty line, choose MSODC in the Object (Client Object & Events) list, open the Event list, and select DataPageComplete to add an event-handling stub.
3. MSODSC events require a parameter placeholder, even if the event doesn't return a value to the parameter. Add (**varParam**) to the event=DataPageComplete argument to make the event handler work.
4. Add a DatePrinted.value = ShippedDate.value statement. The complete event handler code is
 

```
<SCRIPT language=vbscript event=DataPageComplete(varParam) for=MSODSC>
<!--
    DatePrinted.value = ShippedDate.value
-->
</SCRIPT>
```
5. Save your changes, close the Script Editor, and close the page.

You can't test your addition to the script until you complete the next section.

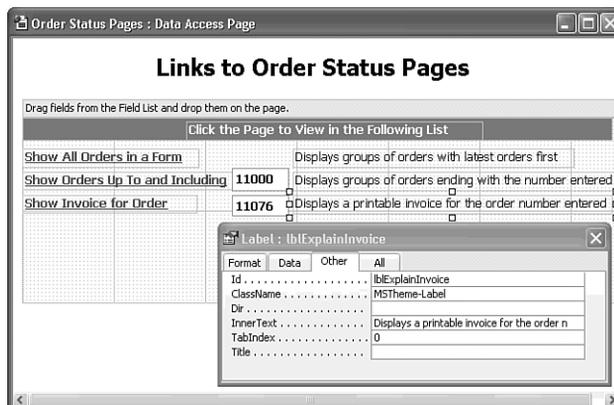
#### ADDING A LINK TO OPEN THE INVOICE PAGE

Adding a link for the Invoice.htm page to the OrderStatusPages.htm page is similar to adding the link to the CurrentOrdersCookie.htm page in the earlier "Passing Filter Criteria from One Page to Another with Cookies" section. You copy the existing link, text box, and label, change their Id values, and add a modified copy of the VBScript code for the new link's onClick event.

To add the link to the Invoice.htm page, do this:

1. Open Order Status Pages in Page Design view, and select the link, text box, and label. Press Ctrl+C and Ctrl+V to create a copy.
2. Change the text of the link and label, and reduce the width of the text box. Type a default order number in the text box (see Figure 25.23).

**Figure 25.23**  
Add links to other pages in the Order Status Pages page by copying and editing preceding links.



3. Double-click the link, click the Other tab of the properties window, and change the Id property value to **lnkInvoice**. Select the text box and change its Id to **txtInvoice**, and change the Id of the label to **lblExplainInvoice**, as shown in Figure 25.23.



4. Open the Script Editor, and copy and paste the VBScript block for lnkLatestOrders. Edit the event handler code as follows:

```
<SCRIPT language=vbscript event=onclick for=lnkInvoice>
<!--
    dim strURL
    dim lngInvoice
    const strParam1 = "InvoiceNum"

    strURL = "InvoiceCookie.htm"
    lngInvoice = txtInvoice.value
    document.cookie = strParam1 & "=" & lngInvoice
    window.navigate(strURL)
-->
</SCRIPT>
```

5. Save your changes, close the Script Editor, and open the page in Web Page Preview. Click the three links to test the code that you added in this section and the preceding section.



The Invoice link to the InvoiceCookie.htm page is in the Data25.mdb file located in the \Seua11\Chaptr25 folder of the accompanying CD-ROM.

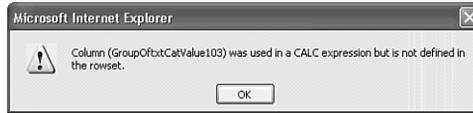
## SAVING THE INVENTORY BY CATEGORY REPORT TO A PAGE

The Inventory by Category report (rptInventoryByCategory) that you created in Chapter 16, “Working with Simple Reports and Mailing Labels,” is a candidate for conversion to a page. This report illustrates problems that you encounter with aggregate values in group footer sections—the mysterious error message shown in Figure 25.24 appears during the transformation process. It’s a common practice to use text boxes with aggregate functions in report group footers, so you’re likely to see this message often. Loosely translated, the error message means that RPT2DAP.xsl doesn’t translate =Sum([FieldName]) expressions into the required GroupOfFieldName function for section aggregates.

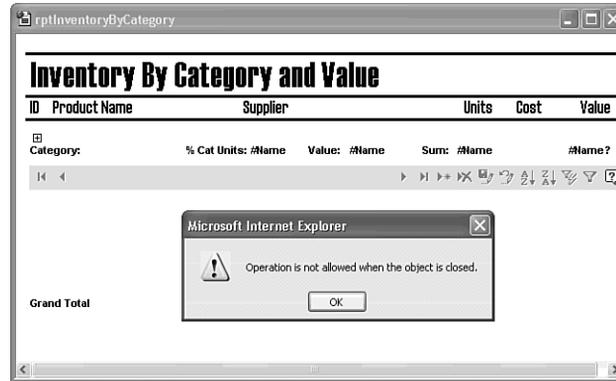
When you click OK to acknowledge the message, the page opens with #Name? as values of aggregate text boxes. Clicking the Expand button to display rows of the detail section opens an “Operation is not allowed when the object is closed” message (see Figure 25.25). The Category name is missing because pages don’t display lookup field values. You can’t put aggregates in the outermost footer, so there’s no Grand Total value. To create a page that you can modify to comply with page design rules, you must delete the text box expressions and then save the modified report as a page.

**Figure 25.24**

This error message appears when the report that you save as a page has an aggregate expression in a group footer.

**Figure 25.25**

After acknowledging the error shown in Figure 25.24, the page opens with errors and missing elements. Another error message appears when you click the Expand control.



## MODIFYING THE SOURCE QUERY TO SUPPLY REQUIRED FIELD VALUES

For this example, substituting CategoryName and supplier CompanyName values for lookup fields requires redesigning the source query—qryInventory, for this example. You must also supply fields to replace the calculated Cost and Value columns because the page section Value aggregates depend on field values. The once-simple qryInventory query requires a total makeover. Fortunately, the required changes to the query don't affect the design of the original rptInventoryByCategory report.

### Jet SQL

The SQL statement for the modified qryInventory query is:

```
SELECT Products.ProductID, Products.ProductName, Products.QuantityPerUnit,
Products.UnitPrice, CCur(0.667*[UnitPrice]) AS Cost, Products.UnitsInStock,
CCur([UnitsInStock]*[Cost]) AS [Value], Categories.CategoryID,
Categories.CategoryName, Suppliers.SupplierID, Suppliers.CompanyName
FROM Suppliers INNER JOIN (Categories
INNER JOIN Products
ON Categories.CategoryID = Products.CategoryID)
ON Suppliers.SupplierID = Products.SupplierID
WHERE (((Products.Discontinued)=False));
```

The modified qryInventory query is in the \Seua11\Chaptr25\Data25.mdb file.



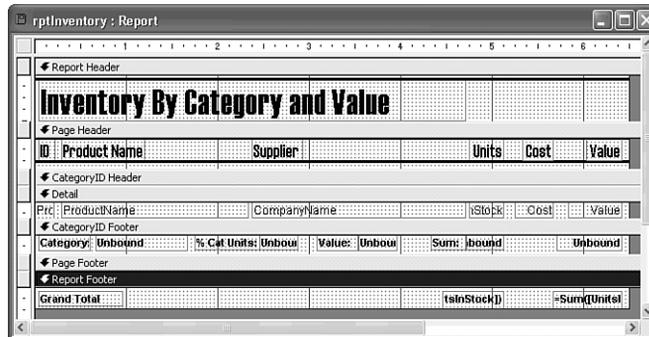
## CREATING AND MODIFYING THE SOURCE REPORT FOR THE PAGE

To permit saving the sample report as a page and to simplify the page redesign process, do the following:

1. Save a copy of rptInventoryByCategory as rptInventory.
2. Open rptInventory in Report Design view. In the Detail Section, right-click the SupplierID drop-down list and choose **C**hange To, **T**ext Box. Replace SupplierID with **C**ompanyName as the Control Source value.
3. Delete the expression in the Cost field, and type **C**ost as the field name. Do the same for the Value text box, but type **V**alue.
4. In the CategoryID footer section, change the CategoryID list to a text box, and delete the CategoryID field name. Delete the expressions in the remaining four text boxes in the CategoryID footer. All text boxes display Unbound as their value.
5. Delete the two text boxes in the Page Footer section, and move the Report Footer section up. Your redesigned source report appears as shown in Figure 25.26.

**Figure 25.26**

The changes shown here are required to eliminate errors when transforming the report to a page and to specify the columns to display from the modified source query.



6. Change to Print Preview to check your work. Close rptInventory and save your changes.
7. Right-click rptInventory in the Database window, choose **S**ave **A**s, and save the report as the Inventory page with Inventory.htm as the file name.
8. Click an Expand control to display the initial version of the Inventory.htm page (see Figure 25.27). Click OK to acknowledge “The DefaultSort property is invalid” message.

## REMOVING THE NAVIGATION CONTROLS AND ADDING A CATEGORY CAPTION

The Inventory page has only eight categories and a few products per category, so you don’t need navigation controls on the form. If you retain the Header: GroupLevel0 section and the Expand controls for the group, instead of expanding all groups by default, a CategoryName caption adjacent to the Expand control aids readability.

**Figure 25.27**  
Deleting the expressions in the source report's text box and changing the data source of the text boxes results in this initial page design.

ID	Product Name	Supplier	Units	Cost	Value
1	Chai	Exotic Liquids	39	12.006	468.234
2	Chang	Exotic Liquids	17	12.673	215.441
34	Sasquatch Ale	Bigfoot Breweries	111	9.338	1036.518
35	Steeleye Stout	Bigfoot Breweries	20	12.006	240.12
38	Côte de Blaye	Aux joyeux ecclésiastiques	17	175.75	2987.826
39	Chartreuse verte	Aux joyeux ecclésiastiques	69	12.006	828.414
43	Iphoh Coffee	Leka Trading	17	30.682	521.594
67	Laughing Lumberjack Lager	Bigfoot Breweries	52	9.338	485.576
70	Outback Lager	Pavlova, Ltd.	15	10.005	150.075
75	Rhônebräu Klosterbier	Plutzer Lebensmittelgroßmärkte AG	125	5.1692	646.15

To make the initial design changes to the page, do this:

-  1. Change to Page Design view. Right-click Header: GroupLevel0, and clear the Record Navigation check box. Do the same for the Header: qryInventory section.
-  2. Double-click the Expand control, click the Other tab, and change the Src property value to Black Arrow.
-  3. Open the Toolbox, select the Bound Span control, and draw the control adjacent to the right of the Expand control. Delete the added label.
4. Double-click the control, click the Data tab, open the ControlSource list, and select CategoryName. The ControlSource property value changes to GroupOfCategoryName: CategoryName.
5. If you want to retain the Category label and CategoryID text box in the Footer: GroupLevel0 section, double-click the text box and set its ControlSource property to CategoryName. In this case, the property value becomes GroupOfCategoryName1: CategoryName.
6. If you change the ControlSource property value of the CategoryID text box or delete it, you must change the value of the DefaultSort property of the GroupLevel0 group properties. For this example, the original value is [GroupOfCategoryID98] ASC, but this field no longer exists in the Field List. Right-click the section, choose Group Level Properties, and set the DefaultSort property value to [GroupOfCategoryName] ASC.

#### TIP

Be sure to change the DefaultSort property when deleting or changing transformed text boxes on which group sorting depends. If you don't, you receive "The DefaultSort property is invalid" messages when changing to Page view. The DefaultSort property value must match a GroupOfFieldName item in the Field List.

7. The offending `DefaultSort` property value that raises the error on expanding a group is `DESC` with no field name for `GroupLevel: Query Inventory`. Change `DESC` to **[Value]** `DESC` to fix the problem.
8. If the font name for the new controls that you've added isn't the same as the existing controls, conform the fonts in other than the `Caption: GroupLevel0` section. Tahoma is the font used for most of the examples in this book.
9. Change to Page view to check your work so far (see Figure 25.28).

**Figure 25.28**  
Removing navigation controls, adding a `CategoryName` caption, and making other minor changes improve the usability of the form.

ID	Product Name	Supplier	Units	Cost	Value
▼ Beverages					
38	Côte de Blaye	Aux joyeux ecclésiastiques	39	12.01	\$468.23
34	Sasquatch Ale	Bigfoot Breweries	17	12.67	\$215.44
39	Chartreuse verte	Aux joyeux ecclésiastiques	111	9.34	\$1,036.52
76	Lakkalikööri	Karkki Oy	20	12.01	\$240.12
75	Rhönbräu Klosterbier	Plutzer Lebensmittelgroßmärkte AG	17	175.75	\$2,987.83
43	Iphoh Coffee	Leka Trading	69	12.01	\$828.41
67	Laughing Lumberjack Lager	Bigfoot Breweries	17	30.68	\$521.59
1	Chai	Exotic Liquids	52	9.34	\$485.58
35	Steeleye Stout	Bigfoot Breweries	15	10.01	\$150.08
2	Chang	Exotic Liquids	125	5.17	\$646.15
70	Outback Lager	Pavlova, Ltd.	57	12.01	\$684.34
<b>Category: Beverages</b>			<b>% Cat Units:</b>	<b>Value:</b>	<b>Sum:</b>
▶ Condiments					
<b>Category: Condiments</b>			<b>% Cat Units:</b>	<b>Value:</b>	<b>Sum:</b>
▶ Confections					
<b>Category: Confections</b>			<b>% Cat Units:</b>	<b>Value:</b>	<b>Sum:</b>
▶ Dairy Products					

## ADDING CATEGORY SUBTOTALS AND GRAND TOTALS TO THE PAGE

Adding subtotals and grand totals to pages differs considerably from the method that you use for reports. The AutoSum feature makes replacing the missing group subtotals and grand totals easy.

To replace the report's missing subtotals and grand totals, do the following:

1. Select the `UnitsInStock` text box in the Header: `qryInventory` section, and click AutoSum to add a new label and `SumOfUnitsInStock` text box in the Footer: `GroupLevel0` section. Delete the label and the original text box, and drag the `SumOfUnitsInStock` text box to the right of the `Sum:` label.
2. Repeat step 1 for the text box under the `Value` label. Move the `SumOfValue` text box to the right of the `Cat. Value` label.
3. Delete the `% Cat.` and `Value` label and text boxes in the Footer: `GroupLevel0` section. Programming calculated text box values is beyond the scope of this chapter. Drag the `Category` label and text box to the right.
4. Select the `SumOfUnitsInStock` text box, and click AutoSum to add a Footer: `qryInventory-SumOfUnitsInStock` section and a `SumOfSumOfUnitsInStock` text box. Delete the associated label. Align the left and right edges of the Footer section with the sections above it.

5. Repeat step 3 for the SumOfValue text box, and then adjust the positions of the text boxes under the Units and Value text boxes.
6. Select the Grand Total: label, and press Ctrl+X to cut it to the Clipboard. Delete the unnamed footer section where the Grand Total: label was located.
7. Select the Footer: qryInventory-SumOfUnitsInStock section, and press Ctrl+V to paste the label. Figure 25.29 shows the final design in Page Design view.

**Figure 25.29**

The final design of the Inventory.htm page includes units and value subtotals by category, as well as grand totals for all categories.

ID	Product Name	Supplier	Units	Cost	Value
Header: GroupLevel0					
Header: GroupOfCategoryName					
Header: qryInventory					
Footer: GroupLevel0					
Category:			Sum:	SumOf	SumOfVal
Footer: qryInventory-SumOfUnitsInStock					
Grand Totals:				SumOfSu	SumOfSum

8. Open the report in Web Page Preview to display the summary version of the page (see Figure 25.30).

**Figure 25.30**

Web Page Preview displays the eight subtotals and grand totals for units in stock and inventory value.

ID	Product Name	Supplier	Units	Cost	Value
Beverages					
1	Chai	Exotic Liquids	39	12.01	\$468.23
2	Chang	Exotic Liquids	17	12.67	\$215.44
34	Sasquatch Ale	Bigfoot Breweries	111	9.34	\$1,036.52
35	Steeleye Stout	Bigfoot Breweries	20	12.01	\$240.12
38	Côte de Blaye	Aux joyeux ecclésiastiques	17	175.75	\$2,987.83
39	Chartreuse verte	Aux joyeux ecclésiastiques	69	12.01	\$828.41
43	Ippoh Coffee	Leka Trading	17	30.68	\$521.59
67	Laughing Lumberjack Lager	Bigfoot Breweries	52	9.34	\$485.58
70	Outback Lager	Pavlova, Ltd.	15	10.01	\$150.08
75	Rhinbräu Klosterbier	Plutzer Lebensmittelgroßmärkte	125	5.17	\$646.15
76	Läkkaikööri	Karkki Oy	57	12.01	\$684.34
Category: Beverages			Sum:	539	Cat. Value: \$8,264.29
Condiments					
Category: Condiments			Sum:	507	Cat. Value: \$8,019.71



The Inventory link to the Inventory.htm page is in the Data25.mdb file located in the \Seua11\Chaptr25 folder of the accompanying CD-ROM.

## TROUBLESHOOTING

### DATA SOURCE PROBLEMS

*A “Microsoft Web Components could not open the database C:\Path\FileName.mdb” message appears when I try to open a page from another machine on the network.*

The most likely cause of this problem is that you forgot to change the `ConnectionString` property for the Jet 4.0 data source of the page from the default well-formed path to the UNC `\\ServerName\ShareName\FileName.mdb` format. On the design computer, open the page in Design view, right-click the page, and choose Page Properties. Click the Data tab, select the `ConnectionString` property, and press Shift+F2 to open the connection string in the Zoom box. Change the `DataSource=` element of the connection string to the UNC path to the share and file.

Another possibility is that the UNC connection string is invalid or the user doesn't have read or read/write permissions for the share. In this case, check the capability of the client to browse the `ShareName` folder named in the error message.

### PARAMETER NAMING AND COOKIE PARSING PROBLEMS

*When I click the link on the navigation page to open a page that requires a query parameter value, the Enter Parameters dialog opens.*

The name that you supplied as the first (parameter name) argument of the `MSODSC.RecordsetDefs("QueryName").parametervalues.Add` method statement doesn't match the name specified in the query design grid. If you're query expects a conventional input parameter, open the query in Design view, copy the `[ParameterName]` element to the Clipboard, and paste it into the Script Editor. (Don't forget to add the surrounding double-quotes.)

*When I click the link on the navigation page to open a page that requires a query or filter parameter value, the page opens with an empty Recordset.*

The most likely cause of this problem is an error in the length argument of the `mid` function used to parse the cookie. For example, if you substitute 9 or less for 10 in the `lngLastOrder = Mid(strCookie, InStr(strCookie, "LastOrder=") + 10)` statement, the parameter value passed to the query or filter begins with `=`, which is invalid for a date value. Count the number of characters in the name element of the cookie carefully to ensure that you're passing the correct value to the parameter.

If the preceding issue isn't the problem, temporarily add a message box (`MsgBox`) statement with the variable name as the `Prompt` argument. For example, to check what the preceding line of script returns, add a `MsgBox lngLastOrder` statement after the line. When you've diagnosed the problem, add an apostrophe (`'`) in front of `msgbox` to comment out the statement. It's also a good idea to add a temporary `MsgBox strCookie` statement to check the crumbs while debugging your cookie-parsing code.



## IN THE REAL WORLD—ENHANCING DATA ACCESS PAGE DESIGN

Much of the content of this chapter concentrates on the lack of support by DAP for common Access form and report designs and control objects. Observations about Save As DAP features that work as expected might be classified as damnation by faint praise. There are workarounds for unsupported objects—such as subforms and subreports—but most workarounds aren't elegant, require VBScript programming expertise, and often need additional HTML code. In most cases, what you see in Access isn't what you get in pages without a considerable amount of work on your part. Bear in mind, however, that creating a datasheet-style data entry Web page, for example, isn't a piece of cake in any authoring tool. This is especially true if you need to implement the equivalent of the Undo, Sort..., and Filter... buttons of the navigation control.



Third-party Access developers have the opportunity to improve on the basic functionality of the Save as DAP feature. For example, Michael Kaplan and Julianne Lee of Trigeminal Software, Inc. (TSI) developed a Form/Report to Data Access Page Wizard that overcomes many Save As DAP limitations. For example, the TSI Wizard handles most subforms and subreports, which open in an inline frame (<IFRAME>), and generates working drop-down lists (see Figure 25.31). The TSI Wizard is an Access add-in library (FrmRpt2Dap.mde) that you can download from <http://www.trigeminal.com/frmrpt2dap.asp>. Installation instructions are at [http://www.trigeminal.com/frmrpt2dap\\_readme.asp](http://www.trigeminal.com/frmrpt2dap_readme.asp). Data25.mdb in the \Seua11\Chaptr25 folder of the accompanying CD-ROM includes the Orders and Orders Subform pages generated by the TSI Wizard. The pages have minor design modifications, and the VBScript in the Orders.htm page has been altered to enable opening the subform directly from the Orders Subform.htm file.

**Figure 25.31**  
Trigeminal Software, Inc.'s Form/Report to Data Access Page Wizard populates combo boxes and displays subforms in an inline frame.

Product	Quantity	Unit Price	Extended Price
Devolio, Nancy		\$34.00	
Dodsworth, Anne		\$174.00	
Fuller, Andrew		\$3.00	
King, Robert		\$98.00	
Leverling, Janet		\$14.00	
Peacock, Margaret			
Suyama, Michael			
Queso Cabrales	1?		

Most of Microsoft's development investment in the upgrade from Office 2002 to Office 2003 is devoted to XML-related features. Microsoft's promotion of the .NET Framework for XML Web services has shifted the company's development focus to XML file generation manipulation by Office System 2003 members and Visual Studio .NET projects. Data access pages rely on HTML, which has a very limited repertoire of native control objects compared to conventional Windows applications. The capability to save complex Access objects to pages is, to a major extent, constrained by HTML 4.0 limitations. "If your only tool is HTML, everything looks like a <TABLE>, <TEXTAREA>, or <SPAN>" oversimplifies the issue but remains a valid aphorism.

Browser users are accustomed to crude Web-based data-entry forms and less-than-lightning response times, but they do expect attractive, consistent page graphics. The appearance of your pages is likely to be more important to executive and management types than their content or performance. Web page form over function priorities of nonusers is one of the reasons why this chapter and the preceding two chapters emphasize design changes to make pages more attractive. If your employer has a organization-wide page layout with standard logos, colors, fonts, and other design attributes, make sure that your DAP conform.