

Chapter 18

Outputting Data

The main purpose of most business applications is to collect data and produce information. The most common way of returning the information is in the form of reports. The data might also need to be inserted into a spreadsheet or other format, which would allow further analysis by the user to produce the required information. In this chapter, we'll take a look at how to accomplish all of this.

Visual FoxExpress allows developers to produce information via its output objects. Output objects can be used to develop reports or export data into several different file formats. Output objects are a very powerful component of VFE and are very easy to implement. VFE provides wizards to guide the developer through the process of creating report output and export output classes. This chapter will give you an introduction to creating and using the output objects in VFE.

In this chapter we assume that you already know how to manipulate report files using VFP's Report Designer; this includes grouping, subtotals and so forth. We also assume that you know how to create views and VFE cursor, data environment and business object classes.

It is worth noting here that the VFE report output class and the wizards included are not intended to be an end-user report creation tool.

Report output class

The Report Output Class wizard consists of 10 steps. After completing the 10 steps of the wizard, you will have a report output class and a VFP report file (FRX/FRT). The output object wizards only work with a single cursor; the actual output object classes can work with multiple tables or views, but you have to add the code to create the join conditions by hand in the `GetFrom()` method. The classes also support returning a UNION clause from the `GetUnion()` method, as well as other additional SQL options not exposed by the wizard. Even though these features are available in virtually all cases, the easiest way to create reports for a VFE application is to design a view that includes the data you want before creating the report. If the view is not a view for which you have already created cursor, data environment and business object classes, you will also need to create these classes before creating a report. Also, if you want to apply security to the report—only certain people can run it—you will need to add the appropriate security name to the Security Name table (see Chapter 17, "VFE Security").

We're going to use VFE's sample application to create a report. This process will show the 10 steps of the wizard and the resulting output. The report will print a list of products grouped by category and provide a count of the number of products per category. To do this, we first need to create a view to join the Products table and the Category table and select the desired fields. **Figure 1** shows the view in the View Designer. *Note that the view is sorted by Category Name, then Product Name.*

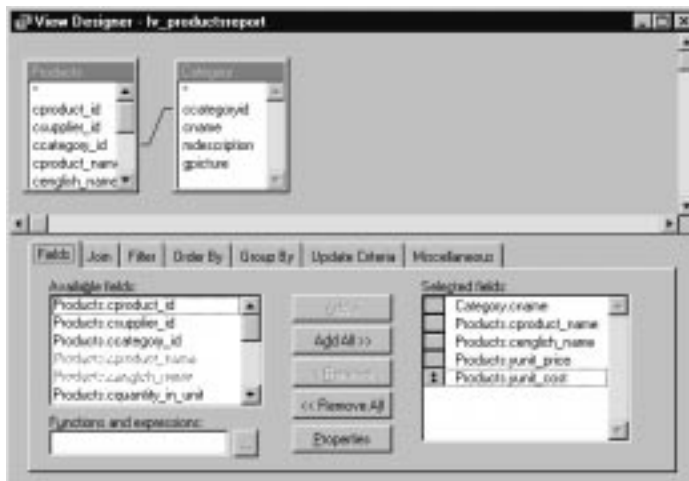


Figure 1. The view used to combine the Products and Category tables for the report.

If you are going to have grouping in your report, it is best to have the fields that are being grouped selected first in the view, and also to sort the view on those fields. After creating the view, create a cursor, data environment and business object based on the view. Now we are ready to create the report class.

In VFE's Application Builder, select the Classes tab then the Output Objects group. Then click on the New Object button. We are creating a report here, so select the Report Output Class wizard. This launches VFE's Report Output Class wizard, which will guide us through the 10 steps of creating the class. The first step is to select the business object the report is to use to get its data. This step is shown in **Figure 2**.

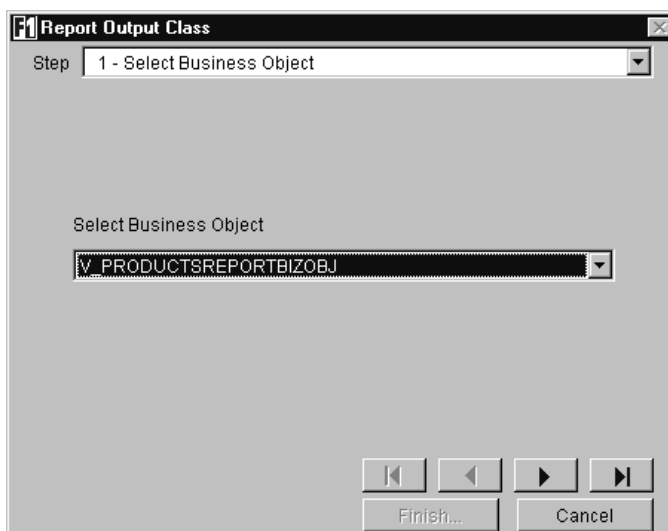


Figure 2. Step 1 of the Report Output Class wizard—selecting the business object.

Select the appropriate business object, and then click the Next button. The second step is to select the fields you want to include in the report. In this case, we want all of them. **Figure 3** shows this step.

Select all of the available fields so that they all appear in the Selected Fields list, and then click on the Next button. The third step is to specify the order for the report; this is shown in **Figure 4**.

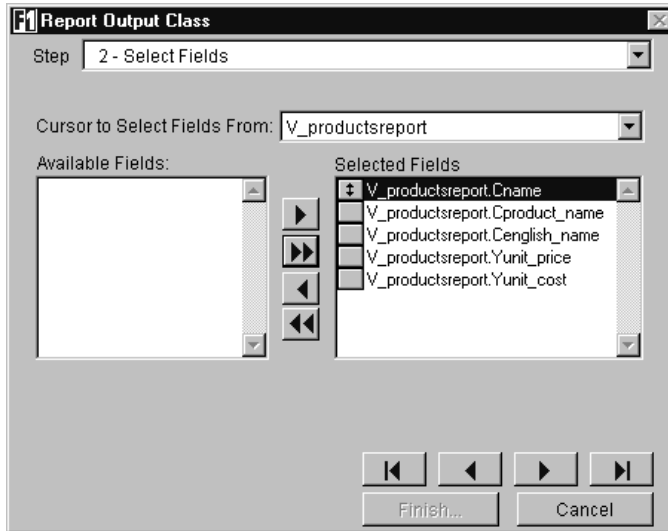


Figure 3. Step 2 of the Report Output Class wizard—selecting the fields.

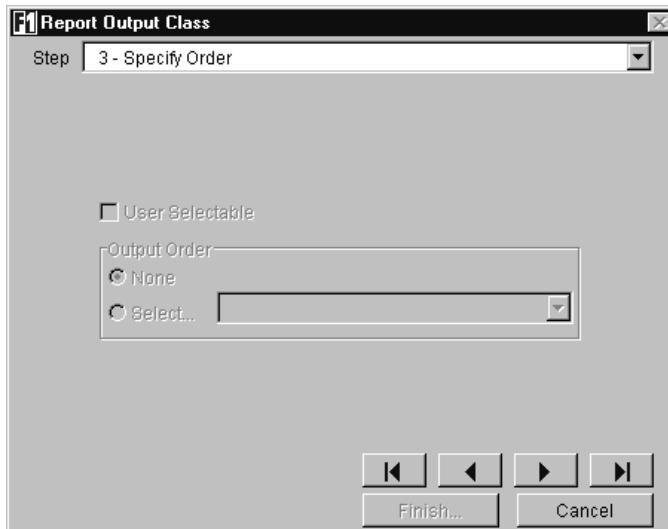


Figure 4. Step 3 of the Report Output Class wizard—specifying the order.

The controls on this step are not available because we have no indexes built on the view. If we had one or more indexes built, we would use this step to select the order and, using the User Selectable check box, specify whether the user will be allowed to select an order when they print the report. Be careful here. If you are grouping the report on a field or fields, the data must be sorted on those fields. If the user is allowed to change the sorting order on this type of report, it could give very different results!

Because we have no indexes in this view, click on the Next button. The fourth step in the wizard is to set your grouping options and summary options; this is shown in **Figure 5**.

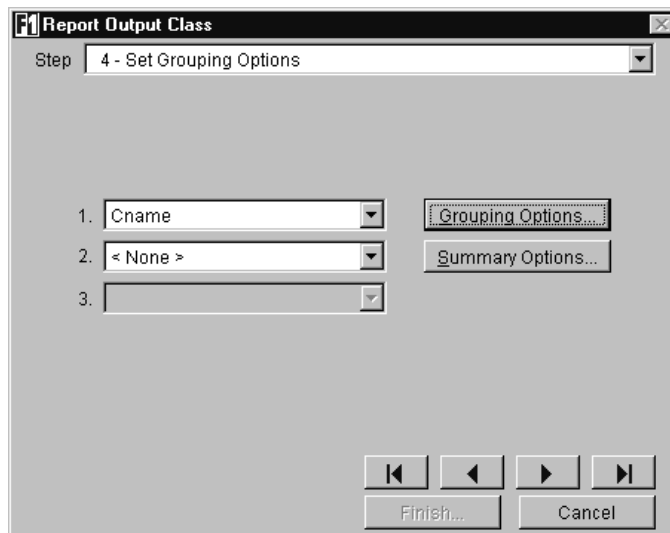


Figure 5. Step 4 of the Report Output Class wizard—setting grouping and summary options.

For our report, we want the products to be grouped by category name. In this step we need to select the cName field for the first field. This step of the wizard has two buttons: Grouping Options and Summary Options. The Grouping Options button allows us to specify the grouping intervals for the group. This tells the report how to break the records into groups. The options are to use the entire field value, the first character, the first two characters and so on, up to the first five characters of the field. If you choose the first two characters option, the report will start a new group anytime the characters in the first two positions of the field change. In our case we want to use the entire field. The Grouping Intervals dialog is shown in **Figure 6**.

Figure 7 shows the Summary Options dialog box. This dialog allows us to provide summary calculations for the items in the group. Because our report requirements want a count of products per category, we need to place a check mark in the Count column for one of the fields in the report. Because we are just counting and not summing or averaging, it does not make any difference which field is selected.

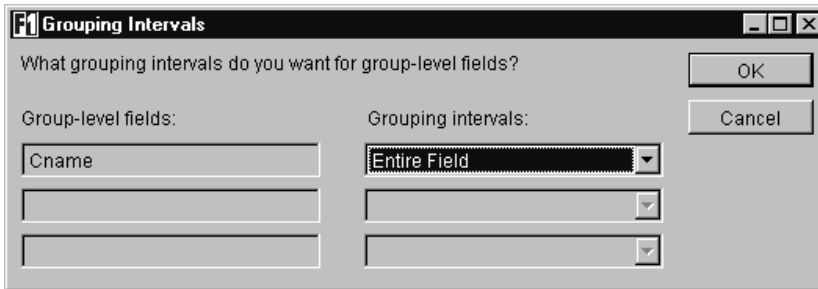


Figure 6. The Grouping Intervals dialog box.

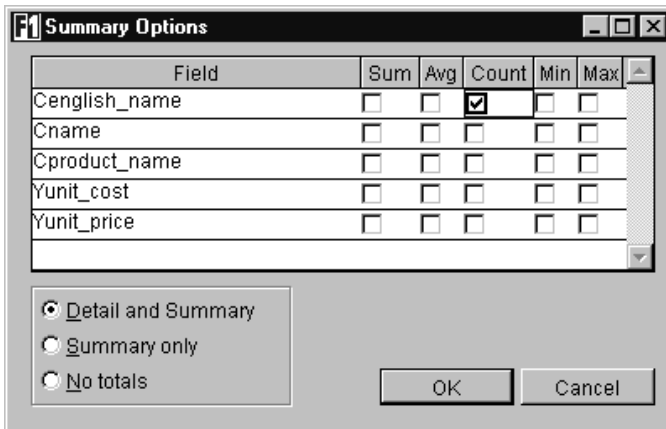


Figure 7. The Summary Options dialog box.

At the bottom of the Summary Options dialog box, we have more options to control the summary calculations. These are Detail and Summary, Summary only and No totals. The Detail and Summary option will cause the wizard to put fields in the group footer band and in the report footer band showing group totals and report totals. The Summary only option will put fields only in the report footer band to show report totals. The No totals option will not display the summary calculations on the report.

At this point you might be asking, “Why wouldn’t I want to display the summary calculations?” You may need the value to do some other calculations on the report but not necessarily need to display the value. The wizard will create the report variable that holds the values. You can then modify the report in the Report Designer to use the values as needed.

For our report we want Detail and Summary. Make sure it is selected, and then click the OK button. Click on the Next button to move on to Step 5 of the wizard.

This step, shown in **Figure 8**, allows you to set output options for the report. This includes setting the report title, the category and the query form, specifying search options, and assigning a security name to the report class.

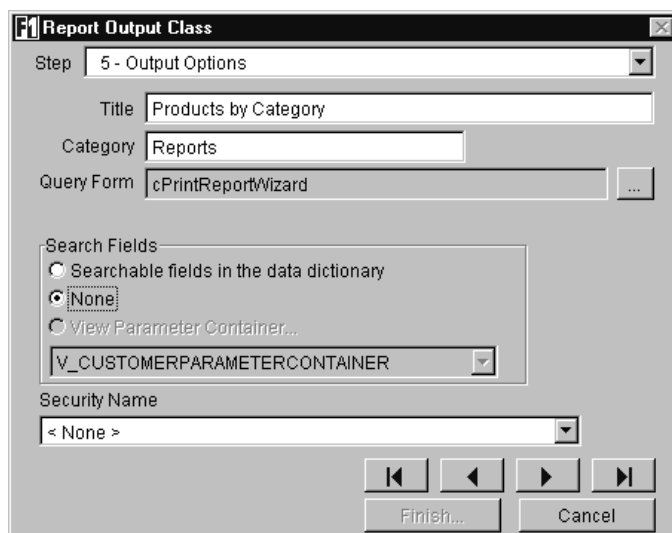


Figure 8. Step 5 of the Report Output Class wizard—setting the output options.

The first option to set here is the report title. This is used in two different places: It is placed on the report as the report title, and it is displayed in the report list in the application where the user selects the reports. This will be discussed in more detail in a later section. The title for our report is “Products by Category.”

The second text box is where you specify the category. The report list that is presented to the user for report selection uses the tree view control. The report titles are placed into groups or nodes in the tree view. This option allows you to specify in which category VFE should place this report title.

The next option allows you to indicate which query form you want to use to allow users to set report options at run time. You can create a custom class for this or use the one provided by VFE. In this case we are using the default wizard from VFE.

The fourth option allows you to specify how the user is going to select records to print. There are three choices here. The first is to allow the user to set filter conditions on any field that is marked as searchable in the data dictionary—the VFE wizard will display a form allowing the user to set the filter criteria if this option is selected. The second option is to prevent the user from setting any filter conditions. The third option is to use a view parameter container—this is used if the report is based on a parameterized view. VFE will display a view parameter container, created by you, to allow the user to enter the criteria for selecting records. For our report we do not want the user to set any filter criteria, so select None.

The last option is to select a security name. This allows you to control who has access to print the report. If the user is not assigned to a group that has the appropriate access to the security name assigned to the report, the report name will not be displayed in the tree view control for selecting reports. For this report we have no security, so leave this set to <None>.

After setting these options, click on the Next button to move on to Step 6 of the wizard, which is shown in **Figure 9**. This step lets us select which style we want our report to use. The wizard is going to use VFP’s Report Wizard to actually create the report files. VFP’s Report

Wizard has several styles to choose from; if you're not familiar with these styles, run VFP's Report Wizard and investigate the differences among them. For our report we are going to use the Executive style.

After selecting the desired report style, click on the Next button to move on to Step 7. This step allows you to select the report layout; again, this information is going to be passed on to VFP's wizard to be used in creating the report file. **Figure 10** shows this step.

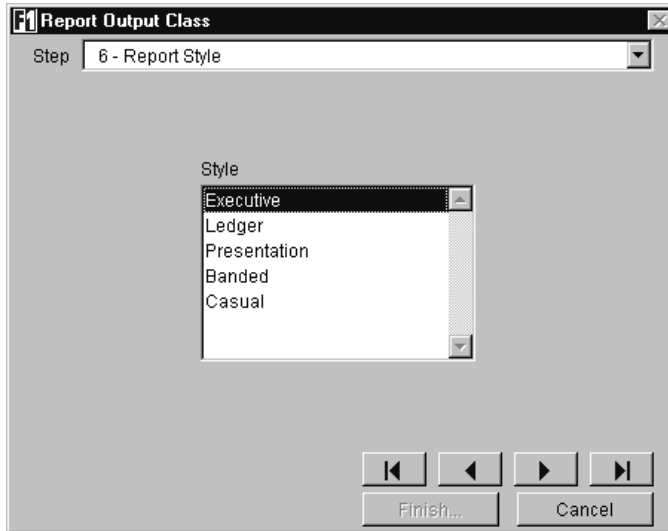


Figure 9. Step 6 of the Report Output Class wizard—selecting the report style.

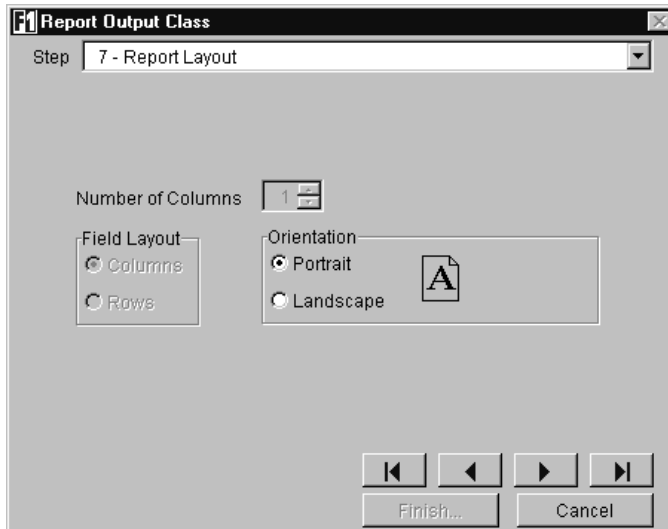


Figure 10. Step 7 of the Report Output Class wizard—selecting the report layout.

In this step we can select the number of columns, the field layout (columns or rows) and the orientation (portrait or landscape). For our report we are going to accept the defaults. Now click on the Next button to move on to Step 8.

This step, shown in **Figure 11**, allows you to specify where VFE should place the table that holds the report entries. This table is used to build the tree view that is displayed to the users for report selection.

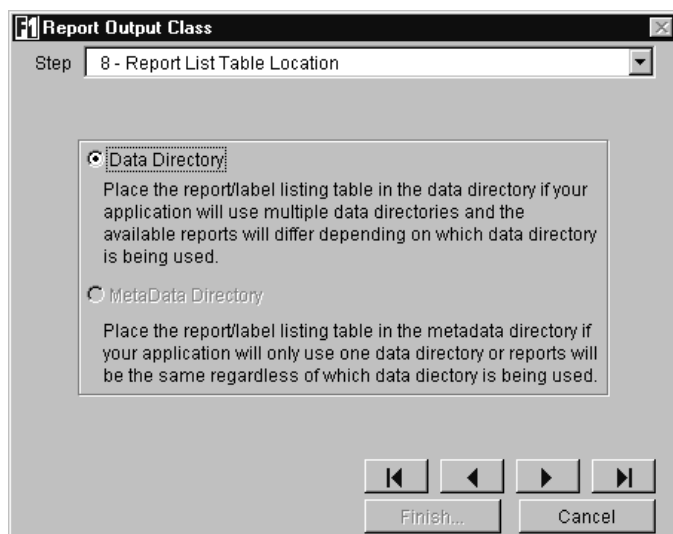


Figure 11. Step 8 of the Report Output Class wizard—selecting the report list table location.

Here our only choice is the data directory because the table already exists. If this were the first report, we would be able to choose the location. Click on the Next button to move on to Step 9. This step, shown in **Figure 12**, allows you to name the report file that the wizard is going to create. It also allows you to specify whether you want the report file to be excluded from the project. If you exclude the report from the project, you will need to distribute the report files. If you include the report, it will be compiled into the EXE or APP, so the report files will not need to be distributed.

Our preference is to exclude the report from the project. This way, if the user wants the report to be modified, you do not have to recompile the EXE or APP in order to send them the updated report. You can just modify the report and send them the updated FRX/FRT files.

After specifying the file name, click on the Next button to move on to the tenth, and last, step in the wizard. This step allows you to name the class and specify the visual class library where it should be stored; it's shown in **Figure 13**.

Select the default values here and click on the Finish button. Now we are ready to test our report.

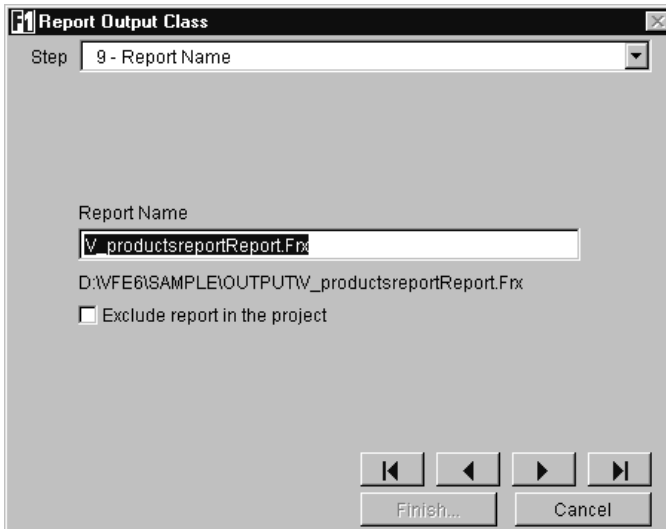


Figure 12. Step 9 of the Report Output Class wizard—selecting a report file name.

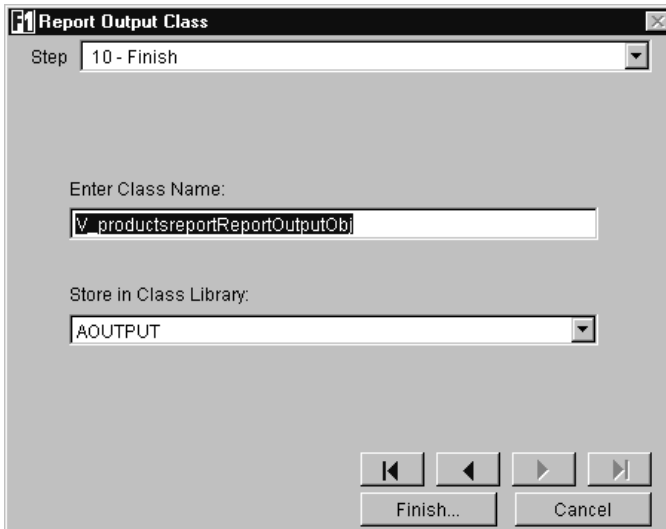


Figure 13. Step 10 of the Report Output Class wizard—specifying the class name and library.

Testing and modifying reports

To test our report and see what the wizard created, we need to run the application. After starting the app and entering the user name and password, go to the File menu and select Reports & Queries, or click on the Reports button on the application switchboard form. This

will display a form that contains the tree view control with all of the reports in the application. The form is shown in **Figure 14**.

Because we put our report title in the Reports group, click on the plus sign next to Reports to expand that node. You will then see our Products by Categories report; click on it, and then click on the Run button. VFE's end-user Report Wizard will display (see **Figure 15**).

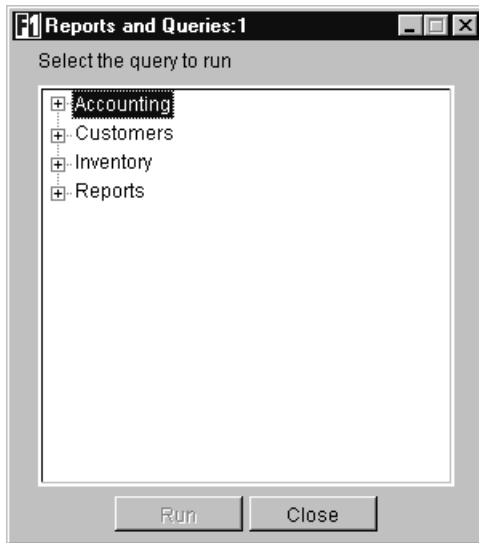


Figure 14. VFE's Reports and Queries selection form.

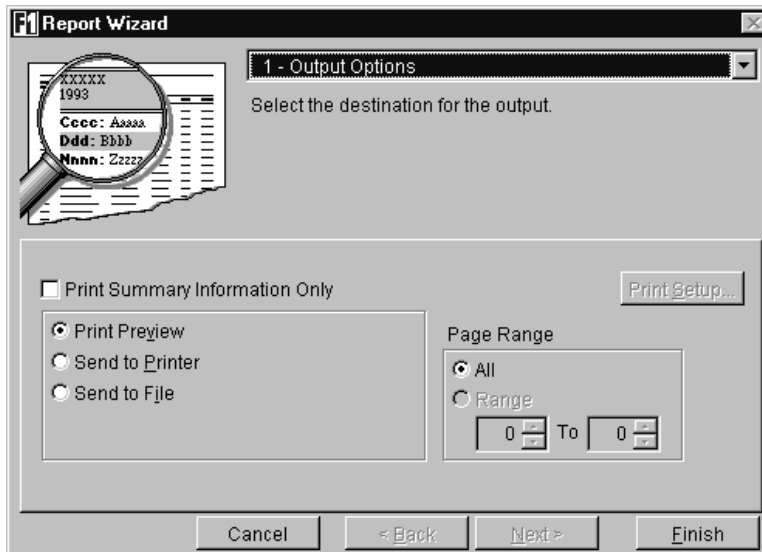


Figure 15. VFE's end-user Report Wizard.

The number of steps in the wizard will depend on what options were selected while running the 10-step Report Output Object wizard; in this case, there's only one step. Click on the Finish button to see our report—a partial view of the report is shown in **Figure 16**.

As you can see, the report needs a little polishing. We want the items and costs to appear all on one line. To accomplish this, exit the application and open the report file. Modify it in the Report Designer so it looks similar to **Figure 17**.

Name	Product Name	English Name	Unit Price	Unit Cost
Beverages				
	Blak			
	Tibetan Barley Beer		13.00	13.30
	Chateaux verte			
	Green Chateaux (Liquor)		18.00	12.00
	Chef Arto's Cajun Seasoning			
	Chef Arto's Cajun Seasoning		22.00	15.40
	Chicken Gumbo Mix			
	Chicken Gumbo Mix		520.00	379.00
	Côte de Blage			
	Côte de Blage (Red Bordeaux wine)		263.50	194.45

Figure 16. A partial view of the output from the Products by Category report.

Report Designer - v_productsreport.rpt

Products by Category

[DATE]

▲ Title

▲ Page Header

[ALLT(CNAME)]

▲ Group Header 1 CNAME

[CPRODUCT_NAME] [CENGLISH_NAME] [YUNIT_PRICE] [YUNIT_COST]

▲ Detail

[Count for ALLT(CNAME) CENGLISH]

▲ Group Footer 1 CNAME

[Page * + ALLT(CNAME) CENGLISH]

▲ Page Footer

[Total Count] [CENGLISH]

▲ Summary

Figure 17. The modified Products by Category report in VFP's Report Designer.

Save this report and run the application again. When you run the report again, the output should look similar to what is shown in **Figure 18**.

Products by Category				
06/24/00				
Name	Product Name	English Name	Sell Price	Sell Cost
Beverages				
	Brah	Tibetan barley Beer	19.00	13.30
	Charbrazo herbs	Brown Charbrazo (Liquor)	18.00	12.60
	Chef Andre's Cajun Seasoning	Chef Andre's Cajun Seasoning	22.00	15.40
	Chicken Gumbo Mix	Chicken Gumbo Mix	500.00	379.00
	Côte de Blaye	Côte de Blaye (Red Bordeaux wine)	263.50	184.45
	Guaraná Fantástico	Guaraná Fantástico Soft Drink	4.50	3.15
	Ipoh Coffee	Malaysian Coffee	45.00	32.20
	Lakshikori	Cloudberry Liquor	18.00	12.60
	Laughing Lumberjack Lager	Laughing Lumberjack Lager	14.00	9.00
	Mike's Cajun Spices	Mike's Cajun Spices	12.00	6.00
	Mike's Test Product	Mike's Test Product	1,000.00	1.00
	Outback Lager	Outback Lager	15.00	10.50
	Rhinokilo Kistebier	Rhinokilo Beer	7.75	5.62
	Sasquatch Ale	Sasquatch Ale	14.00	9.00
	Steeleye Stout	Steeleye Stout	18.00	12.60
	Tom's Secret Mushroom Sauce	Tom's Secret Mushroom Sauce	8.00	4.00
			Count for Beverages:	18
Condiments				
	Aniseed Syrup	Licorice Syrup	18.00	7.00

Figure 18. The output from the modified Products by Category report.



If you want to interactively preview your report with data while editing it in the Report Designer, you will need to do this outside of VFE's Application Manager. Issue the following commands in the command window:

```
set default to c:\vfe6\sample\data
oP = newobject('v_productsreportcursor', 'c:\vfe6\sample\libs\acursor.vcx')
oP.Open()
```

Then open the project file and edit the report. You can now preview while editing the report.

Export output class

The export output class allows you to create an output object that will export the data into one of many different file formats. With only seven steps, this wizard is a little shorter than the Report Output Class wizard. Once again, you should create a view that joins the tables and includes the fields containing the data you want to be exported. Then create VFE cursor, data environment and business objects based on the view. To demonstrate the export output class, we will use the same classes we created for the report output class.

Select the Classes tab on the Application Builder, click on the Output Classes node, and then click on the New Class button. This time we will select the Export Output Class option. This launches the Export Output Class wizard. The first step of the wizard is shown in **Figure 19**.

Select `v_productsreportbizobj` and click on the Next button. The second step, shown in **Figure 20**, allows you to select which fields to export.

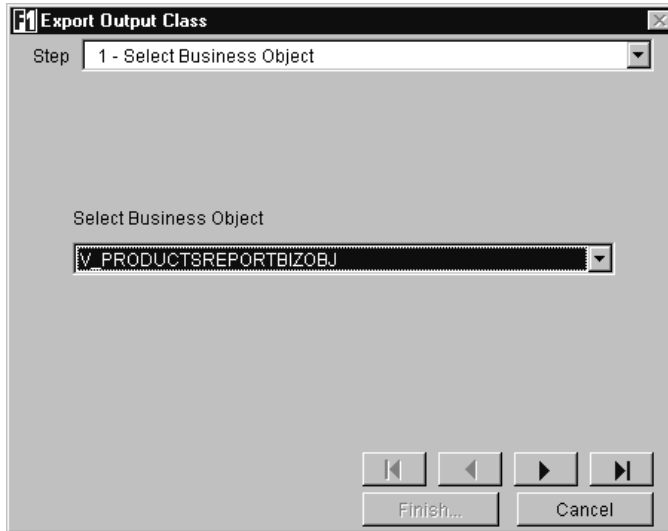


Figure 19. Step 1 of the Export Output Class wizard—selecting the business object.

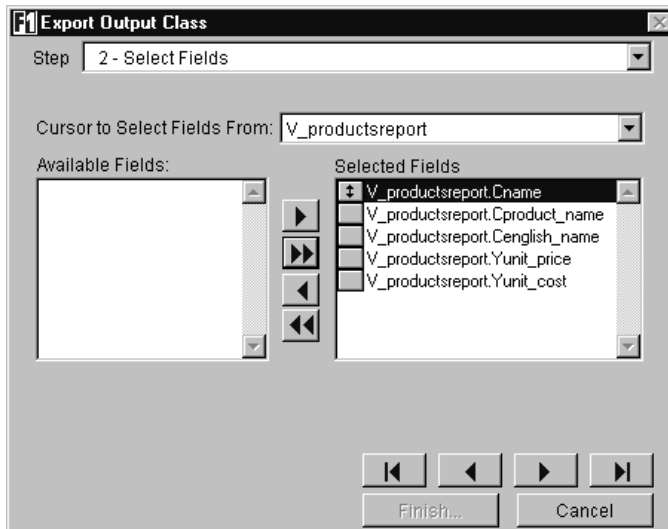


Figure 20. Step 2 of the Export Output Class wizard—selecting the fields to export.

For this export, we want all fields. Move all of the fields to the Selected Fields list, and then click on the Next button. The third step, shown in **Figure 21**, allows you to select how the records will be ordered. Because we have no indexes on our view, the options in this step are not available.

If we had one or more indexes created, we could select a default order and specify whether the user will be allowed to change the order when they actually export the data. We cannot change anything here, so click on the Next button.

The fourth step of the wizard allows us to set the output options; it's shown in **Figure 22**.

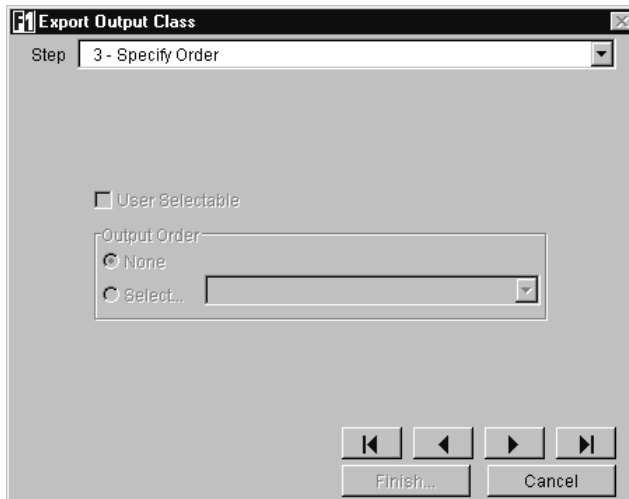


Figure 21. Step 3 of the Export Output Class wizard—specifying the order.

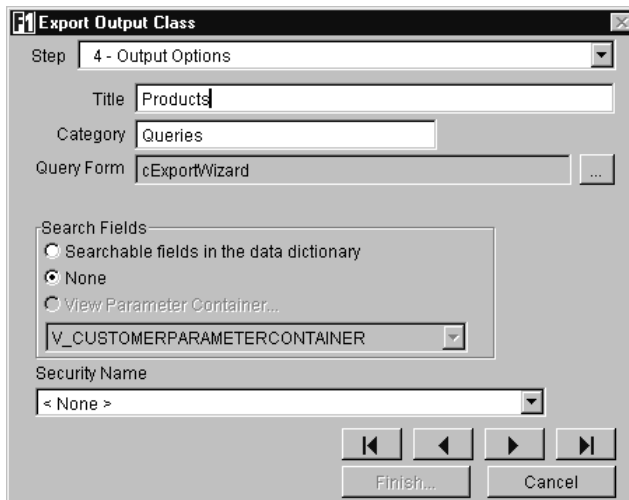


Figure 22. Step 4 of the Export Output Class wizard—setting output options.

The first option to set here is the title to be displayed in the Reports & Queries tree view control. The title for our query is “Products.”

The second text box is where you specify the category. The list that is presented to the user for report selection uses the tree view control. The query titles are placed into groups or nodes in the tree view. This option allows you to specify in which category VFE should place this query title.

The next option allows you to indicate which query form you want to use to allow users to set report options. You can create a custom class for this or use the one provided by VFE. In this case we are using the default wizard from VFE.

The fourth option allows you to specify how the user is going to select records to export. There are three choices here. The first is to allow the user to set filter conditions on any field that is marked as searchable in the data dictionary—the VFE wizard will display a form allowing the user to set the filter criteria if this option is selected. The second option is to prevent the user from setting any filter conditions. The third option is to use a view parameter container—this is used if the query is based on a parameterized view. VFE will display a view parameter container, created by you, to allow the user to enter the criteria for selecting records. For our export we do not want the user to set any filter criteria, so select None.

The last option is to select a security name. This allows you to control who has access to export the data. If the user is not assigned to a group that has the appropriate access to the security name assigned to this export class, the query name will not be displayed in the tree view control for selecting queries. For this export we have no security, so leave this set to <None>.

After setting these options, click on the Next button to move on to Step 5 of the wizard, which is displayed in **Figure 23**. This step allows us to select the type of file we want to create to hold the exported data.

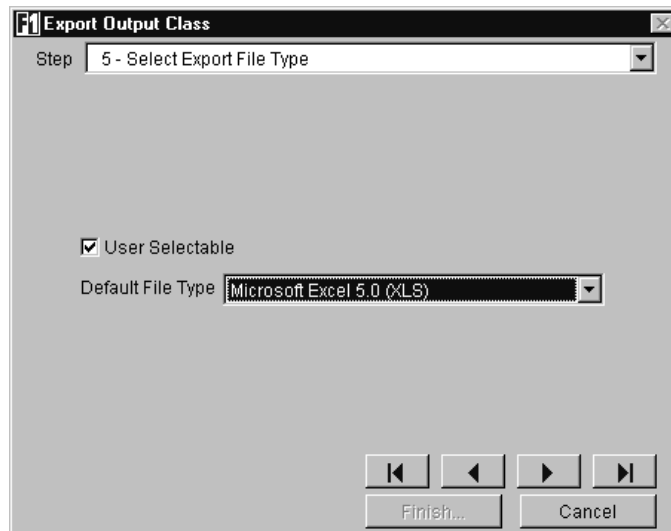


Figure 23. Step 5 of the Export Object Class wizard—selecting the export file type.

Here we want the default file type to be an Excel worksheet, but we will allow the user to select a different file type if necessary. Select Microsoft Excel 5.0 (XLS) from the combo box, and then click on the Next button to move on to Step 6 of the wizard

This step, shown in **Figure 24**, allows you to specify where VFE should place the table that holds the export/query entries. This table is used to build the tree view that is displayed to the users for report selection. This is the same table that is used for the report objects. We will discuss this table in more detail in a later section.

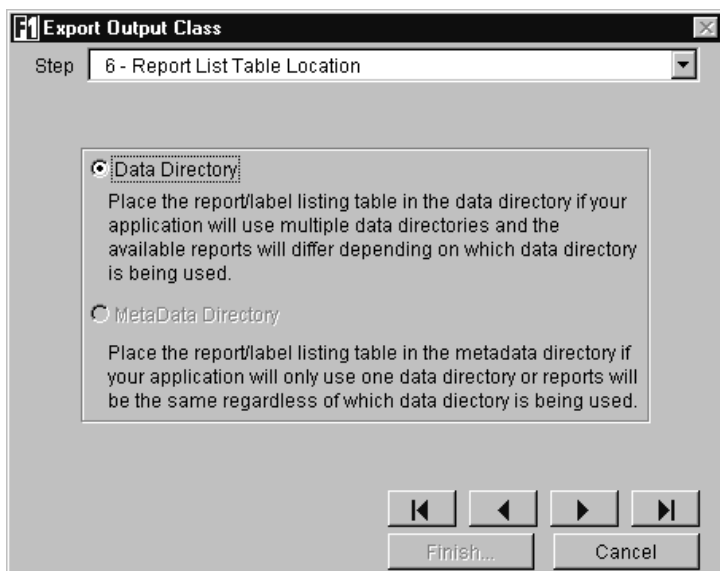


Figure 24. Step 6 of the Export Output Class wizard—selecting the report list table location.

Here our only choice is the data directory because the table already exists. If this were the first query/report object created, we would be able to choose the location. Click on the Next button to move on to the seventh, and final step. This step, shown in **Figure 25**, allows you to name the class and specify the visual class library where it should be stored. Accept the defaults here and click the Finish button.

After finishing the wizard, run the application. Go to the File menu and select Reports & Queries; you will notice a new node named Queries in the tree view. Click on the plus sign next to that node and you will see the Products option (see **Figure 26**).

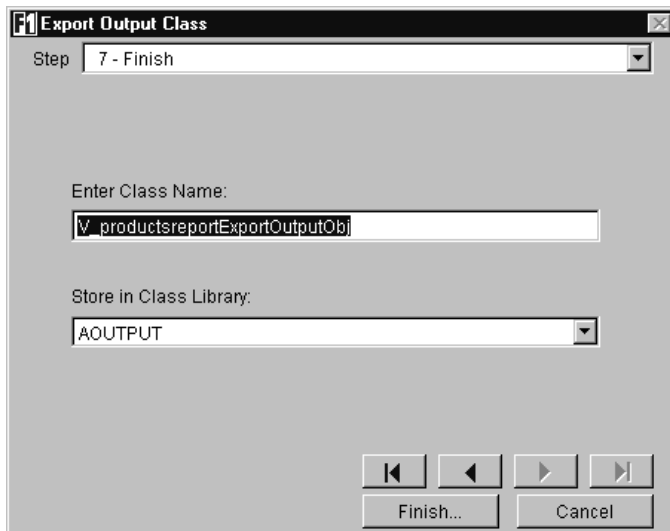


Figure 25. Step 7 of the Export Output Class wizard—specifying a class name and library.

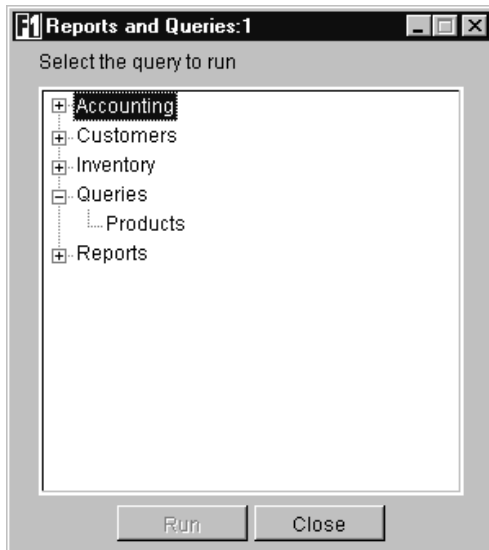


Figure 26. The Reports and Queries form showing our Export Class object in the tree view.

Select the Products entry and click on the Run button. VFE's end-user Export Query Wizard will run (see **Figure 27**).

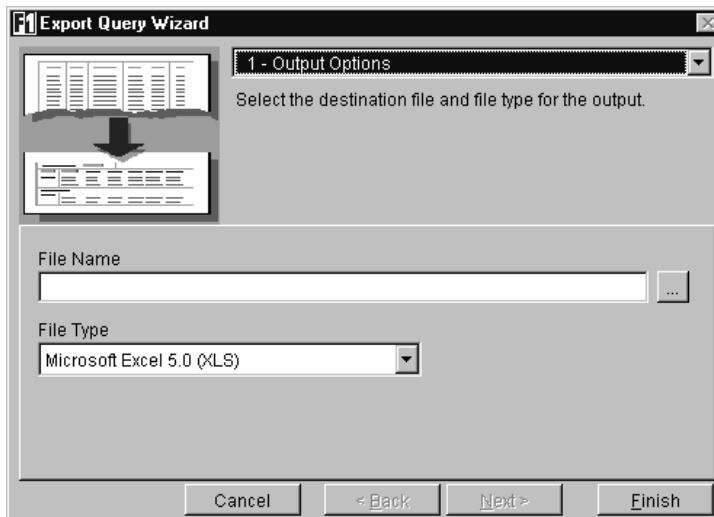


Figure 27. VFE's end-user Export Query Wizard.

The number of steps in this wizard will depend on the options that were selected while creating the export output class; in our case, there is only one step. The end user can select a file name and change the file type if they choose. To test this, enter a file name (including the path) in the appropriate text box and click on the Finish button. It is not necessary to include the XLS extension in the file name; the appropriate extension for the selected file type will automatically be added to the file name. **Figure 28** shows a partial listing of the Excel spreadsheet that was created.

The screenshot shows a Microsoft Excel spreadsheet titled 'products.xls'. The spreadsheet has five columns labeled A through E. The data is as follows:

A	B	C	D	E
name	product_name	english_name	unit_price	unit_cost
2	Beverages	Blah	Tibetan Barley Beer	19 13.3
3	Beverages	Chartreuse verte	Gros Chartreuse (Liquor)	18 12.6
4	Beverages	Chef Anton's Cajun Seasoning	Chef Anton's Cajun Seasoning	22 16.4
5	Beverages	Chicken Gumbo Mix	Chicken Gumbo Mix	500 379
6	Beverages	Côte de Blaye	Côte de Blaye (Red Bordeaux wine)	263.5 184.45
7	Beverages	Guaraná Fantástica	Guaraná Fantástica Soft Drink	4.5 3.15
8	Beverages	Ispah Coffee	Malaysian Coffee	46 32.2
9	Beverages	Lakkaikson	Cloudberry Liqueur	18 12.6
10	Beverages	Laughing Lumberjack Lager	Laughing Lumberjack Lager	14 9.8
11	Beverages	Mike's Cajun Spices	Mike's Cajun Spices	12 6
12	Beverages	Mike's Test Product	Mike's Test Product	1000 1
13	Beverages	Outback Lager	Outback Lager	15 10.5
14	Beverages	Rhônebräu Klosterbier	Rhônebräu Beer	7.75 5.425
15	Beverages	Sasquatch Ale	Sasquatch Ale	14 9.8
16	Beverages	Steeleye Stout	Steeleye Stout	18 12.6
17	Beverages	Toni's Secret Mushroom Sauce	Toni's Secret Mushroom Sauce	8 4
18	Condiments	Aniseed Syrup	Licorice Syrup	10 7
19	Condiments	Camembert Pierrot	Pierrot Camembert	34 23.8

Figure 28. A partial listing of the spreadsheet that was created with the export output class.

Controlling the tree view

The Reports and Queries selection form contains a tree view control that is populated with all of the reports and export/query objects created by the developer. The tree view control is populated from data that is stored in a table called REPOLIST.DBF. You have the opportunity to specify the location of this table when you create the first report or export object using VFE's wizards. Your choices are the data folder or the metadata folder.

If, after creating an output class, you decide to delete it, the entry in this table may not be removed. If the class is removed from the tree view control on the Classes tab of VFE's Application Builder, the entry will be deleted from the table. If the class is removed from the class library using VFP's Class Browser, the entry will not be deleted from the table. If this happens, you will have to open the table manually and delete the entry.

If you delete a report output class, the associated report files (FRX/FRT) will not be deleted from disk or removed from the project (if included). You will have to remove and delete these files as necessary.



If there are no records in the REPOLIST.DBF table or the REPOLIST table does not exist, the Reports & Queries option on the File menu will be disabled.

Working with the report and export object classes

The report and export object classes have many options that are not exposed by the wizards. Also, when a developer creates a report, clients often request additional information on the report. This section describes some of the key properties and methods of the cOutputObject class, which is the parent class for the cReportOutputObject and cExportOutputObject classes. Report output objects and export objects are based on these classes, respectively.

The output objects work in one of two ways, and you should be aware of the differences when you are creating reports. If you are basing your report directly off of tables, or if you are using views and allowing the user to search on fields, the output objects build a SQL statement and execute it to get the data for a report. If you are working with a view and don't allow the user to search on any fields, or if you use view parameters to allow the user to select the data, the output objects simply use and requery the views.

SetFieldList()

This method determines the list of fields that will be included in the query. When an output object wizard is run, VFE actually inserts code into this method. VFE adds a call to the output object's AddFieldToList() method for each field that you included in the report to this method. Normally the VFE wizards simply set properties, but because the Visual Class Designer limits the length of a property value, the developers at F1 used a method to accomplish this in order to not place a limit on the number of fields to include in a report. If you wish to add a field to an existing report, you need to add a line to the SetFieldList() method like the ones generated by the wizard for the new field. Having extra fields in the SetFieldList() method won't cause the report any problems, so sometimes it's easier to simply select all of the fields and then

remove the undesired fields from the report. Once the report has been finalized, extra fields should be removed from the `SetFieldList()` method for optimal performance.

IUseCurrentData

The `IUseCurrentData` property applies when the output object uses a view. The wizards set this property to indicate whether the output object should execute a SQL statement or simply use and requery a view. Although it is generally not good practice to execute a SQL statement against a view, sometimes this is a good option. If you're basing a report on a view, but you need to further control the data that's returned, you can set the `IUseCurrentData` property to `False` so that a SQL `SELECT` statement is executed and performed.

Executing reports from code

It is possible to create a report output object and call it from code. An example of this is attaching the code to a menu option. The code is very simple—all you need to do is create an instance of the report output object and then call the `Execute()` method. Here's an example:

```
Local loReport  
  
loReport = createobject('v_productsreportoutputobj')  
loReport.Execute()
```

Summary

The report and export object classes are very powerful features of a VFE application. We have just scratched the surface in this chapter. We encourage you to create a few of these classes and then examine the methods and properties of the classes that are created. There are several properties available that are not accessible via the wizards. There are also several methods that can be used to customize the classes. Have fun!