# Reporting System Enhancements in VFP 9

*Doug Hennig*
*Stonefield Software Inc.*
*1112 Winnipeg Street, Suite 200*
*Regina, SK Canada S4R 1J6*
*Voice: 306-586-3341*
*Fax: 306-586-5080*
*Email: dhennig@stonefield.com*
*Web: www.stonefield.com*
*Web: www.stonefieldquery.com*

## Overview

The major focus of VFP 9 is improving the reporting system. The list of new and improved features is enormous: multiple detail bands, protection of objects in the Report Designer, design-time events, ability to position objects absolutely, more zoom levels, better menus, etc. This document explores these enhancements in detail, demonstrates some types of reports you couldn't do in earlier versions of VFP, and provides an introduction to the "Extending the VFP 9 Reporting System" documents.

## Introduction

One of the biggest changes in VFP 9 is the incredible improvements made in the reporting system. There are three aspects to this: an enhanced Report Designer and other design-time improvements, enhanced run-time capabilities including output to HTML and XML, and new engine capabilities such as support for multiple detail bands. This document looks at each of these areas.

## Enhanced Report Designer

Microsoft had several goals in mind when working on the Report Designer:

- Protecting our investment in existing reports.
- Using an open architecture so it would be highly customizable.
- Improving the user interface.
- Providing new features: protection, design-time captions, absolute positioning, DataEnvironment handling, and international support.

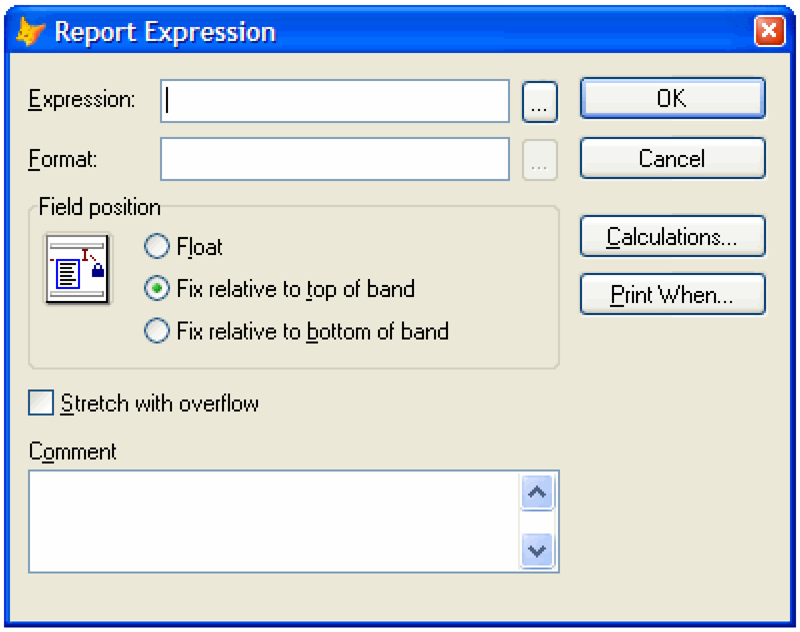
For the first point, the VFP team worked very hard to ensure that existing reports would work exactly the same in VFP 9 as they did in earlier versions. In addition, the FRX structure hasn't changed (although some previously-unused fields are now used in VFP 9 and there's now support for user-defined fields).

The keys to the second point are two things: a new system variable called _REPORTBUILDER and report events. The Report Designer can now call an Xbase application when events occur at design-time (such as adding a field or invoking a dialog). This application is specified in _REPORTBUILDER, which by default points to ReportBuilder.APP in the VFP home directory. You can use your own application if you wish by setting _REPORTBUILDER to point to a different application. This topic is discussed in detail in my "Extending the VFP 9 Reporting System, Part I: Design-Time" document.

Let's discuss the remaining two points.

### Improved User Interface

One of the first things you'll notice in the VFP 9 Report Designer is that the dialogs have been improved greatly. In VFP 8 and earlier, there were a lot of dialogs related to reporting: properties dialogs for each type of object and band, a Page Setup dialog for the report, Data Grouping and Variables dialogs, and so forth. Some of them had somewhat unusual interfaces, and some spawned yet other dialogs. You can see an example of the clunky interface when you double-click on a field object: that action brings up a properties dialog for the object, but that dialog doesn't allow you to change all properties of the object (font and color are missing, for example) and you have to click on buttons to bring up other modal dialogs to change some properties.



Expression:

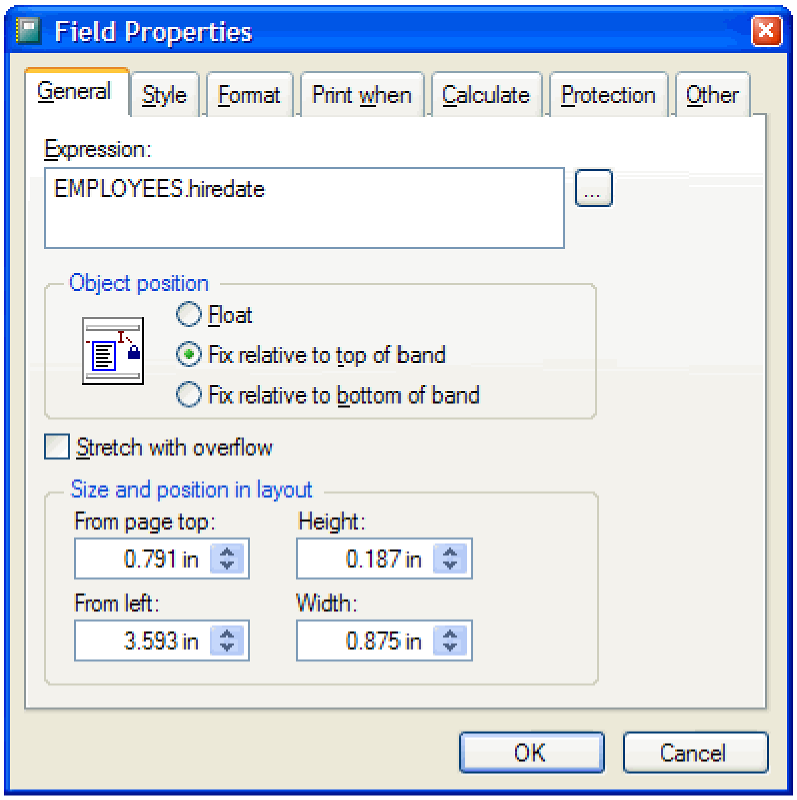
Format:

Field position
( ) Float
(•) Fix relative to top of band
( ) Fix relative to bottom of band

[ ] Stretch with overflow

Comment

In VFP 9, there are only a few dialogs, because all of the properties for an object are now in one place. For example, the Properties dialog for all object types uses a tabbed interface, so all possible properties for the object can be edited in one dialog and without launching additional modal dialogs.



Expression:
EMPLOYEES.hiredate

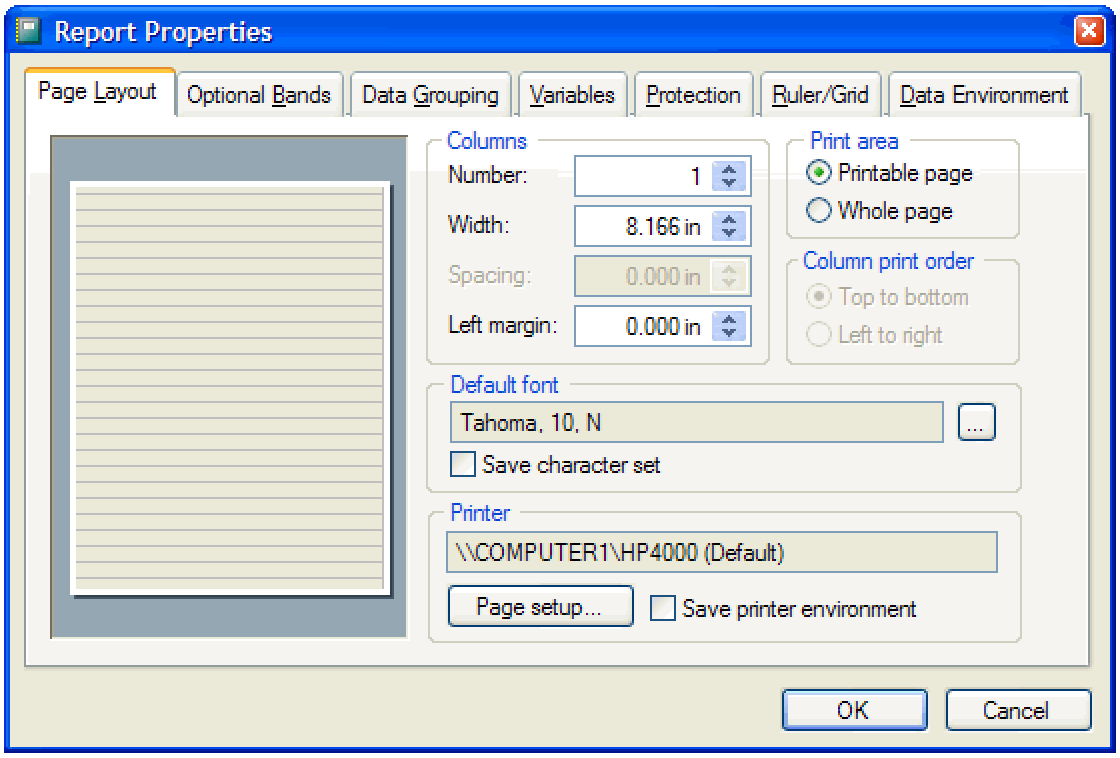
Object position
( ) Float
(•) Fix relative to top of band
( ) Fix relative to bottom of band

[ ] Stretch with overflow

Size and position in layout

| | |
|---|---|
| From page top: 0.791 in | Height: 0.187 in |
| From left: 3.593 in | Width: 0.875 in |

The new Report Properties dialog combines the features of the Page Setup, Title/Summary, Data Grouping, Variables, and Set Grid Scale dialogs, plus adds new features as well.



Columns
Number: 1
Width: 8.166 in
Spacing: 0.000 in
Left margin: 0.000 in

Print area
[✓ Printable page] Whole page

Column print order
[✓ Top to bottom] Left to right

Default font
Tahoma, 10, N
☐ Save character set

Printer
\\COMPUTER1\HP4000 (Default)
☐ Save printer environment

These dialogs aren't actually part of the product but are provided by ReportBuilder.APP. That's right, they're Xbase dialogs! Since the source code for ReportBuilder.APP comes with VFP (unzip XSource.ZIP in the Tools\XSource subdirectory of the VFP home directory, then look in the ReportBuilder folder), you could customize or subclass the various dialogs and behaviors to suit your needs.

Other UI improvements are:

- The Report menu and the shortcut menu for a report have been reorganized and have additional items (including Properties, which invokes the Report Properties dialog) to make it easier to work with reports.

- The mouse cursor now changes to provide a visual cue when an object can be resized.

- The Report Designer toolbar includes Page Setup and Font Properties buttons. Also, the View menu includes an item for the Report Designer toolbar. Previously, you had to choose Toolbars from the View menu and turn on the Report Designer toolbar in the resulting dialog.

- The Reports tab of the Tools, Options dialog has been reorganized and has a couple of new options: how the Expression Builder should deal with aliases for fields and whether the default run-time behavior is backward-compatible (the equivalent of using the new SET REPORTBEHAVIOR 80 command; see the "New Reporting

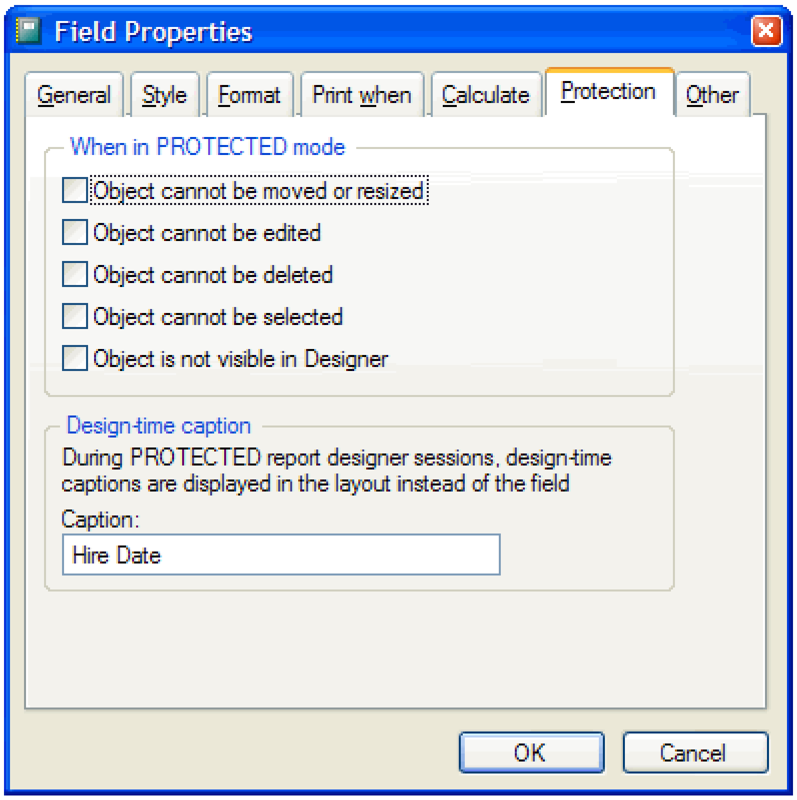
Syntax" section later in this document) or object-assisted (the same as SET REPORTBEHAVIOR 90). Run-time behavior is discussed later in this document.

## Protection

If you allow users to modify reports in the Report Designer in your run-time applications, you may have wished you could protect them from themselves. The Report Designer has a lot of features that can get an unsuspecting user in trouble, such as the DataEnvironment. If you only want the user to make simple changes, such as moving fields around or adding a company logo, all of the features available in the Report Designer are overkill.

VFP 9 has a new keyword for the MODIFY/CREATE REPORT/LABEL commands: PROTECTED. When this keyword is used, certain operations can be prevented. You have control over which actions a user can take at the object, band, and report levels using the Protection tab of the properties dialogs. For obvious reasons, this tab isn't available in protected mode.

You can control whether an object can be moved or resized, edited (that is, whether the Properties dialog can be invoked), or deleted. You can even control whether the object can be selected at all, or whether it appears or not.



When in PROTECTED mode

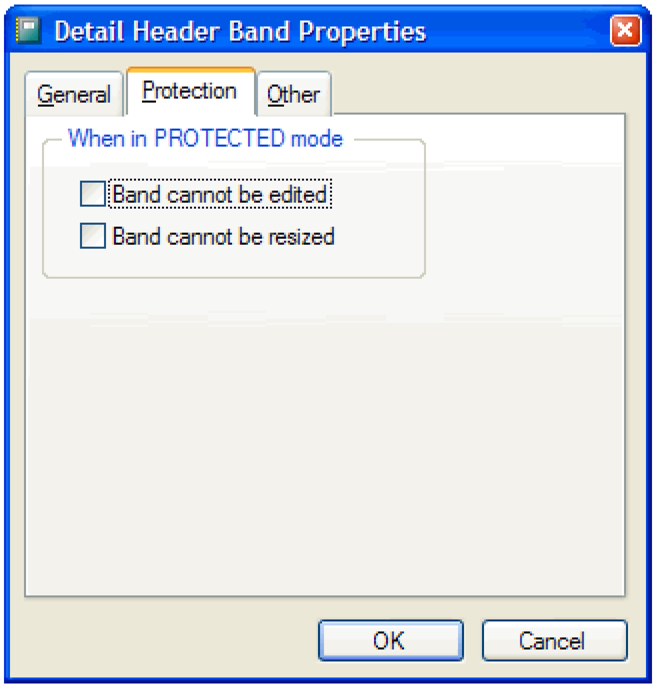
- [ ] Object cannot be moved or resized
- [ ] Object cannot be edited
- [ ] Object cannot be deleted
- [ ] Object cannot be selected
- [ ] Object is not visible in Designer

Design-time caption

During PROTECTED report designer sessions, design-time captions are displayed in the layout instead of the field

Caption: Hire Date

For bands, you can control whether they can be edited or resized.


Detail Header Band Properties

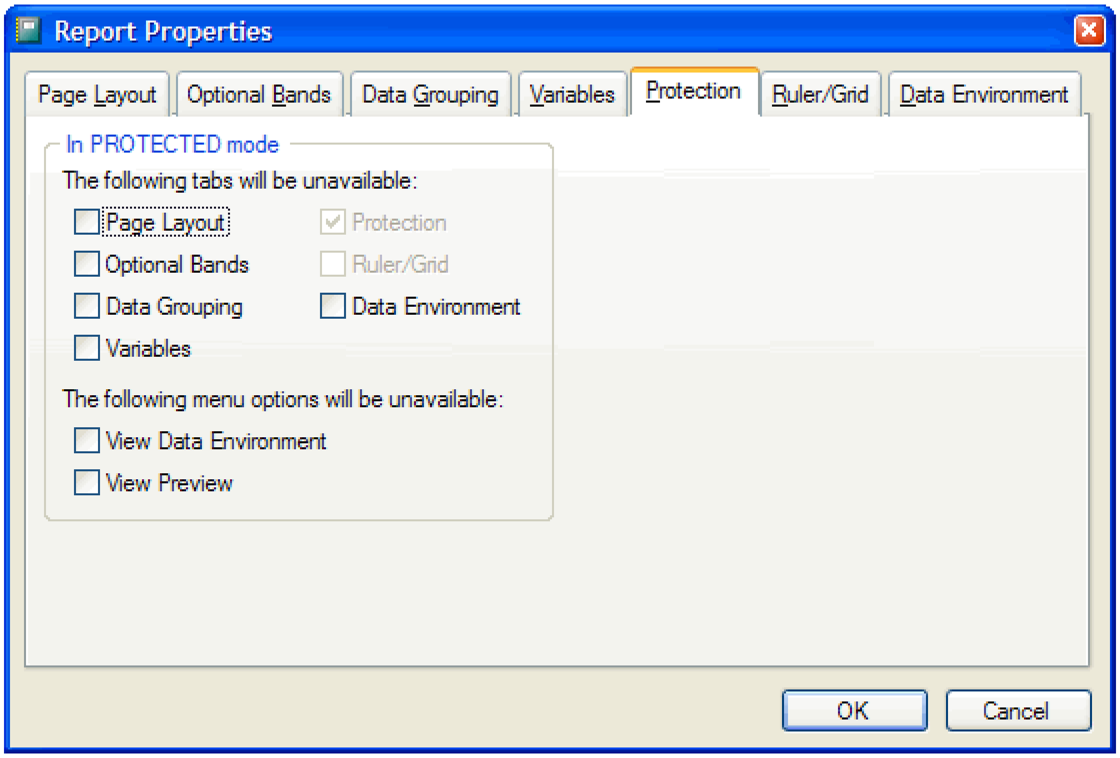
General | Protection | Other

When in PROTECTED mode

- [ ] Band cannot be edited
- [ ] Band cannot be resized

OK | Cancel

At the report level, you can control which tabs of the Report Properties dialog and which menu items are available.


Report Properties

Page Layout | Optional Bands | Data Grouping | Variables | Protection | Ruler/Grid | Data Environment

In PROTECTED mode

The following tabs will be unavailable:

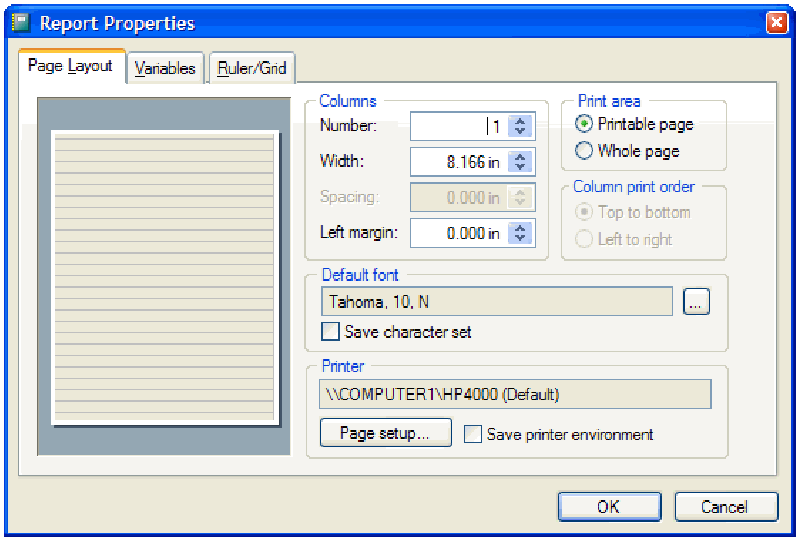
- [ ] Page Layout
- [ ] Optional Bands
- [ ] Data Grouping
- [ ] Variables
- [✓] Protection
- [ ] Ruler/Grid
- [ ] Data Environment

The following menu options will be unavailable:

- [ ] View Data Environment
- [ ] View Preview

OK | Cancel

For example, here's what the Report Properties dialog looks like when MODIFY REPORT … PROTECTED is used for a report with the Optional Bands, Data Grouping, and Data Environment tabs turned off:



Report Properties

Page Layout | Variables | Ruler/Grid

Columns
Number: 1
Width: 8.166 in
Spacing: 0.000 in
Left margin: 0.000 in

Print area
(●) Printable page
( ) Whole page

Column print order
(●) Top to bottom
( ) Left to right

Default font
Tahoma, 10, N
[ ] Save character set

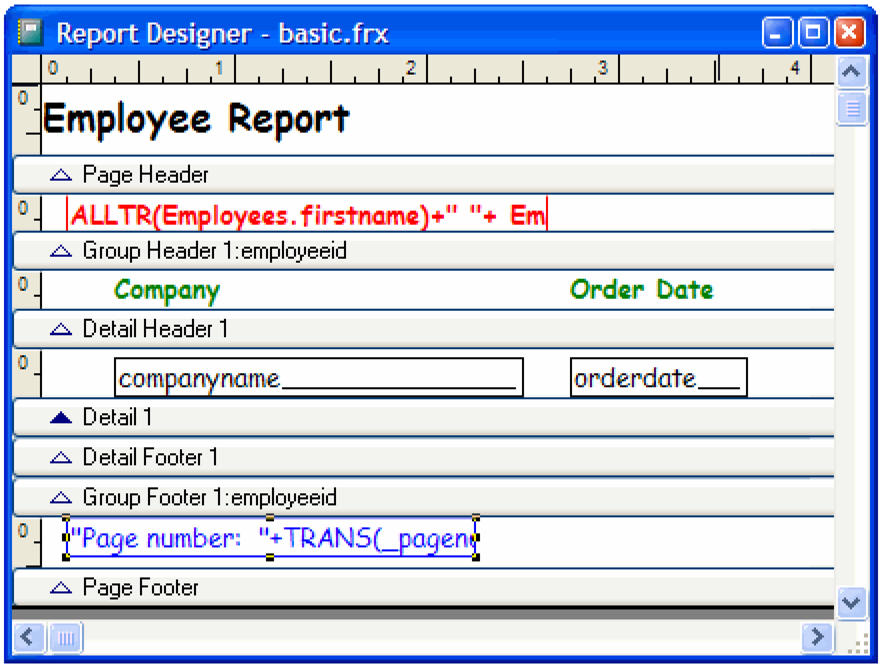
Printer
\\COMPUTER1\HP4000 (Default)
Page setup... [ ] Save printer environment

OK | Cancel

## Design-Time Captions

You may have noticed that the Protection tab for a field object has an additional setting: design-time caption. This allows you to indicate what appears in place of the field expression when a report is modified in protected mode.

For example, compare these two Report Designer sessions. This one used MODIFY REPORT without the PROTECTED keyword:



Report Designer - basic.frx

Employee Report

Page Header

ALLTR(Employees.firstname)+" "+ Em

Group Header 1:employeeid

Company | Order Date

Detail Header 1

companyname | orderdate

Detail 1

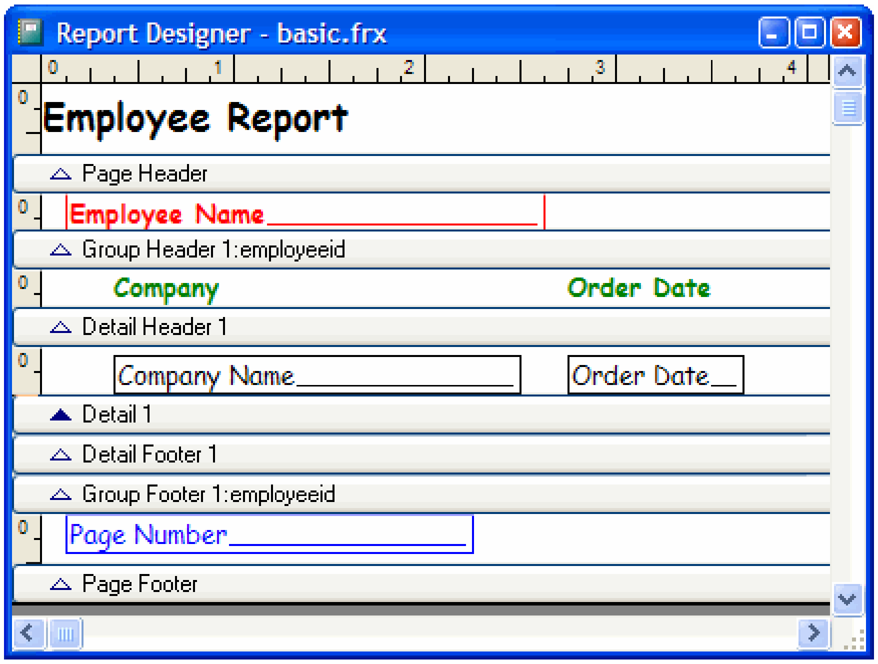
Detail Footer 1

Group Footer 1:employeeid

"Page number: "+TRANS(_pagen

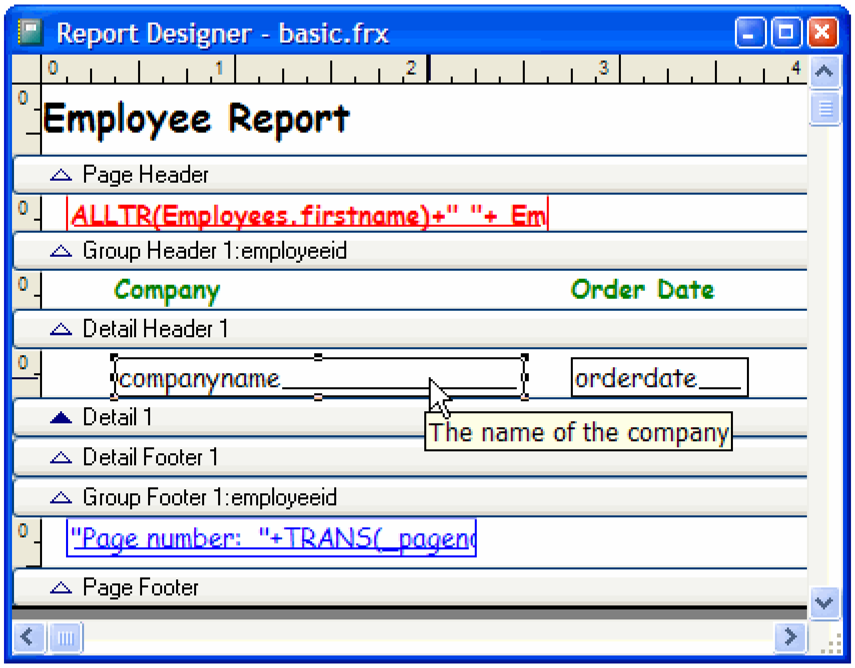
Page Footer

This one used the PROTECTED keyword. Notice that the fields show descriptive names such as Employee Name rather than the actual expression such as ALLTR(Employees.firstname)+" "+ Employees.lastname. This allows you to shield your users from seeing the real expressions used for fields and display meaningful descriptive names instead. Note that if they bring up the Properties dialog, they'll see the real expression because that's the value they'd have to edit rather than the descriptive name.



## Design-Time Tooltips

In addition to design-time captions, you can also specify design-time tooltips for report objects (all object types, not just fields). Set the desired tooltip text in the Tooltip setting on the Other page of the properties dialog for an object.


Report Designer - basic.frx
Employee Report
Page Header
ALLTR(Employees.firstname)+" "+ Em
Group Header 1:employeeid
Company
Order Date
Detail Header 1
companyname
orderdate
The name of the company
Detail 1
Detail Footer 1
Group Footer 1:employeeid
"Page number: "+TRANS(_pagen
Page Footer

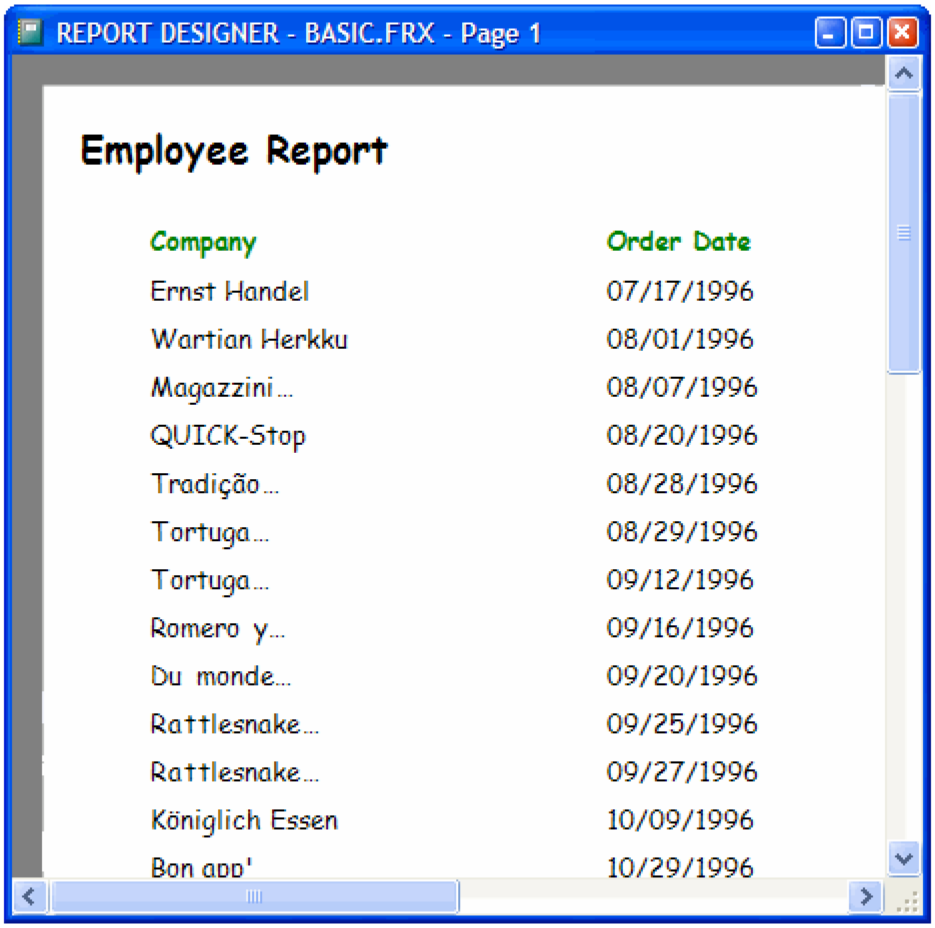
## Absolute Positioning

One thing I've wanted for a long time is the ability to specify the exact size and position of an object by typing the top, left, height, and width values rather than having to carefully move or resize it one pixel at a time to get it to the correct place and shape. The General tab of the object properties dialogs now allow you to do just that.

## Trim Mode for Character Expressions

In earlier versions of VFP, unless you turned on the Stretch with overflow setting, the value of a character expression was truncated if it was too long for the field. In VFP 9, you can specify how the value should appear. The Trim mode for character expressions setting on the Format page of the Field Properties dialog controls this. The choices are:

- Default trimming: an ellipsis is added to the end of the text indicating that there is more data that can't be seen. Notice in the following report that several company names have an ellipsis at the end.



Employee Report

| Company | Order Date |
|---|---|
| Ernst Handel | 07/17/1996 |
| Wartian Herkku | 08/01/1996 |
| Magazzini… | 08/07/1996 |
| QUICK-Stop | 08/20/1996 |
| Tradição… | 08/28/1996 |
| Tortuga… | 08/29/1996 |
| Tortuga… | 09/12/1996 |
| Romero y… | 09/16/1996 |
| Du monde… | 09/20/1996 |
| Rattlesnake… | 09/25/1996 |
| Rattlesnake… | 09/27/1996 |
| Königlich Essen | 10/09/1996 |
| Bon app' | 10/29/1996 |

- Trim to nearest character: this is similar to the truncation that earlier versions of VFP use.

- Trim to nearest word: this cuts off the text at the last whole word.

- Trim to nearest character, append ellipsis: like Trim to nearest character, but adds an ellipsis.

- Trim to nearest word, append ellipsis: same as Default trimming.

- Filespec, show inner path as ellipsis: this has the same effect as the DISPLAYPATH() function; characters at the start and end of the expression appear but middle characters are replaced by an ellipsis.

## DataEnvironment Handling

There are two changes dealing with the DataEnvironment of a report: you can now save the DataEnvironment as a class and you can load the DataEnvironment from another report or from a DataEnvironment class.

VFP 8 added the ability to visually subclass a DataEnvironment. In VFP 9, to save the DataEnvironment of a report as a class, open the DataEnvironment window and choose Save As Class from the File menu.

To load the DataEnvironment of a report from another report or from a DataEnvironment class, choose Load Data Environment from the Report menu or bring up the Report Properties dialog and choose the Data Environment page. Copying the DataEnvironment from another report is pretty straightforward; it simply copies the DataEnvironment-related records from the specified FRX to the current one. Linking the DataEnvironment to a DataEnvironment class, on the other hand, may not do quite what you expect. Unlike a form or form class, an FRX doesn't support referencing a DataEnvironment class. Instead, the various members of the DataEnvironment are loaded into records in the FRX. For example, if there are two cursors and a relation in the DataEnvironment class, records for these objects are added to the FRX. Code in the DataEnvironment class is handled in a very interesting way: code is inserted into various methods of the DataEnvironment, Cursor, and Relation records, and the BeforeOpenTables method has code that instantiates the specified DataEnvironment class and binds events of the DataEnvironment in the report to the appropriate events in the DataEnvironment class. That way, the code in the DataEnvironment class fires as you'd expect. It's just that it's wired up differently than a form or form class is.

## International Support

The Windows Font dialog includes a Script setting that allows a user to select the desired language script. Values include Western, Cyrillic, Japanese, Hebrew, and Arabic. VFP objects support this with their FontCharSet property. Unfortunately, earlier versions of VFP didn't store the selected script in a report, either for report objects or for the default font for the report. In VFP 9, this value is now saved so full support is provided.

In addition, the VFP team ensured that alignment works better in both left-to-right and right-to-left languages.

## Multiple Detail Bands

Crystal Reports is one of the most popular reporting tools in the world. One of the main reasons many VFP developers, who have a report writer built into the product, use Crystal is because it supports sub-reports. A sub-report is a report that runs within a report. The most common use for a sub-report is to report on multiple children for a parent table.

For example, suppose you have a customers table, an invoices table, and a credit notes table. You may want to show customers, their invoices, and credit notes on a report. The complication here is that the report has three tables it needs to go through, and while invoices and credit notes are related to customers, they aren't related to each other. The way you'd resolve this in Crystal is to create a report showing customers and their invoices and then add a sub-report to it showing the credit notes for the current customer.

Unfortunately, until now, there wasn't a good way to do this in VFP. One commonly used workaround is to create a cursor combining invoices and credit notes, with a "record type" field distinguishing which records come from which table, and then have the report include fields from both types of records in the detail band with Print When expressions on the fields so only certain fields print for each type of record. This makes for a very ugly report to maintain!

Fortunately, VFP 9 solves this problem nicely with a new feature: multiple detail bands.

### Record Processing

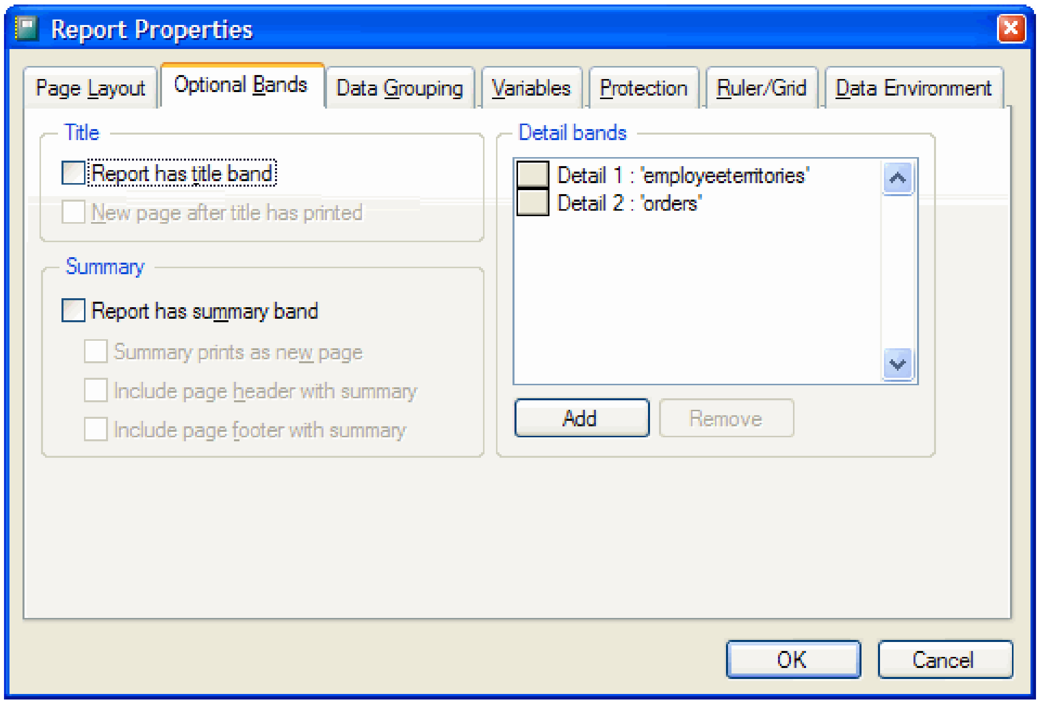
Before we look at multiple detail bands, let's discuss how VFP moves through records in a report. A report has a single "driving" cursor. VFP moves through this cursor in a single pass; that is, the cursor is processed only once. The processing of these records is paused by group breaks; the report engine takes whatever action is specified (for example, printing a group footer for the former group and a group header for the new one) and then continues processing the cursor. The set of records processed without interruption can be referred to as a "detail scope". If there are any groups for the report, the detail scope will be those records inside the innermost group. If not, it'll be the entire report scope.

In VFP 9, there can now be multiple detail scopes (up to 20). The records for a particular detail scope can be from related records in child tables or they can be from the driving alias, which means it can be processed multiple times. The Report Designer presents these multiple detail scopes as multiple detail bands. An important thing to note is that detail scopes are consecutive, not nested like group breaks are.

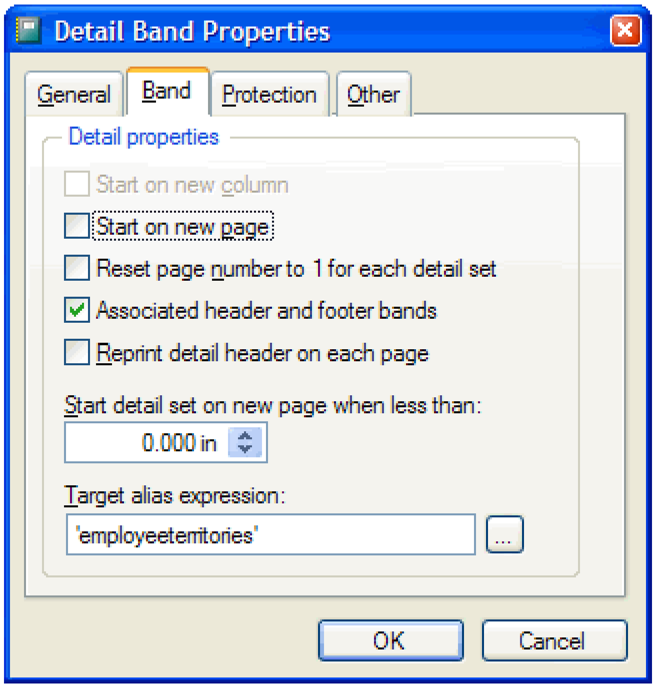
Calculated fields and report variables can now be scoped to a particular detail band. Interestingly, variables will retain their values until the band they're scoped to is processed again. This means you can use these variables in later detail bands if necessary. The Variables page of the Report Properties dialog now uses "reset based on" rather than "reset at" as the prompt for a variable's scope to reinforce this idea.

## Creating Multiple Detail Bands

Every report has at least one detail band. To create additional detail bands, choose Optional Bands from the Report menu or the report shortcut menu, or bring up the Report Properties dialog and choose the Optional Bands page. The Add button adds a new detail band and the Remove button removes the selected detail band. You can rearrange the order of the bands in the list.



In the properties dialog for a detail band, you can specify whether the band has header and footer bands and what the target alias expression is for the detail scope. Ordinarily, the report engine processes a single record in the driving cursor before moving to the next detail band. However, if you specify a child cursor as the target alias, the report engine will process all child records of the current driving cursor record before moving to the next band. Note that you enter the target alias as an expression; to use a hard-coded name, surround it with quotes. However, since this is an expression, you could enter the name of a variable that contains the target alias or even call a UDF. This may lead to some very interesting types of reports!



Detail properties

- [ ] Start on new column
- [ ] Start on new page
- [ ] Reset page number to 1 for each detail set
- [✓] Associated header and footer bands
- [ ] Reprint detail header on each page

Start detail set on new page when less than: 0.000 in

Target alias expression: 'employeeterritories'

The target alias expression can evaluate to one of three values:

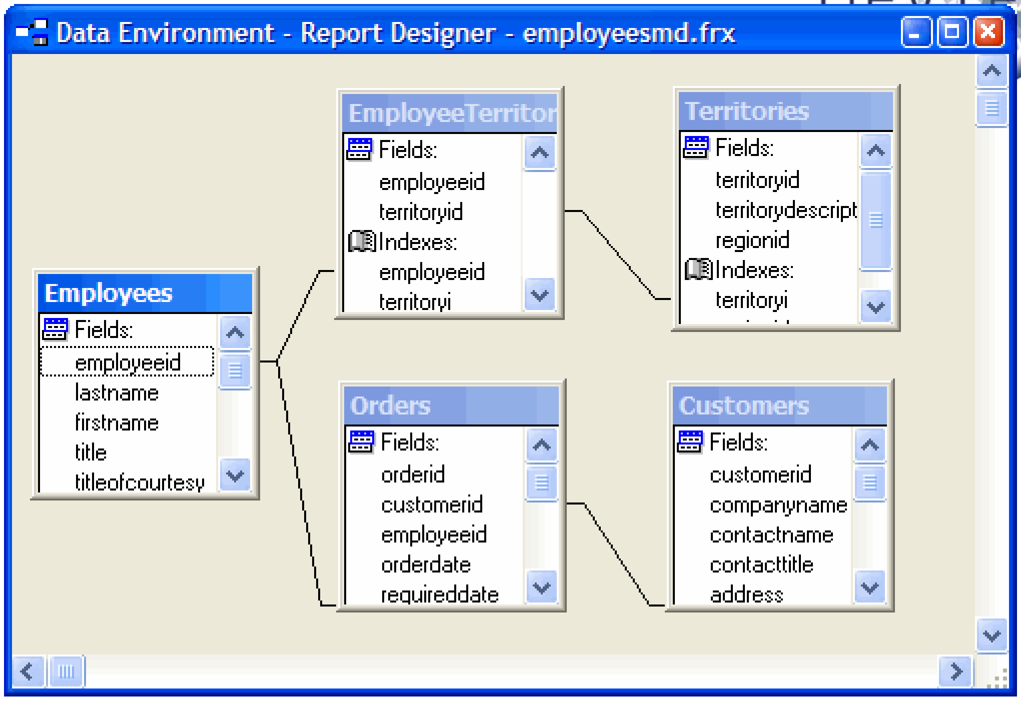
- An empty string means that the driving alias is used.
- The alias of a child table means that the report engine will process all child records of the current driving cursor record before moving to the next band.
- The driving alias, which is valid in the first detail band only, means the report engine will process all driving alias records until it encounters a group break or the end of the report scope before moving to the next detail band.

Let's look at a couple of examples of multiple detail band reports.

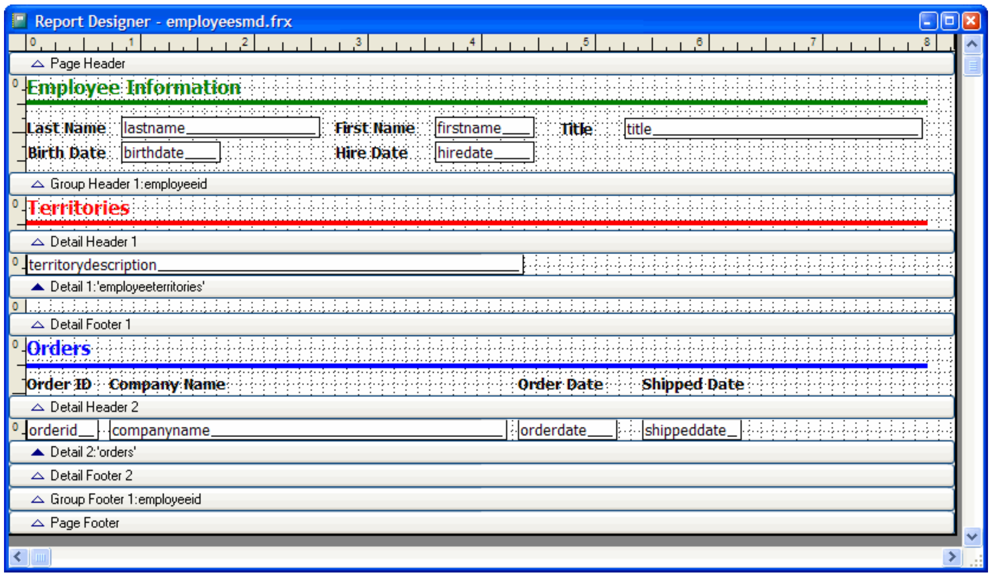
## Example 1: Multiple Children

The first example uses the Employees, EmployeeTerritories, and Orders tables from the Northwind sample database that comes with VFP (in the Samples\Northwind subdirectory of the VFP home directory). EmployeeTerritories and Orders are both child tables of Employees, related on the EmployeeID field in each table. We want a report that shows each employee, the territories they represent, and their orders.

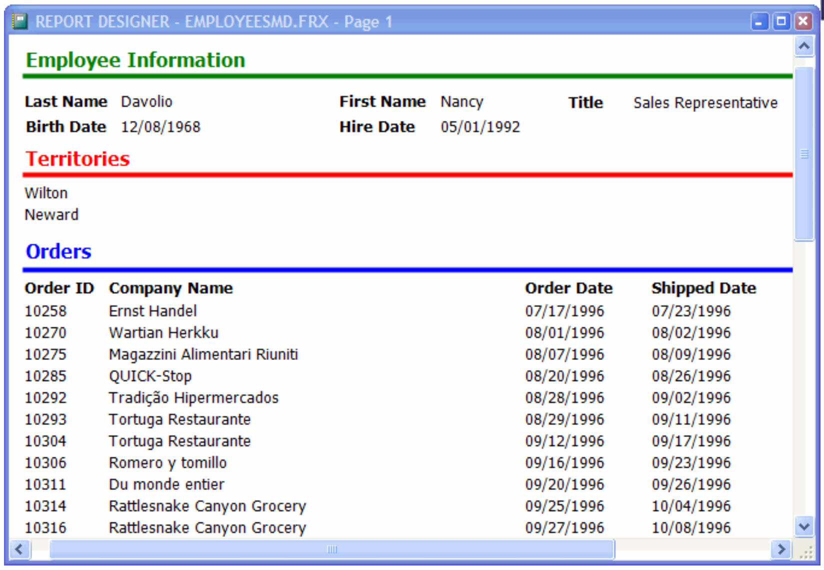
The DataEnvironment for this report (EmployeeMD.FRX) is set up as shown in the following picture. The relationships between Employees and its child tables are one-to-many (the OneToMany property for the Relation objects is .T.) so all child records for a given Employees record will be processed in a detail band. Note this isn't strictly necessary; if you neglect to set OneToMany, the report engine will automatically use SET SKIP to do the same thing.



The report has a group expression on Employees.EmployeeID and the desired fields from the Employees table appears in the group header band. There are two detail bands, one with a target alias of EmployeeTerritories and the other with a target alias of Orders; the appropriate fields appear in each band. Here's what the report looks like in the Report Designer:



This is the result when the report is run:


REPORT DESIGNER - EMPLOYEESMD.FRX - Page 1

Employee Information

| | | | | | |
|---|---|---|---|---|---|
| Last Name | Davolio | First Name | Nancy | Title | Sales Representative |
| Birth Date | 12/08/1968 | Hire Date | 05/01/1992 | | |

Territories

Wilton
Neward

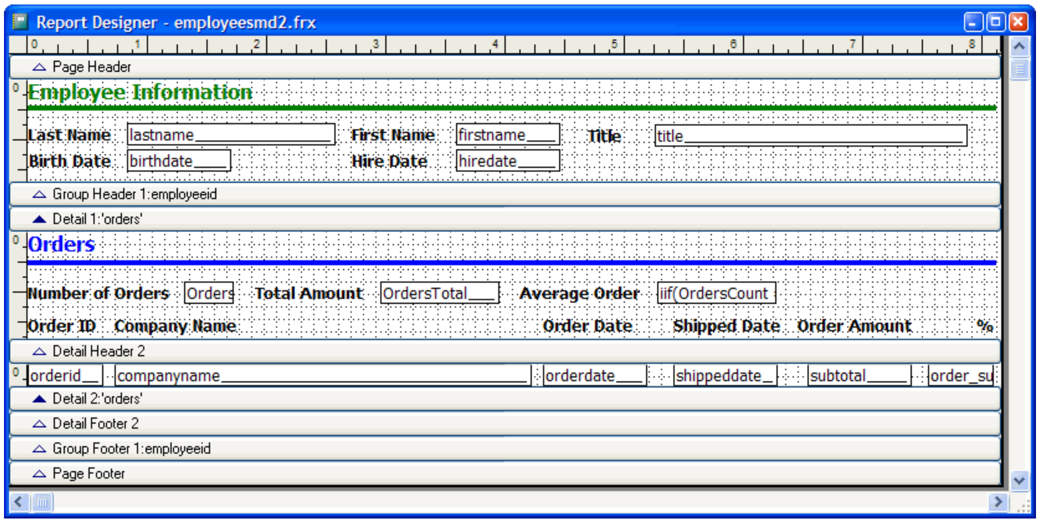
Orders

| Order ID | Company Name | Order Date | Shipped Date |
|---|---|---|---|
| 10258 | Ernst Handel | 07/17/1996 | 07/23/1996 |
| 10270 | Wartian Herkku | 08/01/1996 | 08/02/1996 |
| 10275 | Magazzini Alimentari Riuniti | 08/07/1996 | 08/09/1996 |
| 10285 | QUICK-Stop | 08/20/1996 | 08/26/1996 |
| 10292 | Tradição Hipermercados | 08/28/1996 | 09/02/1996 |
| 10293 | Tortuga Restaurante | 08/29/1996 | 09/11/1996 |
| 10304 | Tortuga Restaurante | 09/12/1996 | 09/17/1996 |
| 10306 | Romero y tomillo | 09/16/1996 | 09/23/1996 |
| 10311 | Du monde entier | 09/20/1996 | 09/26/1996 |
| 10314 | Rattlesnake Canyon Grocery | 09/25/1996 | 10/04/1996 |
| 10316 | Rattlesnake Canyon Grocery | 09/27/1996 | 10/08/1996 |

## Example 2: Pre-calculation of Totals

The next example is similar to the first one, but doesn't show two child tables; instead, it runs through the same child twice. The idea here is that we want to calculate the number and total amount of the orders for each employee, but we want to show these calculations *before* displaying the actual orders. In addition, we want to show the amount of each invoice as a percentage of the total amount, meaning we have to pre-calculate the total.

In previous versions of VFP, this would require doing the calculations prior to running the report, and using the results of those calculations in the report. In VFP 9, this simply means having one detail band that does the calculations and another that displays the results. In the case of this example, EmployeesMD2.FRX, both detail bands have the Orders table as the target alias. Here's what the report looks like in the Report Designer; notice that there are no objects in the Detail 1 band:


Report Designer - employeesmd2.frx

Page Header

Employee Information

Last Name lastname First Name firstname Title title
Birth Date birthdate Hire Date hiredate

Group Header 1:employeeid

Detail 1:'orders'

Orders

Number of Orders Orders Total Amount OrdersTotal Average Order iif(OrdersCount

Order ID Company Name Order Date Shipped Date Order Amount %

Detail Header 2

orderid companyname orderdate shippeddate subtotal order_su

Detail 2:'orders'

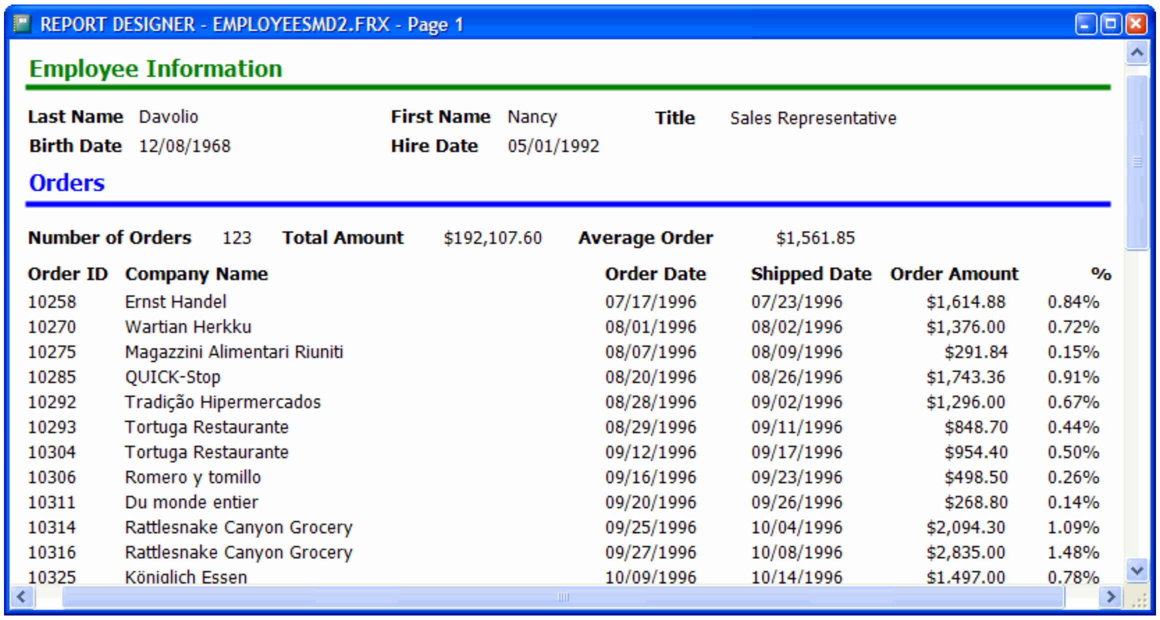
Detail Footer 2

Group Footer 1:employeeid

Page Footer

There are two variables defined: OrdersCount, which is a "count" variable that's reset based on Detail 1, and OrdersTotal, which sums Order_Subtotals.Subtotal and is also reset based on Detail 1. Order_Subtotals is a view in the Northwind database that calculates the subtotal for each order, so summing the Subtotal field from that view (which is related to the Orders table, so it's positioned to the correct record) gives the total of the desired orders. So, all the Detail 1 band is used for is to process all order records for the current employee and calculate the proper values in the OrdersCount and OrdersTotal variables. Then, the order records are processed a second time in Detail 2. The number of orders and total amount is shown in the header for Detail 2 and the orders and the percentage that each is of the total amount is shown in the detail band.

Here's what the report looks like when it's run:



Employee Information

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| **Last Name** | Davolio | **First Name** | Nancy | **Title** | Sales Representative |
| **Birth Date** | 12/08/1968 | **Hire Date** | 05/01/1992 |  |  |

Orders

**Number of Orders** 123 **Total Amount** $192,107.60 **Average Order** $1,561.85

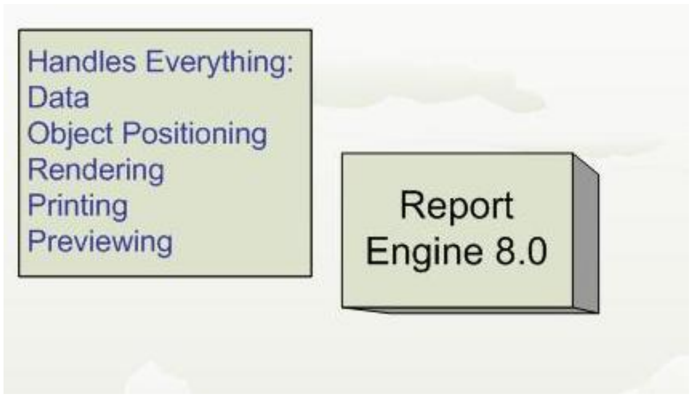
| Order ID | Company Name | Order Date | Shipped Date | Order Amount | % |
|---|---|---|---|---|---|
| 10258 | Ernst Handel | 07/17/1996 | 07/23/1996 | $1,614.88 | 0.84% |
| 10270 | Wartian Herkku | 08/01/1996 | 08/02/1996 | $1,376.00 | 0.72% |
| 10275 | Magazzini Alimentari Riuniti | 08/07/1996 | 08/09/1996 | $291.84 | 0.15% |
| 10285 | QUICK-Stop | 08/20/1996 | 08/26/1996 | $1,743.36 | 0.91% |
| 10292 | Tradição Hipermercados | 08/28/1996 | 09/02/1996 | $1,296.00 | 0.67% |
| 10293 | Tortuga Restaurante | 08/29/1996 | 09/11/1996 | $848.70 | 0.44% |
| 10304 | Tortuga Restaurante | 09/12/1996 | 09/17/1996 | $954.40 | 0.50% |
| 10306 | Romero y tomillo | 09/16/1996 | 09/23/1996 | $498.50 | 0.26% |
| 10311 | Du monde entier | 09/20/1996 | 09/26/1996 | $268.80 | 0.14% |
| 10314 | Rattlesnake Canyon Grocery | 09/25/1996 | 10/04/1996 | $2,094.30 | 1.09% |
| 10316 | Rattlesnake Canyon Grocery | 09/27/1996 | 10/08/1996 | $2,835.00 | 1.48% |
| 10325 | Königlich Essen | 10/09/1996 | 10/14/1996 | $1,497.00 | 0.78% |

## Enhanced Run-Time Capabilities

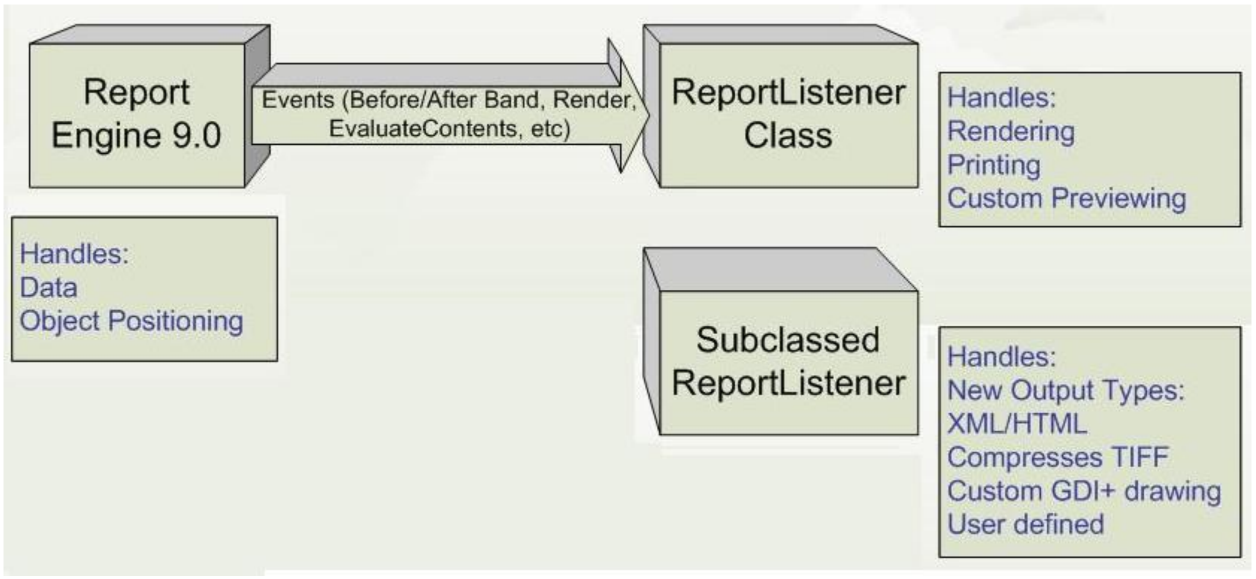
The VFP development team had several goals in mind when they worked on the run-time improvements, including:

- Handling more types of report output than just printing and previewing.
- Using GDI+ for report output. This provides many significant improvements, such as more accurate rendering, smooth scaling up and down of images and fonts, and additional capabilities such as text rotation.
- Providing a more flexible and extendible reporting system.

Before VFP 9, the report engine was rather monolithic; it handled everything and with a few exceptions (UDFs, expressions for OnEntry and OnExit of bands, etc.), you couldn't interact with it during a report run.



The new reporting engine in VFP 9 splits responsibility for reporting between the report engine, which now just deals with data-handling and object positioning, and a new object known as a report listener, which handles rendering and output. Because report listeners are classes, we can now interact with the reporting process in ways we could only dream of before.



VFP 9 includes both the old report engine and the new one, so you can run reports under either engine as you see fit. However, once you see the benefits of the new report engine, you won't want to go back to old-style reporting unless it's absolutely necessary.

## New Reporting Syntax

VFP 9 supports running reports using the old report engine; simply use the REPORT command as you did before (although, as we'll see in a moment, you can use a new command to override the behavior of REPORT). To get new-style reporting behavior, use the new OBJECT clause of the REPORT command. OBJECT supports two ways of using it: by specifying a report listener and specifying a report type. Microsoft refers to this as "object-assisted" reporting.

A report listener is an object that provides new-style reporting behavior. Report listeners are based on a new base class in VFP 9, ReportListener. To tell VFP to use a specific listener for a report, instantiate the listener class and then specify the object's name in the OBJECT clause of the REPORT command. Here's an example:

```
loListener = createobject('MyReportListener')
report form MyReport object loListener
```

If you'd rather not instantiate a listener manually, you can have VFP do it for you automatically by specifying a report type:

```
report form MyReport object type 1
```

The defined types are 0 for outputting to a printer, 1 for previewing, 4 for XML output, and 5 for HTML output. Other types can also be used.

When you run a report this way, the application specified in the new _REPORTOUTPUT system variable (by default, ReportOutput.APP in the VFP home directory) is called to figure out which listener class to instantiate for the specified type.

You're probably thinking "But I've got tons of reports in my application. Do I have to find and modify every REPORT command in the entire application?" Fortunately, there's an easier way: SET REPORTBEHAVIOR 90 turns on object-assisted reporting by default. This means the REPORT command behaves as if you've specified OBJECT TYPE 0 when you use the TO PRINT clause or OBJECT TYPE 1 when you use the PREVIEW clause. SET REPORTBEHAVIOR 80 reverts to VFP 8 and earlier behavior. If most or all of the reports in your application work just fine in object-assisted mode, use SET REPORTBEHAVIOR 90 at application startup. Because there are rendering differences between new and old-style reporting, some reports may need to be tweaked to work properly with new-style reporting, so either tweak them or use SET REPORTBEHAVIOR 80 to run just those reports.

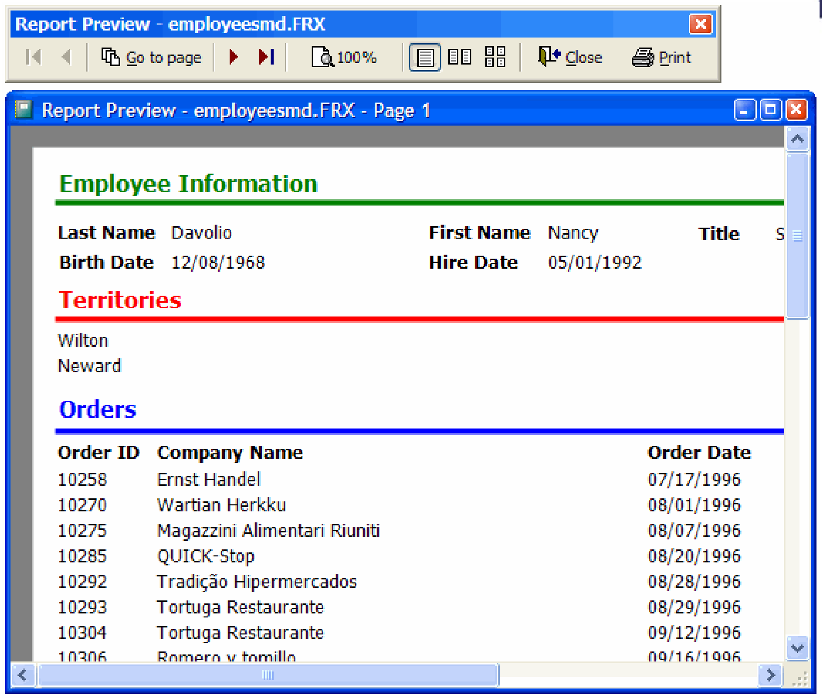
ReportOutput.APP is primarily an object factory: it instantiates the appropriate listener for a report. It also includes some listeners that provide XML and HTML output. However, since it's just an Xbase application, you could substitute it with your own application by setting _REPORTOUTPUT accordingly.

ReportOutput.APP and report listeners are discussed in detail in my "Extending the VFP 9 Reporting System, Part II: Run-Time" document.

Let's take a look at some specifics of the run-time improvements.

## New Preview Window

At first glance, the preview window for a report in VFP 9 may not look much different than it does in earlier versions. However, take a close look at the toolbar.


Report Preview - employeesmd.FRX
Go to page
100%
Close
Print
Report Preview - employeesmd.FRX - Page 1

Employee Information

| | | | | | |
|---|---|---|---|---|---|
| Last Name | Davolio | First Name | Nancy | Title | S |
| Birth Date | 12/08/1968 | Hire Date | 05/01/1992 | | |

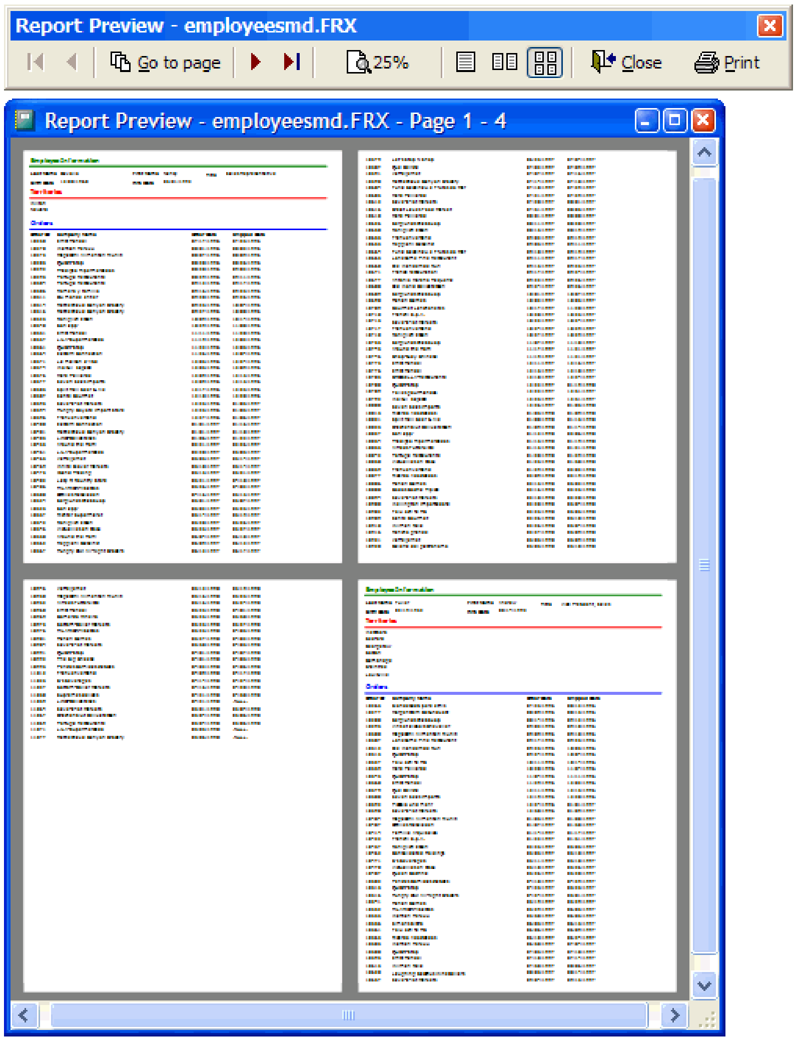
Territories

Wilton
Neward

Orders

| Order ID | Company Name | Order Date |
|---|---|---|
| 10258 | Ernst Handel | 07/17/1996 |
| 10270 | Wartian Herkku | 08/01/1996 |
| 10275 | Magazzini Alimentari Riuniti | 08/07/1996 |
| 10285 | QUICK-Stop | 08/20/1996 |
| 10292 | Tradição Hipermercados | 08/28/1996 |
| 10293 | Tortuga Restaurante | 08/29/1996 |
| 10304 | Tortuga Restaurante | 09/12/1996 |
| 10306 | Romero y tomillo | 09/16/1996 |

Notice there are new buttons allowing you to specify the number of pages displayed at a time. Here's what 4 pages at a time looks like:


Report Preview - employeesmd.FRX
Go to page
25%
Close
Print
Report Preview - employeesmd.FRX - Page 1 - 4

Also notice you can now scale the report above 100%: 200, 300, and 500% are supported (this works in both the new and old preview windows). Another new feature is that a shortcut menu is now provided.

Finally, try the following:

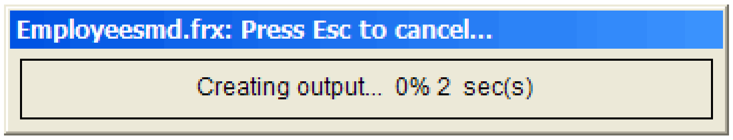
```
_screen.Forms[2].Caption = 'This is my report'
```

That's right: the preview window is an Xbase form rather than a native window. That means you have full control over its appearance, something that was painful to do in previous versions.

A new system variable, _REPORTPREVIEW, specifies the name of an Xbase application that provides the preview window for reports. By default, this variable points to ReportPreview.APP in the VFP home directory, but you could substitute your own application if you wish to. You can specify the form to use for output in other ways as well. I won't go into detail on this as it's beyond the scope of this document.

## Progress Feedback

When a long report runs, you may wish there was some way of showing the user that something's happening. The UpdateListener class built into ReportOutput.APP does this; it displays the progress as the report is processed, and also gives the ability to cancel. You can disable this feedback if you want to, or provide your own using a report listener. An example of this is shown in my "Extending the VFP 9 Reporting System, Part II: Run-Time" document.


Employeesmd.frx: Press Esc to cancel...

Creating output... 0% 2 sec(s)

## HTML and XML Output

Although we've been able to output a report to HTML for several versions using GenHTML.PRG, the results, frankly, aren't very good. Fortunately, in VFP 9, we can create high-quality HTML output, as well as XML output, using some listeners that are built into ReportOutput.APP.

Listener type 5 specifies HTML output and type 4 is for XML output, so you could just use the following command to output to HTML:

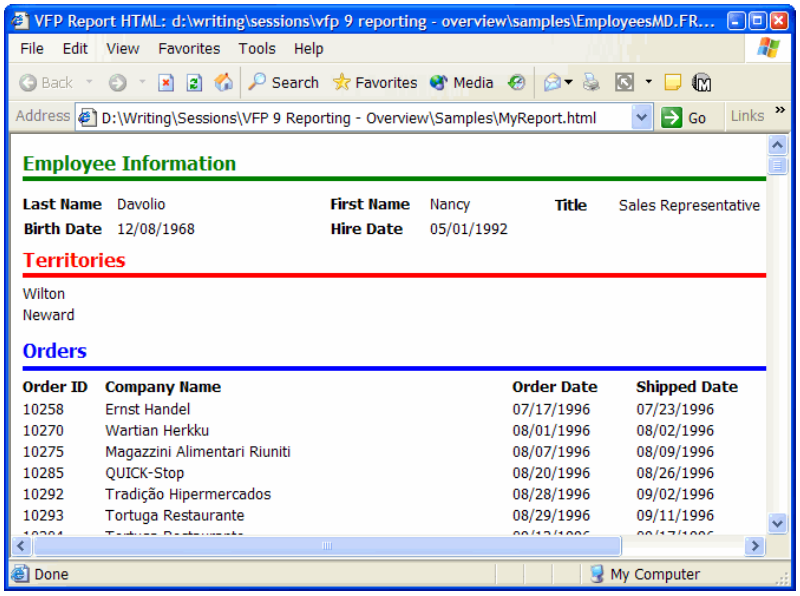
```
report form MyReport object type 5
```

However, this doesn't give you any control over the name of the file to create or other settings. Instead, we'll call ReportOutput.APP to give us a reference to the desired listener, set some properties, and then tell the REPORT command to use that listener.

The following code (taken from HTMLOutput.PRG) creates an HTML file called MyReport.html from the first six pages of the EmployeesMD report. When you specify type 5, ReportOutput.APP uses its built-in HTMLListener class to provide output.

```
loListener = .NULL.
do (_reportoutput) with 5, loListener
loListener.TargetFileName = 'MyReport.html'
loListener.QuietMode = .T.
report form EmployeesMD object loListener range 1, 6
```

Here's what the output looks like:


VFP Report HTML: d:\writing\sessions\vfp 9 reporting - overview\samples\EmployeesMD.FR...
File Edit View Favorites Tools Help
Back Search Favorites Media
Address D:\Writing\Sessions\VFP 9 Reporting - Overview\Samples\MyReport.html Go Links
Done My Computer

Employee Information

| | | | | | |
|---|---|---|---|---|---|
| Last Name | Davolio | First Name | Nancy | Title | Sales Representative |
| Birth Date | 12/08/1968 | Hire Date | 05/01/1992 | | |

Territories

Wilton
Neward

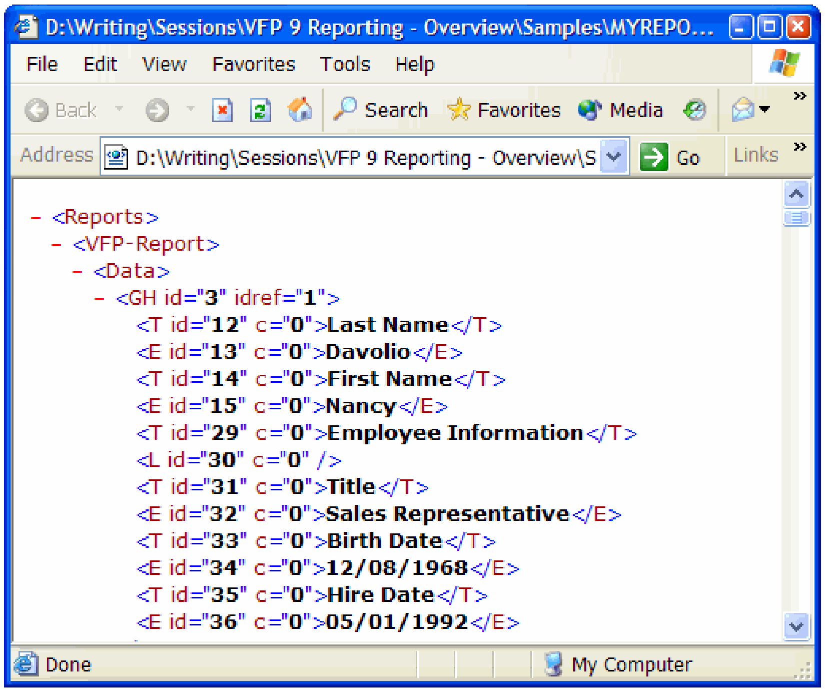
Orders

| Order ID | Company Name | Order Date | Shipped Date |
|---|---|---|---|
| 10258 | Ernst Handel | 07/17/1996 | 07/23/1996 |
| 10270 | Wartian Herkku | 08/01/1996 | 08/02/1996 |
| 10275 | Magazzini Alimentari Riuniti | 08/07/1996 | 08/09/1996 |
| 10285 | QUICK-Stop | 08/20/1996 | 08/26/1996 |
| 10292 | Tradição Hipermercados | 08/28/1996 | 09/02/1996 |
| 10293 | Tortuga Restaurante | 08/29/1996 | 09/11/1996 |

The following code (taken from XMLOutput.PRG) creates an XML file called MyReport.xml from the first six pages of the EmployeesMD report, containing only the data. In this case, the XMLListener class in ReportOutput.APP is used.

```
loListener = .NULL.
do (_reportoutput) with 4, loListener
loListener.TargetFileName = 'MyReport.xml'
loListener.QuietMode = .T.
loListener.XMLMode = 0  && 0 = data only, 1 = layout only, 2 = both
report form EmployeesMD object loListener range 1, 6
```

Here's the result:


D:\Writing\Sessions\VFP 9 Reporting - Overview\Samples\MYREPO...
File Edit View Favorites Tools Help
Back Search Favorites Media
Address D:\Writing\Sessions\VFP 9 Reporting - Overview\S Go Links
Done My Computer

```
- <Reports>
  - <VFP-Report>
    - <Data>
      - <GH id="3" idref="1">
          <T id="12" c="0">Last Name</T>
          <E id="13" c="0">Davolio</E>
          <T id="14" c="0">First Name</T>
          <E id="15" c="0">Nancy</E>
          <T id="29" c="0">Employee Information</T>
          <L id="30" c="0" />
          <T id="31" c="0">Title</T>
          <E id="32" c="0">Sales Representative</E>
          <T id="33" c="0">Birth Date</T>
          <E id="34" c="0">12/08/1968</E>
          <T id="35" c="0">Hire Date</T>
          <E id="36" c="0">05/01/1992</E>
```

HTML output actually uses the XML listener to produce XML and then uses XSLT to produce the HTML end-result.

Both of these listener classes have additional properties you can use to control the output. See the VFP documentation for details.

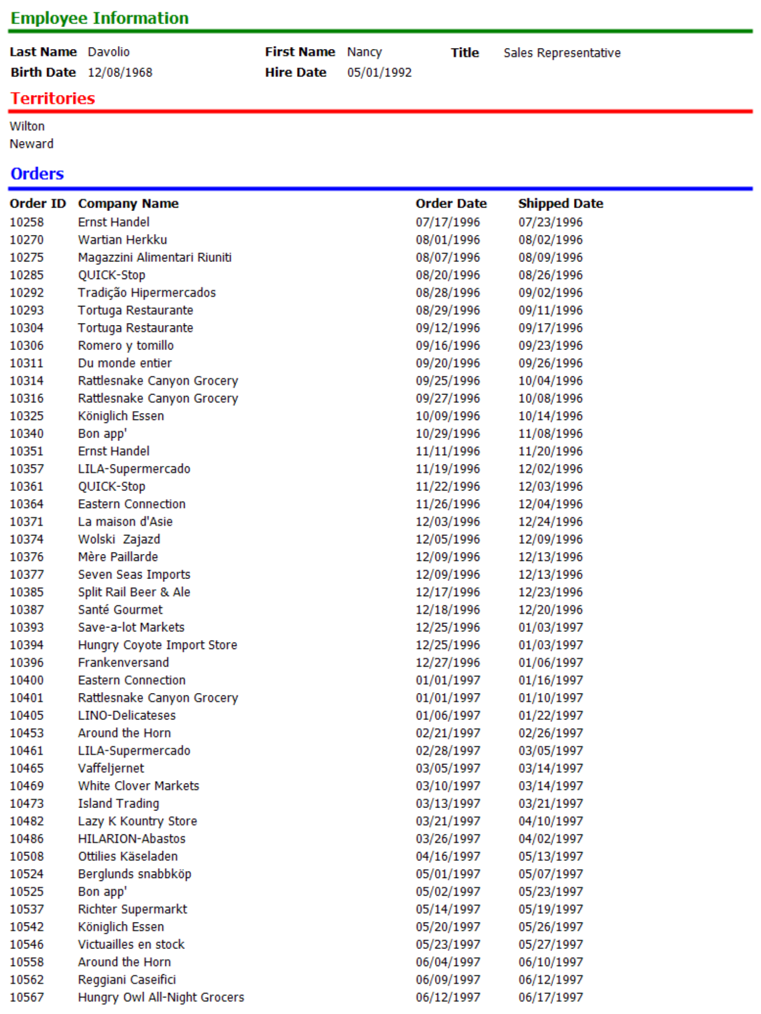
## Graphic File Output

With a report listener, you can also output to a graphic file. VFP 9 supports EMF, TIFF (single and multi-page), JPG, BMP, PNG, and GIF. Third-party listeners may support other files types as well.

Here's some simple code, taken from GraphicOutput.PRG, that outputs a single page of the EmployeesMD report to a GIF file.

```
loListener = createobject('ReportListener')
loListener.ListenerType = 3
report form EmployeesMD object loListener range 1, 1
loListener.OutputPage(1, 'MyReport.gif', 104)
```

Here's the result:



**Employee Information**

**Last Name** Davolio **First Name** Nancy **Title** Sales Representative
**Birth Date** 12/08/1968 **Hire Date** 05/01/1992

**Territories**

Wilton
Neward

**Orders**

| Order ID | Company Name | Order Date | Shipped Date |
|---|---|---|---|
| 10258 | Ernst Handel | 07/17/1996 | 07/23/1996 |
| 10270 | Wartian Herkku | 08/01/1996 | 08/02/1996 |
| 10275 | Magazzini Alimentari Riuniti | 08/07/1996 | 08/09/1996 |
| 10285 | QUICK-Stop | 08/20/1996 | 08/26/1996 |
| 10292 | Tradição Hipermercados | 08/28/1996 | 09/02/1996 |
| 10293 | Tortuga Restaurante | 08/29/1996 | 09/11/1996 |
| 10304 | Tortuga Restaurante | 09/12/1996 | 09/17/1996 |
| 10306 | Romero y tomillo | 09/16/1996 | 09/23/1996 |
| 10311 | Du monde entier | 09/20/1996 | 09/26/1996 |
| 10314 | Rattlesnake Canyon Grocery | 09/25/1996 | 10/04/1996 |
| 10316 | Rattlesnake Canyon Grocery | 09/27/1996 | 10/08/1996 |
| 10325 | Königlich Essen | 10/09/1996 | 10/14/1996 |
| 10340 | Bon app' | 10/29/1996 | 11/08/1996 |
| 10351 | Ernst Handel | 11/11/1996 | 11/20/1996 |
| 10357 | LILA-Supermercado | 11/19/1996 | 12/02/1996 |
| 10361 | QUICK-Stop | 11/22/1996 | 12/03/1996 |
| 10364 | Eastern Connection | 11/26/1996 | 12/04/1996 |
| 10371 | La maison d'Asie | 12/03/1996 | 12/24/1996 |
| 10374 | Wolski  Zajazd | 12/05/1996 | 12/09/1996 |
| 10376 | Mère Paillarde | 12/09/1996 | 12/13/1996 |
| 10377 | Seven Seas Imports | 12/09/1996 | 12/13/1996 |
| 10385 | Split Rail Beer & Ale | 12/17/1996 | 12/23/1996 |
| 10387 | Santé Gourmet | 12/18/1996 | 12/20/1996 |
| 10393 | Save-a-lot Markets | 12/25/1996 | 01/03/1997 |
| 10394 | Hungry Coyote Import Store | 12/25/1996 | 01/03/1997 |
| 10396 | Frankenversand | 12/27/1996 | 01/06/1997 |
| 10400 | Eastern Connection | 01/01/1997 | 01/16/1997 |
| 10401 | Rattlesnake Canyon Grocery | 01/01/1997 | 01/10/1997 |
| 10405 | LINO-Delicateses | 01/06/1997 | 01/22/1997 |
| 10453 | Around the Horn | 02/21/1997 | 02/26/1997 |
| 10461 | LILA-Supermercado | 02/28/1997 | 03/05/1997 |
| 10465 | Vaffeljernet | 03/05/1997 | 03/14/1997 |
| 10469 | White Clover Markets | 03/10/1997 | 03/14/1997 |
| 10473 | Island Trading | 03/13/1997 | 03/21/1997 |
| 10482 | Lazy K Kountry Store | 03/21/1997 | 04/10/1997 |
| 10486 | HILARION-Abastos | 03/26/1997 | 04/02/1997 |
| 10508 | Ottilies Käseladen | 04/16/1997 | 05/13/1997 |
| 10524 | Berglunds snabbköp | 05/01/1997 | 05/07/1997 |
| 10525 | Bon app' | 05/02/1997 | 05/23/1997 |
| 10537 | Richter Supermarkt | 05/14/1997 | 05/19/1997 |
| 10542 | Königlich Essen | 05/20/1997 | 05/26/1997 |
| 10546 | Victuailles en stock | 05/23/1997 | 05/27/1997 |
| 10558 | Around the Horn | 06/04/1997 | 06/10/1997 |
| 10562 | Reggiani Caseifici | 06/09/1997 | 06/12/1997 |
| 10567 | Hungry Owl All-Night Grocers | 06/12/1997 | 06/17/1997 |

See my “Extending the VFP 9 Reporting System, Part II: Run-Time” document for details on this topic.

## What About PDF?

Of course, the question you’re asking now is “What about PDF output?” As of this writing, Microsoft has no plans to include PDF output with VFP 9. However, there are several ways you can get PDF output from VFP (now or in VFP 9):

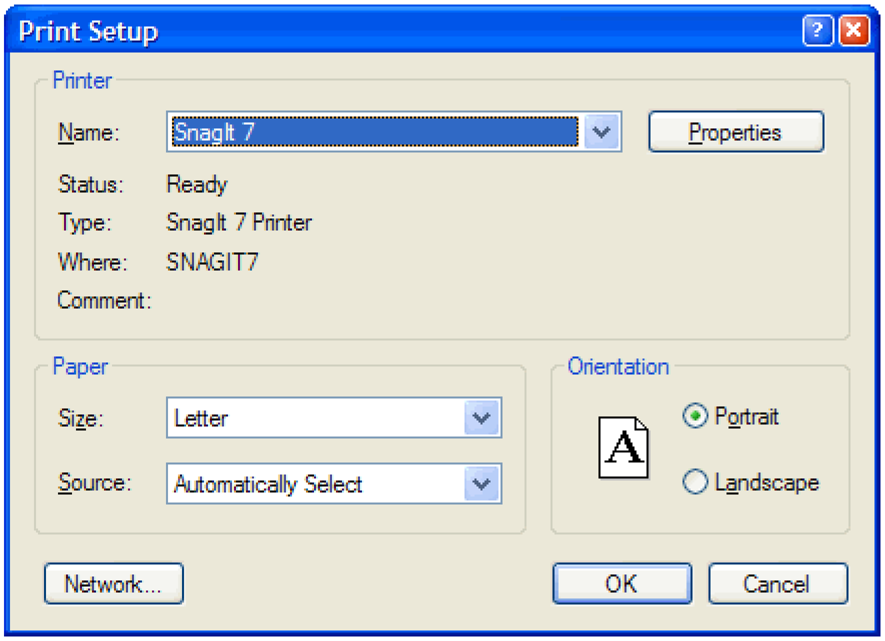
- Use the Adobe or another PDF writer.
- Use a VFP-specific third-party tool that supports PDF output, such as Mind’s Eye Report Engine, XFRX, or FRX2Any.
- In VFP 9, you can output to a TIFF file and then use the freeware GhostScript utility to convert it to a PDF file.

Because this was written early in the VFP 9 beta, this is still up in the air. Stay tuned, though–I can guarantee there will be some great solutions available soon.

## Other Printing Enhancements

There are a number of other general printing enhancements in VFP 9.

SYS(1037), which displays the Page Setup dialog, has some new capabilities. First, the dialog looks different. Here's the VFP 8 version:


Print Setup

Printer

Name: SnagIt 7

Status: Ready

Type: SnagIt 7 Printer

Where: SNAGIT7

Comment:

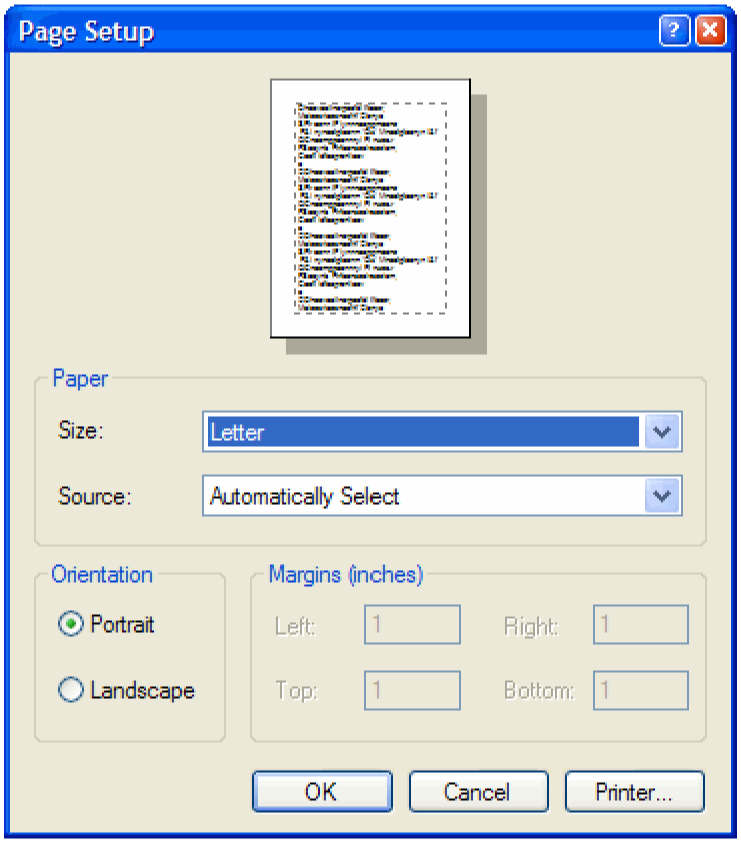
Paper

Size: Letter

Source: Automatically Select

Orientation: [✓ Portrait] Landscape

Here's what it looks like in VFP 9:

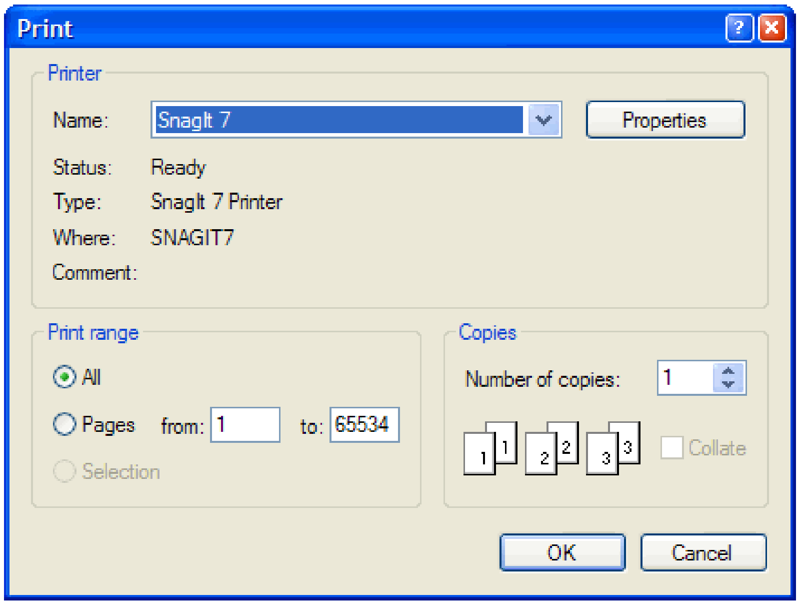


Second, it now has a return value indicating whether the user took any action (“1”) or not (“0”).

The biggest change, though, is that you can pass it a parameter to tell it what to do. Passing 0 or no parameter displays the default Page Setup dialog. To display and possibly change the Page Setup settings for a particular report, open the report as a table (that is, USE MyReport.FRX) and then call SYS(1037, 1). You can save and restore the current default printer settings using SYS(1037, 2) (which writes the default printer settings to an FRX) and SYS(1037, 3) (which sets the default printer settings to those in the FRX); neither of these displays the Page Setup dialog. This is typically used to push and pop printer settings.

APRINTERS() can now accept a new optional argument of 1, in which case the resulting array will have three new columns showing the driver, comment, and location.

The Print dialog displayed when you use PROMPT keyword in the REPORT command has a more modern appearance. Here’s the VFP 8 version:


Print

Printer

Name: SnagIt 7

Status: Ready

Type: SnagIt 7 Printer

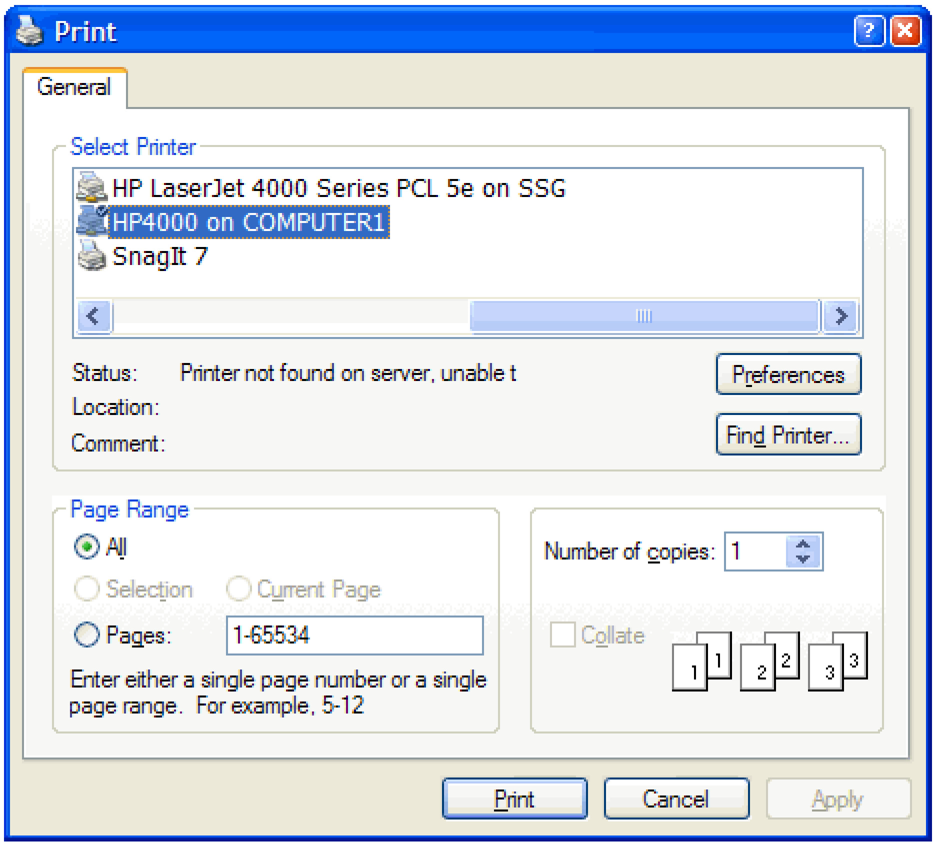
Where: SNAGIT7

Comment:

Print range: (● All) ○ Pages from: 1 to: 65534; ○ Selection

Copies: Number of copies: 1; ☐ Collate

Here’s what it looks like in VFP 9:


Print

General

Select Printer

HP LaserJet 4000 Series PCL 5e on SSG

HP4000 on COMPUTER1

SnagIt 7

Status: Printer not found on server, unable t

Location:

Comment:

Page Range: (● All) ○ Selection ○ Current Page ○ Pages: 1-65534

Enter either a single page number or a single page range. For example, 5-12

Number of copies: 1; ☐ Collate

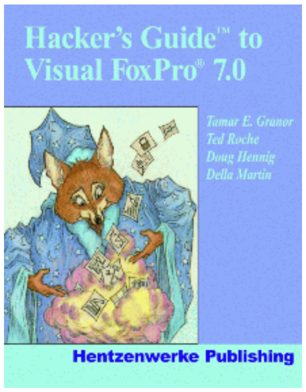
## Summary

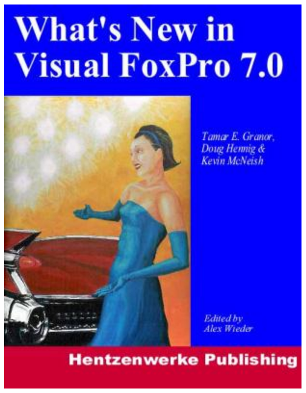
VFP 9 has an incredible number of changes in the reporting engine. These changes make it easier to work with the Report Designer, make it possible to create types of reports we either couldn’t do before or were hard to do, provide new types of output, and allow us to extend the capabilities of the reporting engine in both design-time and run-time environments. We can expect a lot of new uses for VFP reporting, and many new discoveries about what its capabilities are in the years to come.

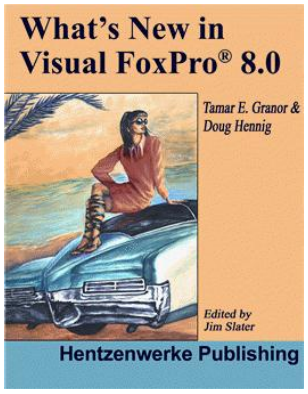
Note: since this document was written during the first beta phase of VFP 9, many of the details described in this document may be different in the release version. Be sure to check my Web site (www.stonefield.com) for updates to this document and the accompanying samples.

## Biography

Doug Hennig is a partner with Stonefield Systems Group Inc. and Stonefield Software Inc. He is the author of the award-winning Stonefield Database Toolkit (SDT), the award-winning Stonefield Query, and the CursorAdapter and DataEnvironment builders that come with Microsoft Visual FoxPro. Doug is co-author of “What’s New in Visual FoxPro 8.0”, “The Hacker’s Guide to Visual FoxPro 7.0”, and “What’s New in Visual FoxPro 7.0”. He was the technical editor of “The Hacker’s Guide to Visual FoxPro 6.0” and “The Fundamentals”. All of these books are from Hentzenwerke Publishing (http://www.hentzenwerke.com). Doug writes the monthly “Reusable Tools” column in FoxTalk. He has spoken at every Microsoft FoxPro Developers Conference (DevCon) since 1997 and at user groups and developer conferences all over North America. He has been a Microsoft Most Valuable Professional (MVP) since 1996.




What's New in Visual FoxPro 7.0
Tamar E. Granor, Doug Hennig & Kevin McNeish
Edited by Alex Wieder
Hentzenwerke Publishing


What’s New in Visual FoxPro® 8.0
Tamar E. Granor & Doug Hennig
Edited by Jim Slater
Hentzenwerke Publishing