March, 2006

Advisor Answers

Formatting Reports Based on Data

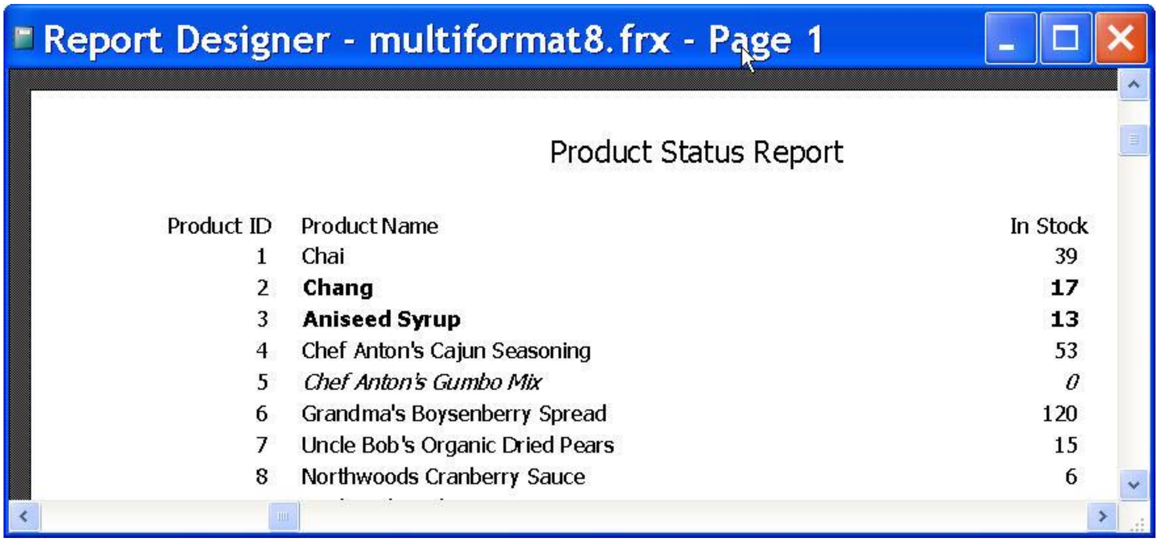
VFP 9/8/7

Q: I'm working on a report where I need to use bold for some items, based on their contents. That is, for certain data values, I want bold, while the other values should be in the normal typeface.

A: The VFP Report Designer doesn't provide any direct way to specify that formatting of an item should be based on the item's content. Fortunately, there is a solution. In VFP 8 and earlier, getting the results you want takes a little work and makes maintaining the report more difficult. In VFP 9, it's a lot easier.

Let's start with VFP 8. To specify different formatting for different values, you put multiple copies of the item into the same place on the report. Each copy is formatted differently and you use the Print When condition to indicate which copy to use at any given time.

Figure 1 shows a simple report based on the Northwind Products table. Items where stock is sufficient are shown in a normal typeface, items that need to be reordered are bold, and discontinued items are in italics.



Product Status Report

Product ID	Product Name	In Stock
1	Chai	39
2	**Chang**	**17**
3	**Aniseed Syrup**	**13**
4	Chef Anton's Cajun Seasoning	53
5	*Chef Anton's Gumbo Mix*	*0*
6	Grandma's Boysenberry Spread	120
7	Uncle Bob's Organic Dried Pears	15
8	Northwoods Cranberry Sauce	6

Figure 1 Formatting based on content—In this report, items that need to be reordered are in bold and discontinued items are in italics.

Figure 2 shows the report in the VFP 8 Report Designer. There's no visual clue as to how it works. To see what's going on, you have to double click one of the affected items. In the figure, you see the italic version. The Print When condition for those items is:

```
Discontinued
```

To see the normal and bold versions, you have to move the italic version out of the way, either by actually moving it or by choosing Send to Back from the Format menu. The Print When condition for the bold version is:

```
UnitsInStock<=ReorderLevel and not Discontinued
```

For the normal version, the Print When condition is:

```
UnitsInStock>ReorderLevel and not Discontinued
```

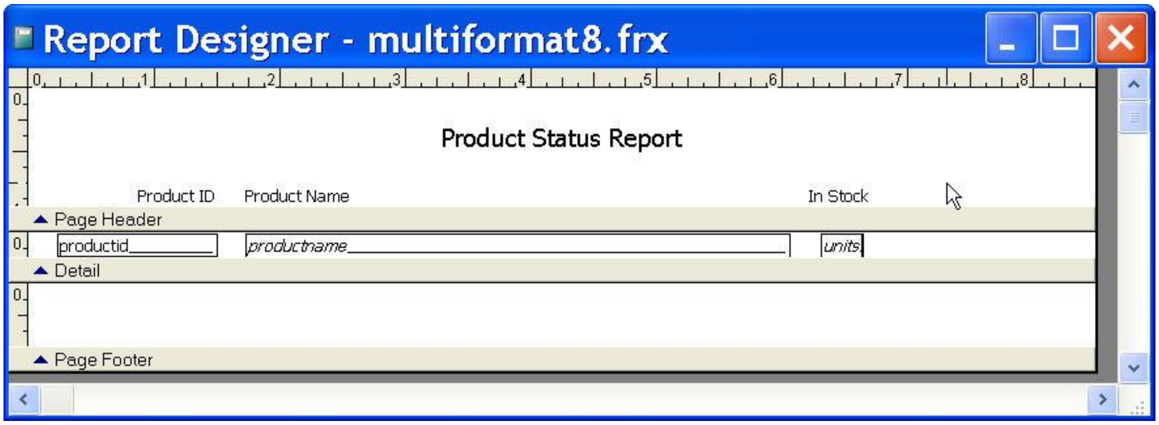


Figure 2 Setting up multiple formats—In VFP 8 and earlier, to specify formatting based on content, you have to use multiple copies of the same item and the Print When condition.

Using this technique, you can create as many different versions of the same item as you want. However, when something about an field changes, you have to be sure to change every version. (Of course, you also should ensure that the various Print When expressions are mutually exclusive.)

The same approach works in VFP 9, but you can create reports that are easier to maintain by taking advantage of a report listener, one of VFP 9's many new reporting features. A report listener is an object that responds to events when a report is running; it has the opportunity to intervene and modify the report's output.

While it's possible to write your own report listeners, VFP 9 comes with report listeners for a number of common tasks; most of them are in _reportListener.VCX in the FFC directory. One that you may have overlooked because it's stored elsewhere provides a variety of dynamic formatting options.

DynamicFormatting.PRG, found in the Samples\Solutions\Europa folder, contains EffectsListener, a report listener subclass offering an extensible approach to special formatting effects. As shipped, it offers effects for the forecolor and style of a report item. You can see this report listener in action by running the "Dynamic formatting using ReportListeners" example in the Solution Samples for VFP 9.

To use this report listener for your own applications, you don't have to understand its architecture. All you have to do is set up your report appropriately and instantiate the report listener before running your report.

The report listener takes advantage of another new VFP 9 feature, the ability to specify user data for an item. The listener checks the user data for each item and if it finds a string in a particular format, processes that string. The format for talking to the effects report listener is:

```
*:EFFECTS <effect name> = <expression>
```

<effect name> is the name of the formatting effect you want to change. The choices provided are "STYLE" and "FORECOLOR". (You can add additional choices, but how you do so is a subject for another column.)

<expression> is any expression that evaluates to a legal value for the specified effect. For forecolor, that's an RGB value. Table 1 shows the available values for style; it's an additive value, and it corresponds to the FontStyle field of an FRX.

Table 1. Setting font style—Add these values together to get the desired font style.

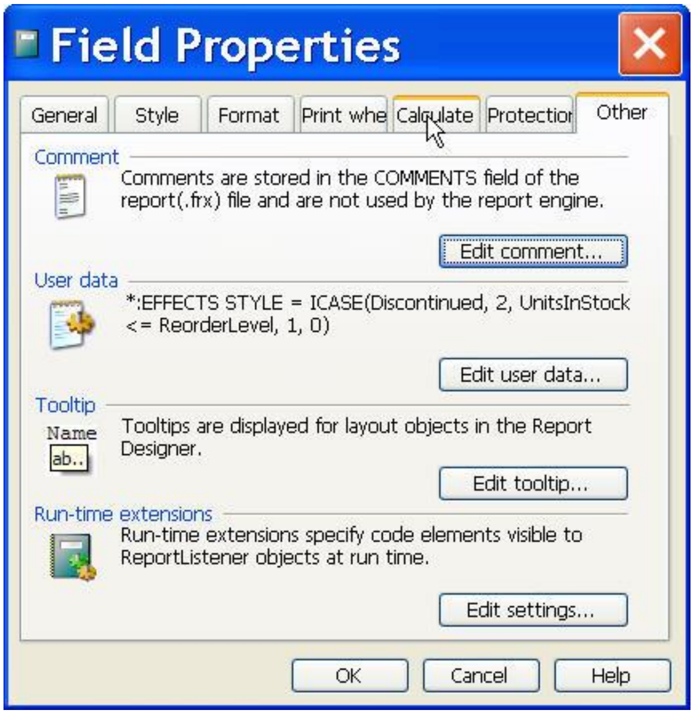
Value	**Meaning**
0	Normal
1	Bold
2	Italic

Value	Meaning
4	Underlined
128	Strikethrough

Normally, you'll use either IIF() or ICASE() in the expression to evaluate the data and decide which value to specify. For the product report, the following expression works:

```
ICASE(Discontinued, 2, UnitsInStock <= ReorderLevel, 1, 0)
```

Figure 3 shows the User tab of the Field Properties dialog with the complete effects directive for the Products report.



Comments are stored in the COMMENTS field of the report(.frx) file and are not used by the report engine.

*:EFFECTS STYLE = ICASE(Discontinued, 2, UnitsInStock <= ReorderLevel, 1, 0)

Tooltips are displayed for layout objects in the Report Designer.

Run-time extensions specify code elements visible to ReportListener objects at run time.

Figure 3. Specifying effects—Using the EffectsListener class, you specify formatting changes as User data.

The last step is hooking the report listener in when you run the report. You have to instantiate the report listener, set the output type and then the report. This code does the trick:

```
oListener = NEWOBJECT("EffectsListener", ;
   HOME(2)+"solution\europa\DynamicFormatting.prg")
oListener.OutputType = 1
```

```
REPORT FORM MultiFormat9 OBJECT oListener
```

This month's Professional Resource CD contains both versions of the report (MultiFormat8.FRX/FRT and MultiFormat9.FRX/FRT), as well as the program to run the report (RunMultiFormat.PRG).

–Tamar