



## **Color Coding Your Items**

How to Dynamically Change the Color  
of Objects on Your Template

# Contents

---

- Overview ..... 3**
- Using Layers to Change Object Color ..... 4**
  - How Layers Work in BarTender ..... 4
  - Conditional Printing ..... 4
  - Example ..... 4
- Using Visual Basic Script to Change Object Color ..... 8**
  - About Visual Basic Script ..... 8
  - Using VBScript with a Database ..... 8
  - Document-level Events ..... 8
  - Examples of Using VBScript to Dynamically Change Colors ..... 9
  - Support for Data-Sourced VBScript ..... 10
- Appendix A: Objects Properties for Dynamic Color ..... 11**
- Appendix B: BtColor Constants ..... 12**
- Related Documentation ..... 15**

## Overview

Many businesses use *color coding* to display and distinguish information by color. For example, the chemical industry uses color-coded labels for chemicals stored in the laboratory, so users can tell at a glance whether the stored chemical is hazardous, and what kind of hazard it poses. Electrical wiring is color-coded so that technicians can easily and safely identify the wire type and avoid accidents.

Using BarTender, you can use color printing to color-code your items, thus improving organization, inventory, scheduling and more. A fitness gym, for example, might want to print color-coded membership cards—one for each level of membership that they offer. A moving company might want to color-code moving box labels according to room.



Printing color-coded items in single-color batches can be time consuming. So BarTender allows you to print differently color-coded items *dynamically* (on the fly), at print-time. You only have to set up the template once, using layers and/or Visual Basic Script, to print items with colors that vary from one item to the next. You never have to go back and edit the template to get different color results.

## Using Layers to Change Object Color

One way to change the color of objects on your template at print-time is by combining layers with conditional printing. A *layer* is an object or group of objects that occupy a particular plane, and that can be stacked on top of each other.

### *How Layers Work in BarTender*

In BarTender, layers may contain one or more text objects, images, barcodes, shapes and/or lines included on the same plane. If you need to dynamically print multiple objects on your template at different times, you can put objects on different layers, and conditionalize the layers to print when certain conditions are met. You use each layer's **When to Print** dialog to determine when that layer will print.

### *Conditional Printing*

*Conditional printing* allows you to specify exactly when you want to print parts of your template.

Conditional printing is based around conditional operators. For example, your conditions for printing might be based on whether the conditionalized item **contains** or **does not contain** certain text, numeric values or images, or whether the conditionalized item is **less than**, **greater than**, or **equals** a set value.

In situations where you need to change colors for one or more objects on your template, you can conditionalize an entire layer rather than setting each object to conditionally print. To do this, on a new or copied layer, you would change the color of the object(s) you want to print differently, and conditionalize the layer(s) so that they print when conditions are met by the referenced database field or named data source value.

For more information about layers and conditional printing, refer to the following topics in the BarTender help system:

- [Overview of Layers](#)
- [Overview of Conditional Printing](#)
- [Building Conditional Expressions](#)

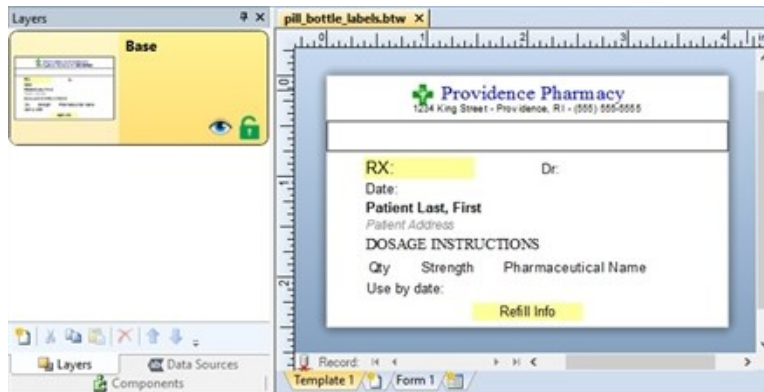
### *Example*

Suppose you are creating a prescription label. The pharmacy uses a color-coded stripe at the top of the bottle to quickly indicate the type of medication. Red is for cardiac medicines, blue is for analgesics and green is for antibiotics. You want the colored stripes to print based on the type of medication listed in the database. To implement this scenario, you need to:

1. Create a base layer with the label's information, adding the "stripe" to the design as a plain rectangle object.
2. Create a new layer (Layer 2) and paste the rectangle object onto the new layer.
3. Copy and paste the new layer twice more to create Layers 3 and 4.
4. Change the colors of the objects for each layer.
5. Conditionalize each layer by connecting it to a data source's value.

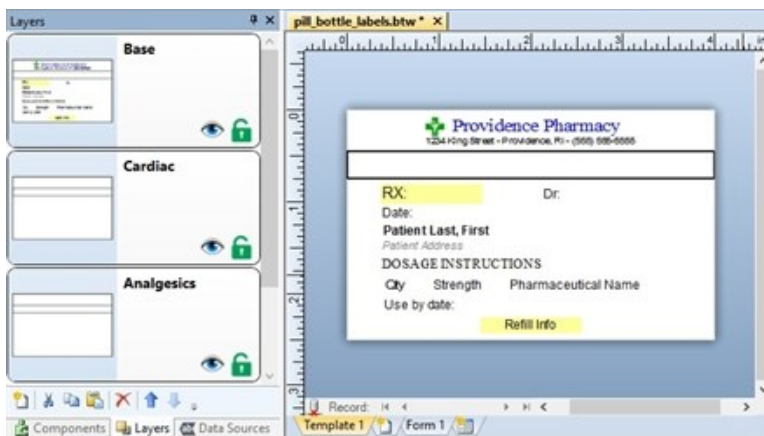
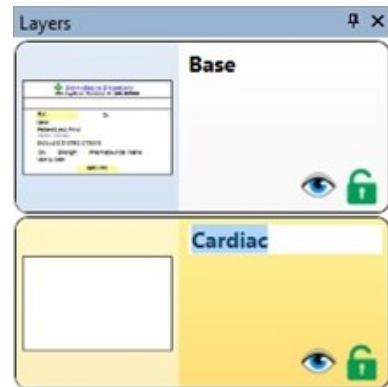
## Creating the Base Layer

1. Create your base layer, including the object(s) that you want to change color. In this example, we are using a rectangle shape as a "stripe".
2. Click the **Layers** tab at the bottom of the Toolbox. The **Layers** pane opens, displaying your design in "Layer 1."
3. To rename this layer, select **Rename** from the layer's context menu.



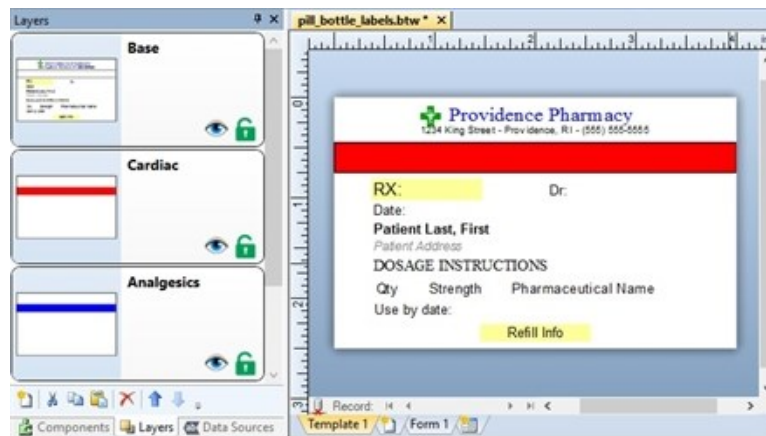
## Creating New Layers for Each Color Variation

1. From the **Layers** pane context menu, select **New Layer**. The **Layer Properties** dialog appears.
2. Click **OK** to close the dialog. You now have a blank Layer 2 in the **Layers** pane.
3. From the new layer's context menu, select **Rename** to assign a name to the layer.
4. Repeat steps 1-3 for the remaining number of color combinations.
5. On each of your new layers, copy and paste the object that you'd like to color code. In this case, it will be the rectangle object from our base layer.




## Setting Each Layer's Object Color

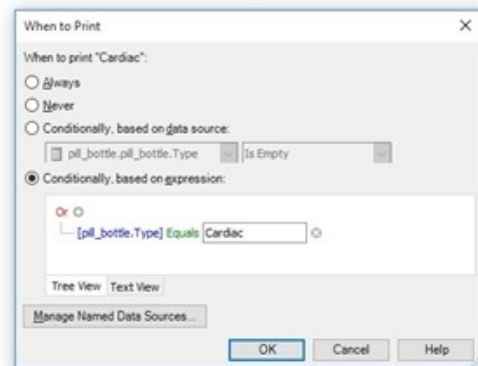
1. Double-click on the rectangle object in Layer 2 (or "Cardiac"). The **Box Properties** dialog opens.
2. In the Navigation Pane, click on **Box**.
3. Under **Line Properties**, use the **Color** dropdown list to change the color of the border.
4. Under **Fill Properties**, use the **Color** dropdown list to change the fill color of the object.
5. Repeat steps 1-4 to change the box colors on your subsequent layers.
6. When you've finished assigning colors, click **Close** to close the **Box Properties** dialog.



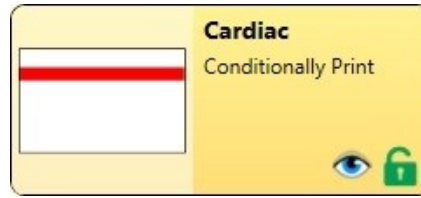
## Conditionaling the Layers

By creating a separate condition for each colored layer, you can control which colors you print and when. In this example, we will determine which layer to print based on the "Type" of medicine listed in our database.




1. In the **Layers** pane, double-click on Layer 2 (or "Cardiac") or select **Properties** from the layer's context menu. The **Layer Properties** dialog opens.
2. Click the **Print Options** tab.
3. At the right of the **Print When** field, click . The **When to Print** dialog opens.
4. Select **Conditionally, based on expression**.
5. Create an expression that is specific to the condition under which you want the layer to print, based on one or more data fields. For example, "[Type] Equals Cardiac" would cause the layer (which contains a red stripe) to print when a database field called "Type" returns the value "Cardiac".
6. Conditionalize the other two layers to print when "Type" equals "Analgesic" and "Antibiotic".



Note that layers that have been conditionalized will display the words "Conditionally Print".



The final output from this example will pull information from our database. When the medicine type matches "Cardiac," the stripe on the label will be colored red. Likewise, analgesics will be colored blue, and antibiotics will be colored green:

<p> <b>Providence Pharmacy</b> 1234 King Street - Providence, RI - (555) 555-5555</p> <hr/> <p><b>RX: 111683836A</b> Dr. KENDALL, J Date: 2015-10-15 Patient: Smith, Maxwell</p> <p>TAKE ONE TABLET EVERY DAY</p> <p>Qty: 5 Warfarin Sodium 5mg Use by date: 2016-10-15 **Refills require authorization**</p>	<p> <b>Providence Pharmacy</b> 1234 King Street - Providence, RI - (555) 555-5555</p> <hr/> <p><b>RX: 111683837A</b> Dr. WHITE, L Date: 2015-10-15 Patient: Willis, Katherine</p> <p>TAKE ONE TABLET EVERY SIX HOURS AS NEEDED FOR PAIN</p> <p>Qty: 20 APAP/Codeine 300/30mg Use by date: 2016-10-15 REF: 5</p>
<p> <b>Providence Pharmacy</b> 1234 King Street - Providence, RI - (555) 555-5555</p> <hr/> <p><b>RX: 111683838A</b> Dr. CARLSON, M Date: 2015-10-15 Patient: Doe, Jane</p> <p>TAKE ONE CAPSULE THREE TIME S DAILY</p> <p>Qty: 30 Amoxicillin 500mg Use by date: 2016-10-15 **No refills remaining**</p>	

## Using Visual Basic Script to Change Object Color

You can set up your BarTender document to execute Visual Basic Script (VBScript), which will change the color of an object in response to certain document-level events. VBScript can be applied to any template object, including text, barcode and shape objects.

You can create a VBScript for your document using the Script Editor. For more information, refer to the [The Script Editor](#) topic in the BarTender help system.

**NOTE:** VBScript is an advanced method of managing your template, and should be used by those who have some technical knowledge of scripting.

### About Visual Basic Script

Visual Basic Script is a subset of Microsoft's Visual Basic programming language. It is specifically designed to add functionality to existing programs, rather than to write programs themselves.

VBScripts for changing color are not terribly complex. You can use VBScript to programmatically modify the color of template objects at different times throughout the document design and print process. To accomplish this dynamically, you need to create a database, and connect it to your BarTender document.

### Using VBScript with a Database

When you connect your BarTender document to a database, you are able to access data contained within the data source. The data can be used to do a variety of things, including changing the color of a template object. By linking a specific database field to an object on the template, the value of the object is replaced with the data contained within the database. Additionally, you can refer to colors stored in database fields, or change the color based on the data inside your database.

For more information on using databases in BarTender, refer to the following topics in the BarTender help system:

- [Reading Data from Databases](#)
- [Referencing Database Fields](#)

### Document-level Events

These are events that occur on the document level, such as OnPrintStart or OnPrintCancel. From the Script Assistant, you can select a document-level event that will trigger your VBScript to execute and change an object's color.

One of the easiest document-level events to use for changing color is "OnNewRecord," which causes the script to execute each time a new record is read from the database.

Before you can use VBScript with a BarTender document, you must first enable the use of VBScript for that document:



1. From the **BarTender Document Options** dialog, click the **Scripting** tab.
2. Select (check) **Enable document level script events**.
3. Select the event you wish to use.
4. Click the **Edit** button to open the Script Editor.
5. Select the desired event from the Script pane.
6. Enter your custom script into the Editor pane.

## **Examples of Using VBScript to Dynamically Change Colors**

Let's look at some examples of how you might dynamically change an object's color using VBScript.

### **Example 1**

Suppose you have a business with membership levels, and you want to change the color of a text object to match each customer's membership level. To do this, you would reference the data source that specifies membership type. In this example, we'll call that data source `MembershipType`. To make a text object turn gold, silver or blue depending on membership level, you would enter script similar to this one in the Script Composition Area of the Script Assistant:

```
ReferenceField("MembershipType")
Set Notice = Objects("Text 1")
If (Field("MembershipType") = "Gold") Then
    Notice.TextColor = btColor.Gold
Else
    If (Field("MembershipType") = "Silver") Then
        Notice.TextColor = btColor.Silver
    Else
        If (Field("MembershipType") = "Platinum") Then
            Notice.TextColor = btColor.Blue
        Else
            Notice.TextColor = btColor.Black
        End If
    End If
End If
```

### **Example 2**

If you have a field in your database that contains the name of a color, you can use VBScript to set the color of an object.

For instance, if you had a database field called `ConditionColor`, and database records that contain color names like `BtColor.Gold` and `BtColor.Silver`, you could use VBScript to determine when a certain color prints. If you wanted to change the color of a box shape object, your script might look something like this:

```
ReferenceField("ConditionColor")
Objects("Box 1").FillColor = Eval(Field("ConditionColor"))
```

- For a list of objects properties for dynamic color (such as `TextColor` and `BarCodeColor`),

refer to [Appendix A](#).

- For a list of BtColor constants, refer to [Appendix B](#).

### ***Support for Data-Sourced VBScript***

You can feed VBScript into BarTender from external sources, such as a database or a program that is controlling BarTender. For example, you could have a field in a database contain VBScript code. Then, when BarTender reads the data record that contains that field, VBScript code in BarTender can use the VB “Execute” statement or the “Eval” function to execute the imported VBScript code.

## Appendix A: Objects Properties for Dynamic Color

The Object object in BarTender's library of VBScript objects represents an object in a document, such as a barcode or text object. This table lists color-related properties that you can change in VBScript for each type of object. For a complete of object properties, refer to the [Template Objects \(Object Object\)](#) topic in the BarTender help system.

For a list of btColor constants, see [Appendix B](#).

**NOTE:** All Object properties are to be used in document scripts, accessed from the **BarTender Document Options** dialog. They cannot be used in data source scripts or transform scripts.

### Examples:

`MyTextObject.TextColor = btColor.Blue` would turn the *text* in a text object blue.

`MyShapeObject.FillColor = btColor.Black` would set a shape's *fill color* to black.

Name	Description
BarCodeColor	Sets or returns a barcode object's bar color.
FillColor	Sets or returns any shape's fill color.
LineColor	Sets or returns the color of a line object.
TextBackgroundColor	Sets or returns the background color of an object's text.
TextColor	Sets or returns the color of an object's text.

## Appendix B: BtColor Constants

When setting the color of an object, you must reference a BtColor constant.

### Example:

`MyTextObject.TextColor = btColor.LimeGreen` would turn the text in a text object *lime green*.

The following constants are available:

Color	Value	Color	Value
AliceBlue	#F0F8FF	LightSalmon	#FFA07A
AntiqueWhite	#FAEBD7	LightSeaGreen	#20B2AA
Aqua	#00FFFF	LightSkyBlue	#87CEFA
Aquamarine	#7FFFD4	LightSlateGray	#778899
Azure	#F0FFFF	LightSteelBlue	#B0C4DE
Beige	#F5F5DC	LightYellow	#FFFFE0
Bisque	#FFE4C4	Lime	#00FF00
Black	#000000	LimeGreen	#32CD32
BlanchedAlmond	#FFEBCD	Linen	#FAF0E6
Blue	#0000FF	Magenta	#FF00FF
BlueViolet	#8A2BE2	Maroon	#800000
Brown	#A52A2A	MediumAquamarine	#66CDAA
BurlyWood	#DEB887	MediumBlue	#0000CD
CadetBlue	#5F9EA0	MediumOrchid	#BA55D3
Chartreuse	#7FFF00	MediumPurple	#9370DB
Chocolate	#D2691E	MediumSeaGreen	#3CB371
Coral	#FF7F50	MediumSlateBlue	#7B68EE
CornflowerBlue	#6495ED	MediumSpringGreen	#00FA9A
Cornsilk	#FFF8DC	MediumTurquoise	#48D1CC
Crimson	#DC143C	MediumVioletRed	#C71585
Cyan	#00FFFF	MidnightBlue	#191970
DarkBlue	#00008B	MintCream	#F5FFFA
DarkCyan	#008B8B	MistyRose	#FFE4E1
DarkGoldenrod	#B8860B	Moccasin	#FFE4B5
DarkGray	#A9A9A9	NavajoWhite	#FFDEAD

DarkGreen	#006400	Navy	#000080
DarkKhaki	#BDB76B	OldLace	#FDF5E6
DarkMagenta	#8B008B	Olive	#808000
DarkOliveGreen	#556B2F	OliveDrab	#6B8E23
DarkOrange	#FF8C00	Orange	#FFA500
DarkOrchid	#9932CC	OrangeRed	#FF4500
DarkRed	#8B0000	Orchid	#DA70D6
DarkSalmon	#E9967A	PaleGoldenrod	#EEE8AA
DarkSeaGreen	#8FBC8F	PaleGreen	#98FB98
DarkSlateBlue	#483D8B	PaleTurquoise	#AFEEEE
DarkSlateGray	#2F4F4F	PaleVioletRed	#DB7093
DarkTurquoise	#00CED1	PapayaWhip	#FFefd5
DarkViolet	#9400D3	PeachPuff	#FFDAB9
DeepPink	#FF1493	Peru	#CD853F
DeepSkyBlue	#00BFFF	Pink	#FFC0CB
DimGray	#696969	Plum	#DDA0DD
DodgerBlue	#1E90FF	PowderBlue	#B0E0E6
FireBrick	#B22222	Purple	#800080
FloralWhite	#FFFAF0	Red	#FF0000
ForestGreen	#228B22	RosyBrown	#BC8F8F
Fuchsia	#FF00FF	RoyalBlue	#4169E1
Gainsboro	#DCDCDC	SaddleBrown	#8B4513
GhostWhite	#F8F8FF	Salmon	#FA8072
Gold	#FFD700	SandyBrown	#F4A460
Goldenrod	#DAA520	SeaGreen	#2E8B57
Gray	#808080	Seashell	#FFF5EE
Green	#008000	Sienna	#A0522D
GreenYellow	#ADFF2F	Silver	#C0C0C0
Honeydew	#F0FFF0	SkyBlue	#87CEEB
HotPink	#FF69B4	SlateBlue	#6A5ACD
IndianRed	#CD5C5C	SlateGray	#708090
Indigo	#4B0082	Snow	#FFFAFA
Ivory	#FFFFF0	SpringGreen	#00FF7F

Khaki	#F0E68C	SteelBlue	#4682B4
Lavender	#E6E6FA	Tan	#D2B48C
LavenderBlush	#FFF0F5	Teal	#008080
LawnGreen	#7CFC00	Thistle	#D8BFD8
LemonChiffon	#FFFACD	Tomato	#FF6347
LightBlue	#ADD8E6	Turquoise	#40E0D0
LightCoral	#F08080	Violet	#EE82EE
LightCyan	#E0FFFF	Wheat	#F5DEB3
LightGoldenrodYellow	#FAFAD2	White	#FFFFFF
LightGray	#D3D3D3	WhiteSmoke	#F5F5F5
LightGreen	#90EE90	Yellow	#FFFF00
LightPink	#FFB6C1	YellowGreen	#9ACD32

## Related Documentation

### BarTender Help System

- [Changing an Object's Color](#)
- [Overview of Layers](#)
- [Overview of Conditional Printing](#)
- [Visual Basic Scripting in BarTender](#)

