

Conditional Formatting: Adding Customized Rules to Excel 2007

Overview

One of the primary goals of the Excel 2007 team when they were planning updates to conditional formatting was to make a greater number of scenarios possible without the need to write formulas. Let me briefly explain what I mean. Excel 2003 gives you two choices when creating conditional formats: you can either set up numerical comparisons (for example, this cell is greater than a value, less than a value, equal to a value, etc.), or you can use a formula to specify conditions. While formulas are powerful and flexible, they take more time to implement and typically require a deeper knowledge of the product to set them up. For example, it can be challenging to set up conditional formatting rules like "highlight the top 20% of this range."

When doing research, the team found that there were a set of very common things that users wanted to do with conditional formatting, many of which required writing formulas, although not all were obvious. Based on these findings, a number of new conditional formatting rules are added in Excel 2007 to address the most common customer scenarios. These rules are broken down into a few categories in the new user interface (UI) to make it easier to find the correct rule type. Specifically, without writing formulas, you can conditionally format cells:

- With data bars, color scales, or icons based on the numeric value in the cell, percentages, percentiles, or a formula. For more information on these features, see All About Conditional Formatting in Excel 2007.
- With conditional operators such as "Containing," "Not Containing," "Beginning With," or "Ending With" specific text. For example, "highlight parts containing certain characters in a parts catalog."
- Containing dates that match dynamic conditions like "Yesterday," "Today," "Tomorrow," "In the Last 7 Days," "Last Week," "This Week," "Next Week," "Last Month," "This Month," and "Next Month." For example, "highlight all items dated yesterday." A big plus with these types of rules is that Excel 2007 handles calculating the date based on the system clock, so you do not need to worry about updating the rules for conditions like leap years.
- To test whether they are blank or not blank.
- To test whether they have errors or do not have errors.
- To see if the data is in the top **n** of a selected range (where **n** is some number) OR that the data is in the top **n** percent of a selected range (again, where **n** is adjustable). For example, "highlight the top 10 investment returns in a table of 1,000 investments."
- That have the bottom **n** values or cells that are the bottom **n** percent of a selected range.

- That are "Above Average," "Below Average," "Equal To or Above average," "Equal to or Below average," 1 standard deviation above, 1 standard deviation below, 2 standard deviations above, 2 standard deviations below, 3 standard deviations above, and 3 standard deviations below a selected range.
- That contain duplicate values or, conversely, cells that contain unique values.
- That are based on comparisons between two columns in tables. For example, "highlight values where the values in the Actual Sales column are less than in the Sales Target column." When working with tables, it is now easy to format the entire row based on the results of a condition.

Other benefits of the changes to conditional formatting in Excel 2007 are the ability to specify more than three conditions, to reorder conditions, and to have more than one condition resolve to **True**.

An Improved User Interface

Let's take a look at the more advanced conditional formatting UI and how it allows you to address these and other conditions. First, let's look at the **Conditional Formatting Rules Manager** dialog box:

1. Start Excel 2007.
2. On the **Home** tab, click **Conditional Formatting** and then click **Manage Rules**. The **Conditional Formatting Rules Manager** is shown (see Figure 1):

Figure 1. The Conditional Formatting Rules Manager dialog box

The **Conditional Formatting Rules Manager** dialog box is the one-stop place to view and manage all of the conditional format rules in a workbook. Using this dialog, you can:

- View the rule condition, see the range it is applied to, and the formatting applied by the rule. The **Show formatting rules for** list at the top of the dialog box allows you to change the scope of the rules. You can view rules for the selected cells, the current worksheet, or other worksheets in the workbook.
- Add new rules with the **New Rule** button. (More on this shortly.)
- Change the order in which rules are evaluated. This is accomplished by using the up and down arrow buttons.
- Edit existing rules with the **Edit Rule** button. The **Edit Rule** UI is the same as the **New Rule** UI.

- Delete rules with the **Delete Rule** button.
- Control whether more than one rule can evaluate to **True**. (We'll discuss this in more detail later in this column.)

When you press **New Rule** on the **Conditional Formatting Rules Manager** dialog box, or when you select **More Options** from the conditional formatting gallery, you see the **New Formatting Rule** dialog box (see Figure 2):

Figure 2. The New Formatting Rule dialog box

This dialog allows you to add new conditional formats to the selected range. Through this UI, you can add all the conditional formats that are available through the Ribbon as well as a set of additional, slightly more sophisticated conditional formats. In addition, you can change the different settings and parameters on any conditional format rule. The conditional formatting options are grouped into a few categories that are listed in the top of the **New Formatting Rule** dialog box under **Select a Rule Type**. The bottom part of the dialog contains the **Edit the Rule Description** section, which changes based on the rule type selected. Here is a summary of the categories:

- **Format all cells based on their values.** Use this to create a data bar, 2-color or 3-color color scale, or icon set rule.
- **Format only cells that contain.** Use this option to create the Excel 2003-style rules and more (format cells greater than, less than, greater than or equal to, less than or equal to, equal to, not equal to, between, not between). This is also the entry point to create rules of type: specific text, date occurring, blanks, no blanks, errors, no errors.
- **Format only top or bottom ranked values.** Use this to create top **n**, top **n%**, bottom **n**, bottom **n%** rule types.
- **Format only values that are above or below average.** Use this to create above, below, equal or above, equal or below, 1 or 2 or 3 standard deviation above, or 1 or 2 or 3 standard deviation below rule types.
- **Format only unique or duplicate values.** Use this to create rules that format unique or duplicate values.
- **Use a formula to determine which cells to format.** Use this to create Excel 2003-style rules where you can enter a formula to determine whether a format should be applied.

Finally, as I touched on briefly in some of my previous columns, there are significant improvements in the infrastructure of conditional formatting. For example, Excel 2007 introduces the following changes:

- **Number of conditions on a range is no longer limited by number.** In Excel 2003, you could create conditional formatting rules with three conditions. In Excel 2007, you are no longer limited by number; you are only limited by system memory.
- **Multiple conditions evaluating to true.** In Excel 2003, you could apply only one conditional format to a given cell. If more than one conditional format evaluated to true, then only the format associated with the first conditional format was applied.

Excel 2007 now allows you to have multiple conditional formats on a cell applied if more than one condition evaluates to **True**. For example, assume you have one conditional format that makes font style bold when true, and another that makes the cell background color red when true. If both conditions are true, you see both formats applied (bold text and red cell background color). When there are conflicts when applying rules, (for example, apply red font AND green font), the first rule wins. You can trigger this effect on or off for any rule by using the checkboxes on the **Conditional Formatting Rules Manager** dialog box.

Controlling Rules Programmatically

In most scenarios, anything you can do in the UI, you can also do programmatically. The Excel 2007 object model supports all conditional formatting functionality that is supported in the UI. This includes creating, editing, or deleting rules, as well as changing the priority of rules.

As those of you who have written conditional formatting by using Microsoft Visual Basic for Applications (VBA) in previous versions of Excel are aware, the **FormatConditions** collection is derived from the **Range** object. The following examples demonstrate how this new functionality is exposed in the **FormatConditions** collection.

Creating a Rule

You can create the new conditional formatting features introduced in Excel 2007 (Databars, Color Scales, Icon Sets, Top **n**) using the **Add** method in the **FormatConditions** collection. For example, to add a **Databar** object, use:



```
VB C# C++ F# JScript
Range("A1:A5").FormatConditions(1).BarColor.ColorIndex = 3
```

To edit the rule, by using an index into the **FormatConditions** collection, you then modify the correct properties. For example, to change the bar color, use:

```
VB C# C++ F# JScript
Range("A1:A5").FormatConditions(1).BarColor.ColorIndex = 3
```

In this statement, the number **1** indexes the first rule on the range.

Editing the Priority

Excel 2007 introduces the idea of rule priorities to support multiple conditions on a range. The priority determines the order of application of the rule. In the object model, you have the **Priority** property on the **FormatConditions** object. This property is tracked at the worksheet level. For example, to verify the priority of a rule, use:

```
VB C# C++ F# JScript
Range("A1:A5").FormatConditions(1).Priority
```

To make this rule the lowest priority:

```
VB C# C++ F# JScript
Range("A1:A5").FormatConditions(1).SetLastPriority
```

To assign a specific priority:

```
VB C# C++ F# JScript
Range("A1:A5").FormatConditions(1).Priority = 3
```

Note that if you had three rules, setting the priority to **3** and using **SetLastPriority** has the same effect.

Deleting the Rule

You can delete a specific rule by using its index and then calling the **Delete** method:

```
VB C# C++ F# JScript
Range("A1:A5").FormatConditions(1).Delete
```

To delete all rules in the specific range, call the **Delete** method on the **FormatConditions** collection.

```
VB C# C++ F# JScript
Range("A1:A5").FormatConditions.Delete
```