Quick answers to common problems

Over 100 simple and incredibly effective recipes to help transform your static business data into exciting dashboards filled with dynamic charts and graphics

SAP BusinessObjects Dashboards 4.1 Cookbook

SAP BusinessObjects Dashboards 4.1 Cookbook

The widely used SAP BusinessObjects Dashboards software lets you transform data from any source into interactive dashboards.

The recipes begin by covering best practices for using the SAP BusinessObjects Dashboards spreadsheet. The book then guides you through the exploration of various data visualization components and dashboard interactivity, as well as using alerts, dashboard connectivity, publishing the dashboard, and making use of the aesthetics of the dashboard. Finally, the recipes conclude by considering the most important add-ons available for SAP BusinessObjects Dashboards, performance tuning, and tips on increasing development productivity.

Who this book is written for

If you are a developer with a good command and knowledge of creating dashboards, but are not yet an advanced user of SAP BusinessObjects Dashboards, then this is the perfect book for you. Prerequisites include a good working knowledge of Microsoft Excel as well as knowledge of basic dashboard practices.

What you will learn from this book

- Become fully equipped with best practices when using the SAP BusinessObjects Dashboards spreadsheet
- Present data using a wide variety of data visualization components
- Discover how to make dashboard components interactive for an enhanced user experience
- Take advantage of Dynamic Visibility features
- Connect your dashboard to live data sources
- Export and publish the SAP BusinessObjects Dashboards model into several environments
- Improve the performance of your dashboards and increase your productivity as a dashboard developer


Foreword by Dr. Bjarne Berg, CIO, Comerit Inc. and professor, SAP University Alliance at Lenoir-Rhyne University.
In this package, you will find:

- The author’s biography
- A preview chapter from the book, Chapter 1 'Staying in Control'
- A synopsis of the book’s content
- More information on SAP BusinessObjects Dashboards 4.1 Cookbook

About the Authors

David Lai is a SAP BusinessObjects consultant and specializes in data visualization and data warehousing. He graduated with a degree in computer engineering from the University of Toronto. He has a passion for providing organizations with smart Business Intelligence solutions that encompass best practices and techniques. In addition, he is an active contributor to the community and provides his knowledge in best practices and solutions.

He started his Business Intelligence blog (http://www.davidlai101.com/blog) in 2008, where he provides tips, tricks, and best practices for Xcelsius and BusinessObjects-related material. He is a bronze-level contributor to the SAP community network, has presented at SAP InsideTrack, and provides BusinessObjects training to students.

Aside from work, David enjoys physical activities such as weight training, basketball, volleyball, and skiing. He also has a strong passion for Latin dance.

David is the owner of Xinfinity Solutions, where he provides consulting services as a SAP BusinessObjects consultant. He has worked for a long list of satisfied clients in various industries.
**Xavier Hacking** is an SAP BI specialist from Eindhoven, the Netherlands, and works as a consultant for Interdobs. He has a master's degree in industrial engineering and management science from the Eindhoven University of Technology. He has worked with a wide range of products from the SAP Business Intelligence portfolio, including SAP BW, SAP BusinessObjects Dashboards, Design Studio, Web Intelligence, Crystal Reports, and the SAP BusinessObjects BI Platform. His main focus is dashboard development within the various SAP environments.

Xavier is the coauthor of the first edition of this book and the book *Getting Started with SAP BusinessObjects Design Studio*, *SAP Press*. He writes for the SAP Experts BI Hub and has his own blog related to Business Intelligence at HackingSAP.com (http://HackingSAP.com/). You can follow Xavier on Twitter (http://twitter.com/xjhacking).
SAP BusinessObjects Dashboards 4.1 Cookbook

SAP BusinessObjects Dashboards 4.1 (formerly known as Xcelsius) is a desktop dashboard and visualization solution that is a core part of SAP BusinessObjects BI 4.1. Once a user creates a dashboard model, it can be deployed in Flash format to web portals, SAP environments, the SAP BusinessObjects BI Platform, and desktop applications such as PowerPoint, Word, or PDF.

For dashboard designers/developers, SAP BusinessObjects Dashboards allows for rapid development of data visualizations through a flexible and easy-to-use graphical user interface.

Using SAP BusinessObjects Dashboards, we can accomplish the following:

► Create interactive dashboards that have a wow factor unlike other dashboard tool competitors.
► Connect dashboards to over 10 different types of data connections.
► Integration and interoperability with existing SAP BusinessObjects BI content.
► We can embed our dashboards into a variety of different formats to allow for convenient sharing between users.
► Ability to create custom add-on components using the SAP BusinessObjects Dashboards SDK.

SAP BusinessObjects Dashboards in its original conception was a way to build visualizations and dashboards using Excel data. That is also where the original name Xcelsius comes from. Over the past decade, BusinessObjects has enhanced Xcelsius into a fully featured enterprise-ready dashboard solution that works with any data source.

After the acquisition of BusinessObjects by SAP, the mission to make Xcelsius a dashboard product to serve all its customers (beyond being just a personal productivity tool) continued. The BI market and SAP customers were also demanding an enterprise dashboard solution for the types of projects they were using Xcelsius for, for example, dashboards for thousands of users using large data warehouses as a datasource. The name Xcelsius was no longer meaningful or relevant.

By changing the name Xcelsius to SAP BusinessObjects Dashboards, SAP is showing its commitment to delivering a solution that serves the needs of all BI customers as well as aligning the name to the product’s growing capabilities and roadmap.

The SAP BusinessObjects Dashboards portfolio consists of several different packages (see the edition comparison later in the preface). In this book, we use SAP BusinessObjects Dashboards to refer to the tool itself.
What This Book Covers

Chapter 1, Staying in Control, presents you with best practices for using the SAP BusinessObjects Dashboards spreadsheet, the data model, and connections with the components on the canvas.

Chapter 2, Data Visualization, presents you with recipes on how to use different components such as charts, tables, and graphs to visualize data on the dashboard.

Chapter 3, From a Static to an Interactive Dashboard, shows you how to add interactivity to your dashboards by adding selectors, maps, buttons, drilldowns, and so on.

Chapter 4, Dynamic Visibility, shows you how to make components visible/invisible and provides scenarios where dynamic visibility becomes useful.

Chapter 5, Using Alerts, contains examples of different ways of showing alerts on a dashboard.

Chapter 6, Advanced Components, provides recipes on SAP BusinessObjects Dashboards' more advanced components.

Chapter 7, Dashboard Look and Feel, teaches you how to tweak the visuals and user experience of the dashboard by customizing the look of components.

Chapter 8, Dashboard Data Connectivity, talks about the various options to connect a dashboard to external data sources.

Chapter 9, Exporting and Publishing, contains recipes on how to export SAP BusinessObjects Dashboards into different environments.

Chapter 10, Top Third-party Add-ons, contains an introduction to some of the most useful third-party add-ons for SAP BusinessObjects Dashboards.

Chapter 11, Performance Tuning, teaches you how to improve the performance of your dashboards by tweaking the spreadsheet and optimizing the data sources' connection setup.

Chapter 12, Increasing Productivity, discusses various development best practices and tips to save precious development time.
Appendix A, *Real-world Dashboard Case Studies*, demonstrates how to implement various techniques covered in this book by creating two applications: a calculator that displays monthly payments of a mortgage and a sales/profit dashboard that displays the sales or profit of each state on a map.

Appendix B, Additional Resources – Supported Excel Functions and System/Software Requirements, lists some helpful online resources for further reference and some useful Microsoft Excel functions supported by SAP BusinessObjects Dashboards.

Appendix C, The Future of Dashboarding with SAP Design Studio, introduces you to a new SAP tool: Design Studio. A comparison with SAP BusinessObjects Dashboards is made and the future roadmap for this tool is shared.
1

Staying in Control

In this chapter, we will begin with the introduction of SAP BusinessObjects Dashboards and understanding the dashboard workspace.

In this chapter, we will cover the following recipes:

- Using the Object Browser
- Searching for components
- Grouping the canvas components
- Making the spreadsheet more readable with colors
- Making the spreadsheet more readable with comments
- Making the spreadsheet more readable with borders
- Using named ranges
- Copying the format of one cell to another cell or range
- Debugging the spreadsheets
- Navigating between worksheets

Introduction

During the development of a typical SAP BusinessObjects dashboard, the number of components as well as the Excel spreadsheet data bindings can become quite complex. To prevent us from getting lost in an unmanageable chaos of components, interactions, bindings, and several different Excel functionalities, a structured approach should be followed right from the start of dashboard development. Also, we should use the advantages Excel gives us to build an optimal data model that is easy to read and maintain.
Understanding the dashboard workspace

Before you begin designing dashboards, it is important that you understand the workspace. The workspace area is illustrated as follows:

Let's have a look at some of the important sections of the dashboard workspace as depicted in the preceding screenshot:

- **Menu bar and toolbar (1):** SAP BusinessObjects Dashboards consists of a menu bar as well as five toolbars that are used to help develop dashboards.
- **Dashboard canvas (2):** This is where the dashboard presentation is built. Users drag and drop dashboard objects here.
- **Embedded spreadsheet (3):** This embedded spreadsheet is used to associate the dashboard objects with data. More information on tips and best practices when building your spreadsheet models can be found later in this chapter.
- **The Components window (4):** Users can drag and drop dashboard components from the Components window onto the dashboard canvas.
- **The Object Browser (5):** All objects existing in the dashboard model can be found in the Object Browser. It provides a way to easily access your dashboard objects. For more instructions on using the object browser, please refer to the following recipe, Using the Object Browser.
- **The Properties window (6):** This contains settings and formatting options for a selected component.
The **Query Browser** (7): This allows users to create and manage dashboard queries. For more information, please refer to the recipe *Using the Query Browser* in Chapter 8, *Dashboard Data Connectivity*.

The **Mobile Compatibility** window (8): This provides mobile compatibility information on all the objects found in the dashboard model. For more information, please refer to the recipe *Going mobile* in Chapter 9, *Exporting and Publishing*.

## Using the Object Browser

The **Object Browser** has a number of features which come in very handy during the development of a complex dashboard. In this section, we will discuss hiding, locking, and ordering of components.

### Getting ready

Drag several components to the canvas.

### How to do it...

1. Go to the **Object Browser**.
2. Click on the dot in the first of the two columns on the right side of the **Object Browser** for the component that you want to hide. The dot turns into a checkmark. As you can see, the component now disappears from the canvas.

   ![Object Browser](image)

3. Now click on the dot on the right for any of the components.
4. Try to move the component or make any other change to it.
5. As you'll see, the component is completely locked and doesn't change.
6. Make sure some of your components are on top of each other in the canvas. We now want to use the **Object Browser** to rearrange these overlapping components.
7. Select the component in the **Object Browser** that is on top of the list.
8. Click the arrow down button in the **Object Browser** multiple times until the component is on top of all the other components.
9. As you can see, the component shifts all the way to the top.
How it works...

As we have seen in this recipe, we can hide components and/or groups of components, which will make your life easier if you are using a lot of overlapping components. By checking **Hide** for some components, you won't be bothered by these and you can work with the others that are unhidden.

There is one thing you should keep in mind. If you hide a component that is part of a group but the group itself is unhidden, the complete group will still be movable and its properties will be changeable.

Also, we saw that we can lock one or more components or groups of components. Doing this makes it impossible to select these components, so it won't be possible to move, change, or do anything else with them. In this way you can be sure you won't accidentally alter these components.

Hiding and/or locking a component from the **Object Browser** only hides and/or locks that component during the development of a dashboard. When you preview or execute the dashboard, the component will appear again and function normally.

Finally, we changed the order of components on the canvas. This is an important feature when we are using overlaying components in our dashboard.

To move a component on top of all other components, you can also right-click on it and select the **Bring To Front** option. **Send To Back** will move the component all the way down. The options **Bring Forward** and **Send Back** do the same as the arrows in the **Object Browser**: they move the component one step up or down at a time.

Searching for components

The ability to do a search for components from the **Object Browser** is a helpful feature new to SAP BusinessObjects Dashboards 4.1. You can perform a search by either name or component type. This comes in handy when you have a lot of objects on the dashboard that are inside groups or canvas containers.

Getting ready

Make sure you have a dashboard that contains a set of components.
Chapter 1

How to do it...

1. Open up the Object Browser window.
2. In the top-left input box, type in the object name or component type that you are searching for. You will see that your search will filter the objects accordingly.

![Image of Object Browser]

How it works...

As you can see, the search functionality is a useful feature as you can easily find objects by their name.

The search functionality works best when you have named all your objects properly.

Grouping the canvas components

Canvas components can be grouped with one or more other components.

Getting ready

Drag several components to the canvas.
How to do it...

1. Select the components that you want to group. You can do this by either selecting multiple components from the dashboard canvas by dragging the mouse over them, or clicking the components one-by-one while holding the Ctrl button on your keyboard.

2. Right-click anywhere on the canvas and select Group from the context menu. You can also use the shortcut Ctrl + G to group these components. As you can see, the components are now a group with a common border.
If you double-click on the grouped object you will see a **Common** tab where you can set the **Dynamic Visibility** and **Entry Effect**.

3. If you create a lot of groups of components, we advise that you name these groups to prevent you from getting lost and confused during the dashboard development. First go to the **Object Browser**.

4. Select the group you want to rename.
5. Double-click the group or right-click and select **Rename** from the context menu.
6. Type in the new name for this group.
How it works...

When your dashboard gets more complex, not only will the data model in the spreadsheet grow, the number of components used on the canvas will also increase. Using groups to differentiate the canvas components from each other is a great way to stay in control of your dashboard.

Name the groups as something that can be visually matched to your dashboard, such as a section heading. In that way, someone who did not originally develop the dashboard can quickly see which set of components the object group refers to.

Making the spreadsheet more readable with colors

The more complex a dashboard gets, the more clogged the spreadsheet might get with data, Excel formulas, and other usages. To make clear what the exact purpose of a cell is, we color code them to make things more clear.

Getting ready

You need a basic SAP BusinessObjects Dashboards file containing a few components in the canvas with some bindings to the data model in the spreadsheet.

How to do it...

1. Go to your data model in the spreadsheet.
2. Select the cell(s) you want to color.
3. Click on the Fill Color button in the Font section of the Home tab and select the desired color.
4. Color the cells that have dynamic visibility values in orange.

5. Color the cells with input values from canvas components in yellow. In the following screenshot, row A3:N3 is used as the destination range for a drill down from a chart.

6. Color the cells that will be filled with data from an external data source in blue.

7. Color the cells with Excel formulas in green.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dynamic Visibility:</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Product 1</td>
<td>100</td>
<td>95</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>100</td>
<td>95</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>1110</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Monthly Product Sales numbers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>JAN</td>
<td>FEB</td>
<td>MAR</td>
<td>APR</td>
<td>MAY</td>
<td>JUN</td>
<td>JUL</td>
<td>AUG</td>
<td>SEP</td>
<td>OCT</td>
<td>NOV</td>
<td>DEC</td>
<td>TOTAL</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Product 1</td>
<td>100</td>
<td>95</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>100</td>
<td>95</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>1110</td>
</tr>
<tr>
<td>8</td>
<td>Product 2</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>100</td>
<td>95</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>100</td>
<td>105</td>
<td>1110</td>
</tr>
<tr>
<td>9</td>
<td>Product 3</td>
<td>90</td>
<td>100</td>
<td>95</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>100</td>
<td>1100</td>
</tr>
<tr>
<td>10</td>
<td>Product 4</td>
<td>100</td>
<td>102</td>
<td>95</td>
<td>100</td>
<td>95</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>91</td>
<td>91</td>
<td>91</td>
<td>1100</td>
</tr>
<tr>
<td>11</td>
<td>Product 5</td>
<td>90</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>95</td>
<td>97</td>
<td>95</td>
<td>101</td>
<td>103</td>
<td>103</td>
<td>103</td>
<td>110</td>
<td>1130</td>
</tr>
</tbody>
</table>

**How it works...**

As you can see, there are many roles that cells in the spreadsheet can play. If these were not color coded, you would be faced with a daunting task when updating the Excel model in the future. In our example, we colored calculations in green, external data input in blue, and component input data in yellow.
There's more...

To make the data model readable, not only for yourself but also for others, it is helpful to create a legend in your spreadsheet that explains what each color represents. Any color scheme can be used, but it is important that you stick to the chosen scheme and use it consistently throughout the development of your dashboard.

It is important to create a separate worksheet that houses the legend, as seen in the following screenshot. You can also use this overall summary worksheet to include the information such as project name, description, usage, version (history), and so on.

Making the spreadsheet more readable with comments

Sometimes, cells need additional information to explain how they are used. You can create comment text on an adjacent cell. Or, if you do not want to fill up other cells, you can right-click on the same cell and select Insert Comment.

Getting ready

You need a basic dashboard containing a few components in the canvas with some bindings to the data model in the spreadsheet. You can also just reuse the dashboard from the previous recipe.

How to do it...

1. Right-click on the cell to which you want to add the extra information.
2. Choose Insert Comment.
3. Add your text. A small red triangle will appear in the upper-right corner of the cell.

4. Now hover your mouse over the cell and the comment you just entered will appear.

How it works...

Comments are related to one spreadsheet cell only and are shown if you hover over the cell. This is a great way to document information that you do not need to see all the time, and keeps your data model clean.
A little remark about the usage of comments: they increase the size of the SAP BusinessObjects Dashboards file a bit.

If you want the comment to always show up without hovering over the cell, you can right-click on the comment and then select Show/Hide Comments.

**Making the spreadsheet more readable with borders**

To separate cells from each other and create different areas within a spreadsheet, you can use cell borders.

**Getting ready**

You can use the same basic dashboard as in the previous examples.

**How to do it...**

1. Select the cell(s) you want to add a border to and right-click on it.
2. Now select **Format cells...**
3. Go to the **Border** tab.

4. Select the desired style of the border line.

5. Select on which side(s) of the cell(s) the border should appear.

6. Click on **OK**.

**There's more...**

We will now discuss three more topics regarding spreadsheet borders: using the toolbar border button, using multiple worksheets, and placing Excel logic within the spreadsheet.
Using the toolbar border button

Instead of right-clicking on the cells and using the Format Cells option, you can also use the Border button on the toolbar to adjust the border styles for a cell or a group of cells. You can find this button in the Font section of the Home tab. If you select the cell(s) and click on this button, a list of options will be shown, which you can choose from.

Using multiple worksheets

You can use borders to split data within a spreadsheet. But if your dashboard contains data from a lot of different (functional) areas, it is recommended that you split your spreadsheet into several worksheets. This will help you to keep your dashboard maintainable.

A good strategy to split up the spreadsheet is to divide your data in different areas that correspond to certain layers or tabs that you created on the dashboard canvas. You can also use separate sheets for each external data connection. Give each worksheet a meaningful name.
Chapter 1

Placing your Excel logic wisely
Another general guideline is to place as many cells with Excel logic and SAP BusinessObjects Dashboards interactivity functionality at the top left of the spreadsheet. This place is easy to reach without a lot of annoying scrolling and searching. Even more importantly, your dataset may grow (vertically and/or horizontally) over time. This can be a risk especially when you are using an external data connection and you don't want your logic to be overwritten. For example, if you use a column summation, place it at the top of the column instead of the bottom. This way, if you need to add another value to the list of cells to be summed, you can add it to the bottom without having to shift down the formula cell.

Using named ranges
With named ranges, it is possible to define a worksheet cell or a range of cells with a logical name.

Getting ready
You can use one of the dashboards from the previous recipes, or just create a new blank dashboard.

How to do it...
1. Select a range of cells (for example, B1:B12).
2. Insert a description (for example, Total_Sales) for this range in the Name Box in the upper left-hand side of the worksheet.
3. Now, this named range can be used in formulas in other worksheet cells. Type the formula =SUM(Total_Sales) in cell B13.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jan</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Feb</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Mar</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Apr</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>May</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Jun</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Jul</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Aug</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Sep</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Oct</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Nov</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Dec</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td>65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**How it works...**

Using named ranges makes your formulas more readable, especially when you are working with multiple worksheets and using formulas that refer to cells in other worksheets.

There is a restriction to using named ranges in SAP BusinessObjects Dashboards: the defined named range must refer to a single cell or must use formulas that return a single value from a range of cells. For example, let's say we have two named ranges: Sales (A1:A12) and Cost (B1:B12). A supported formula would be =SUM(Sales) - SUM(Cost). An unsupported formula would be =Sales - Cost.

**There's more...**

In this section, we will discuss two ways to select and manage your named ranges.
Defined named ranges
Clicking on the little triangle in the Name Box will show a list of all your defined named ranges in all your worksheets.

The Name Manager
If you use a lot of named ranges, the Name Manager can be a helpful tool to manage your named ranges. Here, you can also edit and delete the existing named ranges. You can find the Name Manager under the Defined Names section of the Formulas tab or by using the shortcut Ctrl + F3.
Copy the format of one cell to another cell or range

This recipe shows you how to copy the formatting of one cell to another cell or range. For example, we can copy a yellow background and Calibri font from cell A1 to cell A2.

How to do it...

1. Click on the source cell that you want to copy the formatting from.
2. Click on the Format Painter icon, which you can find in the Clipboard section of the Home tab.
3. Click on the cell or range that you want to copy the source cell’s formatting to.
How it works...

For SAP BusinessObjects Dashboards developers, it happens very often when they have to copy colored cells that represent different types of logic such as dynamic visibility cells, insertion cells, and so on.

The **Format Painter** tool works by taking the source cell that you have selected and applying the formatting to the cell(s) that you paint to. This is very useful because we can copy cell formats without having to perform a **Copy** and **Paste Special** action every time.

**There's more...**

An alternative to accomplishing the same task is to copy a cell and then click **Paste Special...** and choose the **Formats** option from **Paste**.

**Debugging the spreadsheets**

It is common that SAP BusinessObjects Dashboards developers may accidently put in the incorrect formula when developing logic on their spreadsheets. Using the **Ctrl + \``** hotkey will make things much easier.

**How to do it...**

1. Select the worksheet you want to see formulas for.
2. Hit the **Ctrl + \``** (grave accent) hotkey.
3. You will see the value in the cell change to the formula.

**How it works...**

The **Ctrl + \``** hotkey works by showing the underlying formula of a cell. This is extremely useful if you are comparing formulas from multiple cells, as the developer does not have to flip between formulas in order to see what they are doing wrong when comparing multiple cells. Developers can quickly analyze their worksheet and find the cause of their problem.
The following screenshot shows the results of two Excel formulas in cells A1 and A2:

After using the Ctrl + ` hotkey, the formulas of both cells are displayed, as you can see in the next screenshot:
There's more...

An alternative way to accomplish the same task is to go to the Formula Auditing section of the Formulas tab and then click on Show Formulas. Refer to the following screenshot:

Navigating between worksheets

When developing dashboards it is a common problem to have to manually scroll through tabs when there are too many worksheets. To access tabs that are not visible, we are used to pressing the arrow keys to move to the desired tab.
**How to do it...**

1. To have all tabs displayed in one menu, right-click on any of the arrow keys at the bottom, in the left-hand side navigation area. You will then see the list of tabs that you can choose from.

2. You can also use Ctrl + PageUp, which will move to the previous sheet in your workbook.

3. Ctrl + PageDown will navigate to the next sheet in the workbook.

**How it works...**

This tip allows developers to quickly toggle between worksheets. Being able to quickly right-click and view a menu of all available tabs is faster than scrolling through each tab in order to reach tabs that are not visible. In addition, the ability to use a hotkey to cycle through each tab brings some time-saving benefits to those who are comfortable with using the keyboard to perform all their actions.
**Where to buy this book**

You can buy SAP BusinessObjects Dashboards 4.1 Cookbook from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

[Click here](#) for ordering and shipping details.