Learning SAP BusinessObjects Dashboards

SAP BusinessObjects Dashboards is a leading Business Intelligence and reporting tool that provides you with a real-time understanding of your business with agile visualizations.

Starting with an introduction to Dashboards and its benefits, the book slowly moves on to explain the dashboard creation process. After this, you will learn how to add charts, single-value components, maps, selectors, and other third-party plugins to the existing dashboards. Furthermore, it shares many best practices and will also help you to connect your dashboard to real data by establishing a data connection to a data source. You can also explore more about mobile BI and learn how to create dashboards for mobile devices. By the end of the book, you will be able to prepare, plan, and design interactive dashboards based on your business requirements using this cutting-edge BI tool.

Who this book is written for

This book will help beginners to create stylish and professional looking dashboards in no time. It is also intended for BI developers who want to use SAP BO to facilitate BI in their organizations. No prior knowledge is required, however, you must have a basic knowledge of MS Excel and some analytical skills to build expressive business charts.
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 4 'Using Maps and Other Components'
- A synopsis of the book’s content
- More information on Learning SAP BusinessObjects Dashboards
Taha M. Mahmoud (PMP, TOGAF, ITIL, and CSM) is a senior BI consultant, BI project manager, and solution architect. He has a BS degree in computer science and automatic control from Alexandria University, Egypt. He has a great passion for new technologies, especially those related to business intelligence. Taha has more than 9 years of experience in working on, consulting for, and deploying successful BusinessObjects projects in the banking and telecom industries. He is the author of Creating Universes with SAP BusinessObjects. You can contact him on Twitter at @tahama_2000 or visit his blog (http://business-objects-xi.blogspot.com/).
Business Intelligence (BI) is one of the fastest growing fields in the market. The main goal of BI is to help us get good insights by utilizing the historical information that we have. Historical information helps us take the right decisions at the right time, based on the trends and patterns that occurred in the past.

BI is also concerned about delivering information to the right users in the right format. Operational users, for example, are more concerned about details and transactional information. This is why BI reports are the proper format for them. On the other hand, the top management and executives are more concerned about high-level information, enriched with indicators that can help them spot wrong or unusual behavior at first sight. This will then help them take corrective actions, which is why dashboards are the most proper format for them.
Why do we need dashboards?

"Dashboard" is not a new terminology. Actually, we use dashboards in many aspects in our life. A car's main dashboard is a good example, as we all know how to read a car's gauges and speedometers. You can see an example of a car's main dashboard in the following diagram:

![Car Dashboard Diagram](image)

The speedometer is used to indicate the current speed of the car. As the speed changes, the speedometer's needle moves towards the actual current speed. The gauge, dials, and the speedometer are components that we will discuss in detail in Chapter 3, UI Components. We should also note that we have an indicator that starts when our speed reaches 120 km/hr to indicate that we are at high speed (risk) and so we need to slow down.

Fuel, heat, and RPM meters also work in the same manner. They are used to indicate the current value and also to indicate (highlight) the danger values. Danger values signal that we need to take some action. The main use of a dashboard is to help us decide when we should act, but it will not give us detailed information on what is going wrong. For example, when a car engine's heat reaches the red zone, we know that we need to stop and examine the car to find out what the problem is, but we will not get information on whether there is leakage of water or there is a damaged part that needs to be replaced.
Now we should have a good idea about why we need dashboards, so before we start talking about SAP BusinessObjects (SAP BO) Dashboard (formerly known as Xcelsius), we need take a moment first to define what a dashboard is.

What is a dashboard?

A dashboard is a visual representation of information that can help us spot a risk, or bad or wrong behavior. It also can help us monitor and track our performance. You can see a dashboard example in this screenshot:

A dashboard is a container or view that can contain any number of the components listed as follows:

- Indicators
- Key performance indicators (KPIs)
- Key risk indicators (KRI)
- Scorecards
- Reports

We will discuss each component in detail in the upcoming sections.
Indicators
An indicator is a visual effect that can add extra information that is not included in the original metric.

Let's have an example to help you understand this in a better way. Let's say our profit this month is $10. As you can see, you can't judge whether this number is good or bad. But by adding some indicators, we may get a better idea about this metric's performance. As you can see in the following screenshot, the first row display information without indicator while the second row display information with yellow color and side arrow indicators.

There are many types of indicators, such as these:

- Traffic light colors
- Icons

Traffic light colors
The traffic light colors type is the most traditional indicator that we have. In this type, we utilize the colors common in traffic lights (red, yellow, and green) to give proper indications.

Red color is used to grab user attention and warn him. We use red color with loses or bad performing KPIs. Red color indicates that immediate correction action should be taken.

Yellow will give the impression that we should be prepared to do something, such as slowing down our car and preparing to stop at the traffic lights, or trying to increase our sales to increase your net profit.

Finally, green will give the impression that everything is okay and we are performing well. We can use green color with profit metrics and well performing KPIs.
Icons

Icons are another type of indicators. We can use an icon to give the required impression to end users. For example, a trend-up icon beside a profit metric will give the impression that we are trending up. We can find some other types of icon indicators, as follows:

- Arrows, such as up, side, or down arrows
- Faces, such as a smiley, normal, or sad
- Progress bars

Use icons if your dashboards will be printed in grayscale (black and white).

You also need to note the following:

- You can use more than one indicator type at the same time. For example, you can use traffic light colors and arrows to indicate your profit performance, as you saw in the previous screenshot.
- You can use more than three levels in the traffic light indicator type. For example, a five-color indicator may use the following colors: red, orange, yellow, light green, and dark green.

You can see an example of sets of indicator types in the following screenshot:

![Icon Examples](image-url)
**Key performance indicators (KPIs)**

When we start working on something, we should first define our goals and objectives. After that, we should start trying to achieve our goals. Then, from time to time, we need to check how far we are from our goals and whether our performance with respect to achieving our goals is good or bad. The main purpose of a key performance indicator (KPI) is to show how close we are to our goals (target). Normally, we will need more than one KPI to indicate how far we are from our goals.

Different industries will have different KPIs, even if they are related to the same goals, such as increasing profit, because the metrics are different. A KPI is a metric used to measure and monitor our performance in order to achieve our goal (or goals), and it gives us an indication of our performance.

Let's now look at a small example of a business case.

Let's suppose that we have a new website. There are many ways of income implemented in our site, such as advertisement, exam registration fees, products sold, and so on. First, we need to set our goals. Let's set a simple goal here: our net profit is $10 million, and our goal is simply to make it $15 million by the end of the fiscal year. So, the defined goal here is as follows:

**Goal: Increase the net revenue of our website by 50 percent during this year.**

Now we have a goal and we need to find out how to achieve it. There are many factors that will affect our goals, and we need to focus on the important ones. We call those factors metrics. A metric is usually a number that will affect our goal somehow, such as the number of sold products or Product price, so let's define our metrics here.

\[
\text{Net profit} = \text{Net income} - \text{Net cost}
\]

\[
\text{Net income} = \text{Product income} \cdot \text{Number of sold products} \times \text{Unit price} + \text{Advertisement income} \cdot \text{Number of visitors} \times \text{AD revenue per view} + \text{Exam income} \cdot \text{Number of scheduled exams} \times \text{Exam fee}
\]

\[
\text{Net cost} = \text{Fixed monthly site maintenance} + \text{Product processing cost} + \text{Exam setup} + \text{Other expenses}
\]
Now, as we can see, there are some variable metrics and some static metrics. For example, the number of visitors is a dynamic variable metric and monthly site maintenance fee is an example of a static metric. You should concentrate on dynamic metric in your KPIs.

We will use # of visitors as a metric in our KPI, but first we need to check whether this is enough of an indicator. Of course not! We need to link it somehow to our goal (achieving 50 percent growth in profit by the end of the year). To make it clearer, let's take a look at the # of visitors trend graph here:

As we can see, the graph displays the number of visitors (in thousands) per month. It is clear that this is not enough to know whether those figures are good or bad. Let's try to answer this question using the previous graph: Does the count of 100,000 visitors in January mean that we will be able to achieve our goal by the end of the year or not?

As we can see, a metric is just a plain number, and all that we can indicate here is the number of visitors trend by linking our metric's values across time. We can get an idea on whether our number of visitors is increasing or decreasing over time. Also, as we already saw in our goal definition, we need to increase the number of visitors in general to get more advertisement income. So far, this is just a trend metric and there is something missing.
In order to gain $15 million, let's say that our strategy is to focus on advertisement profit this year. If we maintain constant values of the remaining factors, then we should get $10 million by the end of the year and, to increase our profit, we have to increase our average number of visitors per month. Let's say the old average number of visitors to our site was 60,000 per month. If we get an average of 60,000 visitors per month and everything else remains the same, then we should make a profit of $10 million by the end of the year. We need to calculate the new required average number of visitors (target), assuming that we will not change any other factors. Let's say that we need, on average, 90,000 visitors per month to achieve our target, which will somehow lead us to our goal. Now the graph should like this, after adding the calculated target:

Now we have a KPI, as we can see after adding our monthly target that needs to be met in order to achieve our goal; we can indicate our performance month by month to achieve our goal. We can easily see that we performed well in Jan, Apr, and Aug. We nearly achieved our target in May, and performed badly in the remaining months.
How to define your KPIs

To define a KPI, we need to complete the following:

- **Define a goal**: First, we need to define our goals, or set of goals. Our goals should tell us what we want to achieve.
- **Define a metric**: The next step is to define our metrics. A metric is a number that will affect our goal.
- **Define a Target**: A target will help us understand how our metric should behave in order for us to achieve our goal.
- **Build your KPI**: A KPI will show us how our metric will behave against a preset target and will indicate our performance against our target.

Visual elements used to present KPIs (charts)

In the previous example, we had one measure (number of visitors) and one dimension (time), and this is why we selected the line chart—because it is the best visual element for showing a time trend. There are many other chart types, such as a pie chart for example, which can be used to show the relationship between one measure and one dimension. We can use a pie chart if we want to show, for example, sales by product. Also, we can use a combined chart (bar and line chart combined) to see the relationship between two measures and one dimension, such as the relationship between the communication channel, number of complaints, and average service time. We will discuss how to select the most proper chart components based on our metrics in Chapter 3, UI Components.

Key Risk Indicator (KRI)

A key risk indicator (KRI) is mostly the same as a KPI but with a few differences, as listed in the following table:

<table>
<thead>
<tr>
<th>Feature</th>
<th>KPI</th>
<th>KRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measuring</td>
<td>Performance</td>
<td>Risk</td>
</tr>
<tr>
<td>Against</td>
<td>Target</td>
<td>Threshold</td>
</tr>
</tbody>
</table>

In many cases, we may need to incorporate an alerting system with a KRI to send it immediately by mail or a warning SMS message to the risk owner. This is because, in most cases, we want to act immediately when the risk is triggered.
Scorecards

A scorecard is a group of related KPIs that contribute to achieve a major goal. There are two types of scorecards:

- **Balanced**: This is an equal-weight score card, which means that all KPIs under this goal have the same importance; when we calculate our achievement percentage for our goal, we simply take the average.

- **Not balanced**: This is a none equal-weight scorecard, which means that every KPI has its own weight (importance); when we calculate our achievement percentage for our goal, we consider the KPI weight to calculate the average.

You can see a scorecard example in this screenshot:
Reports
A report is a summary or detailed information displayed in a simple table or chart format. An example of a detailed report is shown in the following screenshot:

![Branches Summary Table]

In this book, we will discuss all that you need to learn SAP BusinessObjects Dashboard Designer. This is a SAP tool that can be used to create stunning dashboards, KPIs, KRIIs, and scorecards using the Flash and MS Excel technologies.

Congratulations for taking a step towards learning how to create dashboards using SAP BO Dashboard. Are you ready? Then let's go...

In this book, you will learn how to create a complete, interactive dashboard that contains charts, single-value components, selectors, and maps. You will learn how to apply advanced features, such as dynamic visibility, alerts, and color binding.

What this book covers
Chapter 1, Getting Started with SAP BO Dashboards, shows you how to download, install, and run SAP Dashboard Designer. After that, we explore this tool's capabilities and features by accessing SAP BO Dashboard designer templates and samples. We do this to demonstrate the capabilities of this tool and make you more excited to learn about it further. Then we discuss the SAP BO Dashboard Designer interface, menus, and panels.
Chapter 2, *Understanding the Dashboard Creation Process*, makes you familiar with the process of creating dashboards. We start by talking about the business requirement gathering phase. Then we discuss how important it is to sketch the initial requirements on a plain paper and think beyond data. After that, you get to learn how to create a prototype for our dashboard project, which we build step by step as we progress through this book.

Importing data is the first step in the process of dashboard creation, so here you learn how to import data into your dashboard project. Then you learn how to maintain your Excel sheet and make it more readable. You also learn how to use imported data, which is meant for later chapters.

Chapter 3, *UI Components*, is where we start building the model by adding chart components to our dashboard project. Then you learn how to link this with the data that we imported in the previous chapter. After that, we see how to play with our charts' properties and how to handle missing data.

The single-value component is another visual element that we can use inside our dashboard, but because it is totally different from charts, we discuss it in detail.

Chapter 4, *Using Maps and Other Components*, teaches you how to add a map to your dashboard project. In this chapter, you also learn how to install, configure, and use third-party add-ons for Google Maps. Then you get to know the other available components in SAP BO Dashboard Designer.

Chapter 5, *Interactive Analysis with Dashboards*, explains selectors and shows you how to use them.

Chapter 6, *Advanced Interactive Analysis with Dashboards*, is the core of this book. In this chapter, we explain how to make our dashboard interactive using dynamic visibility. Also, you get to learn how to add alerts to your charts, single-value components, and selectors. After that, we see how to use data insertion and containers.

Chapter 7, *Styling Up*, demonstrates how to customize the look and feel of our dashboard by applying themes, changing colors, adding media and a logo, and so on.

Chapter 8, *Exporting, Publishing, and Importing Dashboards*, teaches you how to export your dashboard in different formats. Then you learn how to publish it to make it available for others.

Chapter 9, *Retrieving External Data Sources*, explains how to connect to data retrieved from other data sources.

Chapter 11, Creating Mobile Dashboards, shows you how to create a dashboard for mobile applications.

Appendix, References, will include references for Supported excel functions, List of Built-in maps, and Supported mobile components and connections.
Using Maps and Other Components

We have many dashboard components that we can use inside our dashboard. In the previous chapter, we discussed how to add charts and single-value components to our dashboard. In this chapter, we will learn how to add maps and other dashboard components.

In this chapter, we will discuss the following topics:

- Using SAP BO Dashboards built-in maps
- Configuring and using the CMap plugin
- Using other dashboard components, such as trend, icon, and calendar
- Configuring and using the dash print plugin
- Configuring and using the micro chart plugin

Maps

Maps are efficient when we want to represent data related to geographical dimensions, especially when we use indications and alerts to highlight the presented information. We can get more value from the data presented on a map than data presented on a chart or in a table. For example, you can easily find that your company's major sales are done in the MENA region using a map to represent sales information, while you may take a while to find out the same fact using the same sales information presented using a chart or table. This is because maps contain more geographical information than just the location. Another example is as follows: you can easily find a relation between the coastal states of the US. and, let's say, low average temperatures across the year. So, maps are prefect for locations and geographical dimensions, and as a best practice, it is recommended to use maps in such cases.
There are many prebuilt, or out-of-the-box, maps in SAP BO Dashboards that can be used directly by linking data, adding alerts, and using the drill feature. We can also use a third-party add-in to integrate and use Google maps inside our SAP BO Dashboards. You can refer to Appendix, Built-in maps available.

You will learn how to use a map to drill down to another dimensional level in the hierarchy. You will also learn how to configure the Alert and Insert properties for your maps in Chapter 5, Interactive Analysis with Dashboards.

Besides this, we will discuss other dashboard components, such as the calendar, trend icon, history components and many more.

Using maps
Maps are a very important subject, and so we need to pay special attention here. In this section, we will discuss these topics:

- Using built-in maps
- Using the CMap plugin

Adding built-in map dashboard components
In this section, you will learn how to add a US-based map to the eFashion sales data by state. Then, we will link this map with data already prepared from the Excel model.

Before we start, however, we need to open the latest .xlf dashboard file that we completed in the previous chapter. If you didn't complete the steps provided in the previous chapters to create the eFashion dashboard, or if you just want to practice with the map part, then you may open the Chapter3.xlf dashboard from the Dashboard (Ready) folder, which is available under the example code folder. We can see that folder in the following screen:
Chapter 4

Now, let's add the US maps together:

1. Navigate to the Maps section under the Components panel.
2. Drag the USA (continental) map and drop it onto the canvas.

We can see these steps in the following screenshot:

Now, let's link our USA map with our data about eFashion sales by state:

3. Select the Map sheet from the Excel model.
4. Make sure that the USA map is selected, and then navigate to the General tab under the Properties panel.
5. Edit Region Keys by clicking on this edit icon: 
6. Change the Washington region key to DC.
7. Link the Display Data field with the Map!$B$4:$C$11 data range.
We can see these steps here:
8. Click on the **Preview** button to see the map after linking it with the data. You can see it in the following screenshot:

Before moving on to the next section, let's discuss some important terms related to maps:

- **Region keys**: Each map has a predefined set of regions. Each region has a name and a key. The map region's name can't be changed because it points to a location in the map, while the region keys are dynamic and can be adjusted as per our data. In our example, we changed the key for the Washington state from Washington to DC to match our data. We didn't need to make any other change because our data keys were matching with the right map region keys.

  The map region keys are case-sensitive and should exactly match your data region keys in order to be displayed correctly.

We can map the region keys to a range in your Excel model, but we need to make sure that our region keys' order and count matches the right order of region names and count, otherwise we will get a wrong mapping. We also need to note that the map of USA contains all the states, whereas we are displaying only eight states in our eFashion sales information.
Installing, configuring, and using the CMap plugin

CMap is a plugin provided by Centigon (http://cmapsanalytics.com/) to integrate Google maps with SAP BO Dashboards. The first thing we need to discuss is how to download and install this plugin. Then, you will learn how to create the same map of USA based on our eFashion sales information, but using the CMap plugin this time:

2. Navigate to Products | CMaps Analytics | SAP Dashbords.
3. Click on the Download button to download the trial version of the CMap plugin.
4. Fill in the form. Within a few seconds, you will receive an e-mail with the download link and your trial serial number.

The page at http://cmapsanalytics.com/ is shown in this screenshot:

Now, we need to use Add-On Manager to install the CMap plugin:

1. Navigate to File | Manage Add-Ons.
2. Click on the Install button.
3. Navigate to the downloaded file, CMapsPluginMobile_4.2.0.xlsx.
4. The program will ask you to save and exit, so do it.
You can see these steps in the following screenshot:

![Add-On Manager screenshot](image)

We can see the Add-On Manager in the following screenshot:

![Add-On Manager screenshot](image)
Now, let's use the CMap plugin to create the same map that we created using built-in dashboard components:

1. Open the Components panel.
2. Then, navigate to the Maps category.
3. Drag the CMaps Plugin components and drop them onto the canvases.

Then, we need to follow these steps to configure our CMap Plugin component:

4. Select the CMaps Plugin components and open the Properties panel.
5. Navigate to the General tab, and then link the Key field to the Excel cell that contains a valid CMap key.

   ![CMap Key Reminder]
   
   You will receive this key by e-mail if you registered and have downloaded a trial version of this plugin. Normally, the trial period is 14 days.

6. Type sales in the Series Name field.
7. Select Address Data to enter the location by address or coordinates (latitude and longitude).
8. Select the Map!$B$4:$B$11 range in the Address/Lat, Lng Range field.
9. Select the following Map!$B$4:$B$11 range in the Labels field and Map!$C$4:$C$11 in the Values field.

You can see these steps here:

Before we conclude this part, let's discuss some important concepts:

- **Address data/shape data**: There are two modes that we can use to represent our data using the CMaps plugin. The first mode is **Address Data**, which can deal with address-related information in various formats such as zip code, city, state, or coordinates. Coordinates can be defined as a latitude and longitude separated by a comma (,). We can use the **Shape Data** mode to display polygons and lines that can be used to specify areas on the map. We may have a shape file that divides the map of USA into, for example, five main regions (central, eastern, western, northern, and southern).
For more information about these two options, refer to the Tutorial tab under http://centigonknowledge.com/, or click on the question mark icon beside address data. You can also check for any restriction on address recognition for the location that you want on their site.

- **Labels and values**: These fields will be used to display the associated label and value with each address or coordinate. Each location will be presented as a point on the map. You can see the map in this screenshot:

![Map with states and coordinates](image)

Now, let's add a shape data series to our map:

1. Select the CMap component and navigate to the General tab under Properties panel.
2. Create a new Shape series by clicking on the Add button under Series.
3. Select the Shape Data mode and click on Shape Data Options.
4. From the Shape Options window, select Shape File URL.
5. Select the following Excel range (list of states): `Map!$B$4:$B$11` in Shape File Order Keys under the Data to Shape Linking & Visibility section.
6. Select this Excel range: `Map!$C$4:$C$11` in Data Order Keys under the Data to Shape Linking & Visibility section. Click on OK.
7. Map the **Single Shape File URL** field to **Map!$B$1**, which contains the URL for the USA shape file.

You can find many shape files at [http://cmapsanalytics.com/](http://cmapsanalytics.com/). You can also download and use shape files directly instead of using a URL.

8. Map **Labels** and **Values** field to the **Map!$B$4:$B$11** and **Map!$C$4:$C$11** ranges.

9. Click on the **Refresh** button.

You can find the steps illustrated in the following screenshot:

To enhance the map's visibility, we may need to adjust the **Transparency** setting for the shape series:

1. Navigate to the **Appearance** tab under the **CMap Component** properties.
2. Go to the **Icon** section and select the **Shape** series.
3. Set the **Transparency** setting to 75.
You can see these steps marked in this screenshot:

The final map should look like this:
You will learn how to configure map alerts and how to use the insertion feature for maps in the following chapters. All that we need to focus on here is how to add and configure map components—either built-in maps or CMap plugin components.

Using other SAP BO Dashboard components

There are many other components that can't be categorized as charts, selectors, maps, or containers. These dashboard components are grouped under the Others category, and, as we will see, there is no relation among them. You will learn how to use other dashboard components in this section, such as:

- Calendar
- The trend icon
- The trend analyzer
- The print, reset, and local scenario buttons
- The history and source data components.

As mentioned before, dashboard components are divided into categories under the Components panel to make it easy to navigate and select the required dashboard component. We have the following main categories:

- Charts
- Selectors
- Single Value
- Maps
- Other
You can see the main component categories in the following screenshot:

All the functions that cannot be performed under the existing/mentioned categories are stated under the Other dashboard components. Before we start discussing the various dashboard components under the Other category, we need to organize our dashboard.

As a best practice, we should give a unique name to each dashboard component in the Object Browser panel. We also need to hide and lock the completed and finalized components. As of now, we have a pie chart behind the combination chart and a built-in map of USA behind the CMap dashboard component.
To do this, let's follow these steps:

1. Navigate to **Object Browser**.
2. Rename the dashboard components as displayed in the following table:

<table>
<thead>
<tr>
<th>Old component name</th>
<th>New component name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pie Chart 1</td>
<td>Sales by Product</td>
</tr>
<tr>
<td>Combination Chart 1</td>
<td>Sales and Quantity by Product</td>
</tr>
<tr>
<td>Bubble Chart 1</td>
<td>Managers Performance</td>
</tr>
<tr>
<td>Gauge 1</td>
<td>Sales Current Month</td>
</tr>
<tr>
<td>Horizontal Progress Bar 1</td>
<td>YTD Sales</td>
</tr>
<tr>
<td>Value 1</td>
<td>Value 1</td>
</tr>
<tr>
<td>USA (continental) 1</td>
<td>Sales by State</td>
</tr>
<tr>
<td>CMaps Plugin 2</td>
<td>Sales by State CMaps</td>
</tr>
</tbody>
</table>

You will see **Object Browser** like this after the modifications:

![Object Browser](image)

Now, let's start exploring the other dashboard components.
Using Maps and Other Components

Using a calendar dashboard component

Calendar components are perfect for date input as they provide a calendar interface for selecting a specific date. They also provide easy navigation between months and years. The only disadvantage of this component is its large size. Therefore, we should use the dynamic visibility option to hide and show it as per requirements.

A Calendar component can be used to specify only one date. If we have a date range (a start date and an end date), then we may use two calendar components to define the period.

As we don’t have any calendar component in our eFashion dashboard prototype, we will create a side example for the Calendar component:

1. Activate the Side2 tab and drag a Calendar component into the canvas.
2. Navigate to the General tab under the Properties panel.
3. Enter Snapshot Date in the Title field.
4. Change Insertion Type to Date.

   ![Information Icon]
   You can click on the information icon, to get more information about the Insertion Type functionality.

5. Link Destination to the following Excel cell 'Side Examples'!$B$2, Month Destination to 'Side Examples'!$C$2, and Year Destination to 'Side Examples'!$D$2.

   You can see these steps here:
You can see the information box for **Insertion Type** in the following screenshot:

Now, let's adjust the behavior properties of our calendar component:

1. Navigate to the **Behavior** tab.
2. Select Use Custom Date as **Default Date**.
3. Link Day to 'Side Examples'!$B$3, Month to 'Side Examples'!$C$3, and Year to 'Side Examples'!$D$3.
4. Tick the box beside **Enable Calendar Limits.**
You can see these steps in this screenshot:

5. Now, add three text labels and map them to the 'Side Examples'!$B$3, 'Side Examples'!$C$3, and 'Side Examples'!$D$3 calendar destination Excel cells. Then, click on Preview to see a calendar like this:
Before we end this section, let's discuss some calendar functionalities that we’ve used in our example:

- **Date/day insertion type**: We can choose to insert the entire calendar date or only the day portion of the date. For example, if we select 7-Jan-2015, the date insertion will be 7-Jan-2015, while the day insertion will be only 7.

- **Calendar default date**: This can be the current date, and it can be customized to a specific date. Normally, in a data warehouse implementation, the most recent snapshot date is the previous day’s data because batch jobs and ETL jobs run overnight. We use the Today()-1 Excel function to get the previous day’s date and to set it as the default calendar date.

- **Enable calendar limits**: This can be used to limit the date navigation to a specific period. In our example, we limited the navigation to the current year.

  We can use the **Show Formula** option in Excel to display the formula entered in Excel cells.

You can see how **Show Formula** works in the following screenshot:

![Show Formula](image)

It is worth mentioning that the calendar is one of the selector components that we will discuss in detail in the next chapter.

**Using the trend icon**

The trend icon is one of my favorite dashboard components. It will display a blue up arrow if the linked value is positive, a yellow neutral sign if the value is zero, and a red down arrow if the value is negative. Let's try to create one:

1. Drag three **Trend Icon** dashboard components into the **Side2** tab set canvas.
2. Link the data field for the first trend icon to 'Side Examples'!$B$6, the second to 'Side Examples'!$B$7, and the last to 'Side Examples'!$B$8.
Using Maps and Other Components

We can customize only the Trend Icon's colors.

You can see one of the Trend Icon properties in this screenshot:

![Trend Icon properties screenshot]

Trend icons should look like this when previewed:

![Previewed trend icons]

**Using a trend analyzer**

A trend analyzer can help us show a trend of a data series, as well as forecasted trends based on historical data. The forecasted data is generated by many standard forecasting algorithms. These can be defined in the Type field in the trend analyzer component.

To test it, let's build the following side example, as we don't need a trend analyzer to build our eFashion dashboard:

1. Create a combination chart based and map the first data series (primary axes) to the sales information under the Side Examples Excel sheet.
2. Drag and drop the Trend Analyzer dashboard component onto the canvas.
3. Navigate to the General tab under the Trend Analyzer properties panel.
4. Link the Data field to 'Side Examples'!$B$12:$E$12.
5. Select linear in the Trend/Regression Type.
6. Enter the 'Side Examples'!$B$13:$F$13 range in the AnalyzedDataDestination field.
7. Make sure that NumberofForecastedPeriod is set to 1.
8. Select the combination chart again and create new series called trend.
10. Make sure that the trend series is set as secondary axes.

Some of these steps are shown in this screenshot:
You can see the combination chart series in the following screenshot:

The line column chart output is shown here:

**Using the print, reset, and local scenario buttons**
The *Print* button can be used to print your dashboard. It is really a basic *Print* button, and there is not much to talk about here, but if you want a better alternative, you can go with the third-party add-on print button developed by Data Savvy.

The **Reset** button will simply reset everything to the default initial state as if you have closed the dashboard and opened it again.

The local scenario button can be used for the **Save**, **Load**, **Delete**, and **Set Default** scenarios. For example, we can use a slider to implement a what-if scenario, and to save and load it, we can use this **Scenario** button. You can see the available **Scenario** button options in the following screenshot.

Using the history and source data dashboard components

The history component can be used to track data history. The component will store the previous values of a single value, and so, you can track the trend of that value. To set up a history component, you need to define a data cell, which will contain the dynamic changing metric, and a data destination range, which will contain the historical records of the data cell. We can set up the history component to capture a data cell value in time intervals or when data is changed.

The source data component can be used to select a value, row, or column from the source range to insert it into a target cell, based on an index cell. So, for example, if we have a source data range of four cells, then we can select one cell from that range to insert in a target cell based on the current index value.

There is no specific use case for the source data component.
Using Maps and Other Components

Let's use a side example to explain these two components, as we don't need to use **History** or **Source Data** components in our eFashion dashboard project:

1. Drag one of each of the following dashboard components into the side2 tab under the current tab set:
   - History
   - Source Data
   - Line chart
   - Spinner

The canvas should look like what is shown in this screenshot:

2. Select the **Source Data** component and navigate to the **General** tab under the **Properties** panel.
3. Make sure that **Insertion Type** is set to **Value**.
4. Link the **Source Data** field to the 'Side Examples'!$B$17:$E$17 Excel range, and the **Destination** field to the 'Side Examples'!$B$18 Excel cell.
5. Navigate to the **Behavior** tab and link the **Selected Item Index** field to 'Side Examples'!$B$19.
You can see the **Source Data** component's configuration steps in the following screenshot:

Now, let's configure the **History** component:

6. Select the **History** component and navigate to the **General** tab under the Properties panel.
7. Link the **Data** field to the 'Side Examples'!$B$20 Excel cell.
8. Then link the **Data Destination** field to the 'Side Examples'!$B$21:$E$21 Excel range.
9. Make sure that the **When Value Changed** option is selected.

The steps for configuring the **History** component are displayed in this screenshot:
10. Finally, we need to configure our chart to display the 'Side Examples'!$B$21:$E$21 history destination range, and the spinner to change the Source Data index value 'Side Examples'!$B$19. The final result should look like this:

Before concluding this section, let's discuss the main features of the Source Data component:

- **Insertion Type:** We can use one of the three insertion types in the Source Data component. These are value, row, and column. We will discuss insertion types in detail in Chapter 6, *Advanced Interactive Analysis with Dashboards*. You can use the information icon, as described earlier, to learn how to use the Source Data component, as shown in this screenshot:
• **Selected Index Item**: This cell can be used to decide which cell, row, or column will be inserted into the source data destination.

We also need to discuss how the *History* component is used in this example. The line chart data is linked to the destination of the *History* component. This will allow us to see how the *History* data will change when we change the spinner value. The spinner value will control the index field for the source data. So, the first data to be entered in the history data field is 10 because the default spinner value is 1. When we change the spinner value to 2, it will change the source data index to 2, so it will insert the second value into the source data, which is 5. This will change the value of the history data, and so the history component will detect the change. This is because we set the insertion of historical data to *When Data Changed*. This will insert a new value into the historical destination range, will plot another point on the line chart, and so on.

Now that we've completed this section, we will talk about some important third-party plugins.

**Using other third-party plugins**

In this section, we will discuss the following third-party plugins:

- Dash printer
- Micro chart

**Using a dash printer**

Dash printer is a plugin developed by Data Savvy. It enables the end user to print their dashboard. This plugin will help you select an area to print, annotate, and then share flexibly, as you can see in the following screenshot:
You can register and download it from the **Products** section at [http://datasavvytools.com/](http://datasavvytools.com/). After you've downloaded the plugin, you can install it in exactly the same way we installed the CMap plugin.

Let's create a side example to learn how to use this plugin. Make sure that you've installed the plugin before you continue reading:

1. Navigate to the **Data Savvy Tools** category under the **Component** panel.
2. Then, we drag and drop **Dashboard Printer** onto our **Side2** tab canvas under the **Current** tab set.
3. Click on the **Preview** button and then on the **Print** icon.
4. Select the area that you want to print, and then click on **Preview** (this is the printer preview).
5. Check how the selected area will look after printing. You're also able to annotate on it.

Besides printing, you can also share it.

**Using Micro charts**

Micro charts is a plugin developed by Inovista (http://www.inovista.com/) to help dashboard developers create Micro charts. This is one of many plugins provided by Inovista, but we'll take this one as an example here.

There are many other dashboard plugins provided by Inovista, such as SVG components, organization chart, and text and shape suites.

Micro charts can be used to display charts in a small area.
Using Maps and Other Components

We can download a trial version of the Micro chart plugin from the Inovista site and install it using the add-on manager just as we did before:

1. Navigate to the **Inovista Micro Charts** category under the **Component** panel.
2. Drag a **MicroBarChart** component into the canvas.

3. Select the **MicroBarChart** component and navigate to the **General** tab under the **Properties** panel.
4. Link the **Value** field to the 'Side Examples'!$B$24:$E$24 Excel range.
5. Make sure that the color selected for **Negative Line** is red and the checkbox is selected.
You can see some of the preceding steps marked here:

![MicroBarChart](image)

For the chart to be visible in a preview, the values must be set.

- **Values**: Side Examples\$\{SB\$24:SB\$24\}
- **Color**: [Select Color]
- **Bevel Chart Items**: [Check]
- **Bar Spacing**: [Adjust Spacing]
- **Negative Line**: [Check]
- **Remove Begin/End Blanks**: [Check]

You can see the Micro chart in the following screenshot:
Now, it's time to save your dashboard in your development folder, enabling you to use it for exercises that we will cover in the upcoming chapters.

There are a number of add-ons that might be of interest to you. You can explore them at the following sites:

- RangeFinder from Altek solutions (http://www.alteksolutions.com/)
- Organization Chart from Inovista (http://www.inovista.com/)
- There are many add-ons (more than 40) available from visualbi (http://visualbi.com/)

**Summary**

In this chapter, you learned in detail about maps, their usage, and the CMap plugin. Then, we explored the dashboard component under the Other category. We concluded the chapter by covering third-party add-ons, such as dash printer (Data Savvy) and micro chart (Inovista).

In the next chapter, we will dive deep into **how to use selectors**, which was highlighted during the discussion of single-value and calendar components earlier. However, we will focus on traditional selectors in the next chapter, such as the radio button, checkbox, drop-down menu, and so on.
Where to buy this book

You can buy Learning SAP BusinessObjects Dashboards 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.