



salesforce®

Reporting Workbook

Reports and Dashboards Everywhere



**30 min
Tutorials**

Easy-to-Use Dashboards
Powerful Reports
Sharing and Collaboration
Custom Formulas
Embedded Charts
REST API Access

CONTENTS

About the Reporting Workbook	1
Tutorial 1: Getting Started with a Simple Dashboard and Report	3
Step 1: Create a New Dashboard	3
Step 2: Create the Closed Sales This Quarter Report	3
Step 3: Create the Gauge Dashboard Component	5
Step 4: Filter the Dashboard	6
Summary	6
Tutorial 2: Creating Combination Charts and Custom Table Components	8
Step 1: Create the Sales Manager Leader Board Report	8
Step 2: Add a Combination Chart on the Report	9
Step 3: Create the Top Five Salespeople Custom Table Component	10
Step 4: Add the Combination Chart to the Dashboard	11
Step 5: Create a Neglected Accounts Report With Cross Filters	12
Summary	13
Tutorial 3: Creating a Matrix Report with Custom Summary Formulas	14
Step 1: Create the Revenue Trend by Type Matrix Report	14
Step 2: Create a Formula Using the PARENTGROUPVAL Summary Function	15
Step 3: Create a Formula Using the PREVGROUPVAL Summary Function	17
Step 4: Add a Cumulative Line Chart Component to the Dashboard	19
Step 5: Add a Bucket Field and Show It as a Pie Chart	20
Summary	22
Tutorial 4: Creating a Joined Report with Custom Summary Formulas	23
Step 1: Create the Pipeline Predictor Joined Report	23
Step 2: Add a Custom Cross-Block Formula	24
Step 3: Build a Sales Rep Scorecard	25
Step 4: Add the Sales Rep Scorecard to the Dashboard	26
Summary	28
Tutorial 5: Creating a Pipeline Push Report	29
Step 1: Create a Simple Historical Report	29
Step 2: Find Deals that Have Been Pushed Out	30
Summary	30
Tutorial 6: Sharing and Collaboration	31
Step 1: Create a Shared Dashboard Folder	31
Step 2: Embed a Chart in a Record	32

Contents

Step 3: Turn on Feed Tracking for Reports and Dashboards 33

Step 4: Follow Your Favorite Dashboards and Get Alerts on Components 34

Summary 35

[Tutorial 7: Building Custom Apps with the Salesforce Reports and Dashboards API](#)
..... 36

Step 1: Run a Report Synchronously 37

Step 2: Run a Report Asynchronously 37

Step 3: Filter Report Data 38

Step 4: Find, Show, and Refresh Dashboards 38

Summary 41

[Final Summary](#) 42

ABOUT THE REPORTING WORKBOOK

USER PERMISSIONS

To create, edit, and delete reports:	"Create and Customize Reports" OR Editor access to the report folder.
To create, edit, and delete dashboards:	"Create and Customize Dashboards" OR Editor access to the dashboard folder.
To create, edit, and delete public report folders:	"Create Report Folders" OR Manager access to the report folder
To create, edit, and delete public dashboard folders:	"Create Dashboard Folders" OR Manager access to the dashboard folder

This series of tutorials introduces you to dashboards and reports in Salesforce. *Salesforce Reports and Dashboards* is the term we use to refer to everything dashboards and reports enable you to do with your data.

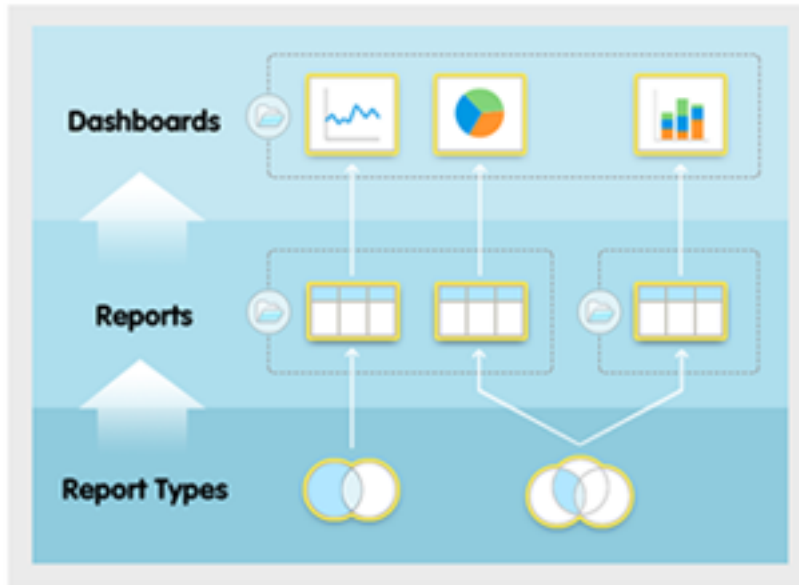
A lot of different features come together to make up Salesforce Reports and Dashboards. Don't get caught in the trap of thinking of the pieces separately. The reporting and dashboard features work together to present the information that's important to you—from the big picture to the data point.

The tutorials in this workbook build on one another and are designed to be done in order. We'll start by walking you through a few interesting use cases that suggest what reports and dashboards can do for you. Then we'll discuss how you can take advantage of built-in tools to share the information with others. Along the way, we'll highlight some cool features, like custom table components, combination charts, conditional highlighting, cross filters, buckets, and custom summary formulas.

If you're still game, we'll go into more advanced topics, such as merging multiple reports into joined reports and embedding report charts in record detail pages. And we'll take a look at how you can interact with your reports from other apps through the Reports and Dashboards REST API.

Conceptual Overview

A *dashboard* shows data from source reports as visual components, which can be charts, gauges, tables, metrics, or Visualforce pages. The components provide a snapshot of key metrics and performance indicators for your organization. A *report* returns a set of records that meets certain criteria, and displays it in organized rows and columns. Report data can be filtered, grouped, and displayed graphically as a chart. Reports are stored in folders, which control who has access. A *report type* defines the set of records and fields available to a report based on the relationships between a primary object and its related objects. Reports display only records that meet the criteria defined in the report type. Administrators control access to reports and dashboards by storing them in folders. Folders can be public, hidden, or shared.



To learn more, see [Getting Started with Salesforce1 Reporting](#).

Workbook Version

This workbook is updated for the Winter '15 release of Salesforce.com, and was last revised on September 16, 2014. To download the latest version of this workbook, go to <https://developer.salesforce.com/docs>

Permissions You Need

This workbook is designed to be used with any organization that has the reporting features enabled. You'll need the following permissions to perform the exercises in this workbook:

This book assumes you have the following features enabled:

- Report builder
- Dashboard builder
- Chatter
- Historical trending
- Reports folder sharing

To use cross filters, buckets, and joined reports, you must be logged into a Developer Edition, Enterprise Edition, Unlimited Edition, or Performance Edition organization.

If you don't have the right permissions, but still want to try out the exercises, use a Developer Edition organization, which has all of the required features and functionality. Sign up at <http://sforce.co/1tp5K2Z>.

TUTORIAL 1: GETTING STARTED WITH A SIMPLE DASHBOARD AND REPORT

Level: Beginner

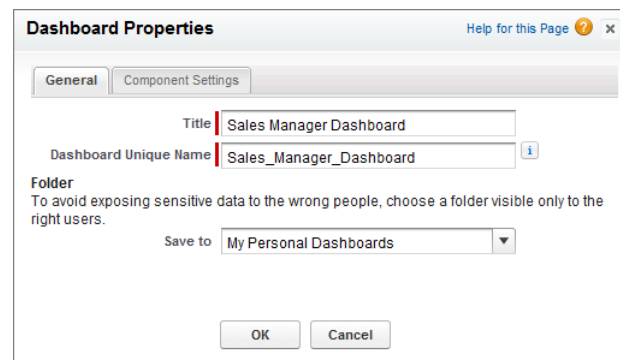
Dashboards give you the big picture, letting you see your key metrics at a glance. In this tutorial, we'll create a dashboard, build a report, then go back to the dashboard to create a component based on that report. The end result will show closed sales for the current quarter against projected quota. And we'll throw in a way for users to zero in on the data that's most interesting to them.

Step 1: Create a New Dashboard

Let's start by creating a dashboard. It won't look like much without any components, but we'll fill it in as we go through these tutorials.

Each dashboard has a *running user*, whose security settings determine which data to display in a dashboard. For this dashboard, the default running user will be you, the creator.

1. Click the Dashboards tab. You'll see your most recently viewed dashboard.
2. Click **Go to Dashboard List**.
3. Click **New Dashboard**. This takes you to the dashboard builder, a drag-and-drop editor for creating a dashboard and its components. Since we haven't created our source reports, we can't add any dashboard components just yet.
4. Click **Dashboard Properties**.
5. Enter *Sales Manager Dashboard* for the title and accept the auto-generated unique name.
6. Choose the *My Personal Dashboards* folder for now. In a later tutorial, you'll create a shared folder.

The screenshot shows a 'Dashboard Properties' dialog box with two tabs: 'General' and 'Component Settings'. The 'General' tab is active. It contains three input fields: 'Title' with the text 'Sales Manager Dashboard', 'Dashboard Unique Name' with the text 'Sales_Manager_Dashboard', and a 'Folder' dropdown menu set to 'My Personal Dashboards'. Below the 'Folder' dropdown is a note: 'To avoid exposing sensitive data to the wrong people, choose a folder visible only to the right users.' At the bottom of the dialog are 'OK' and 'Cancel' buttons. A 'Help for this Page' link with a question mark icon is in the top right corner.

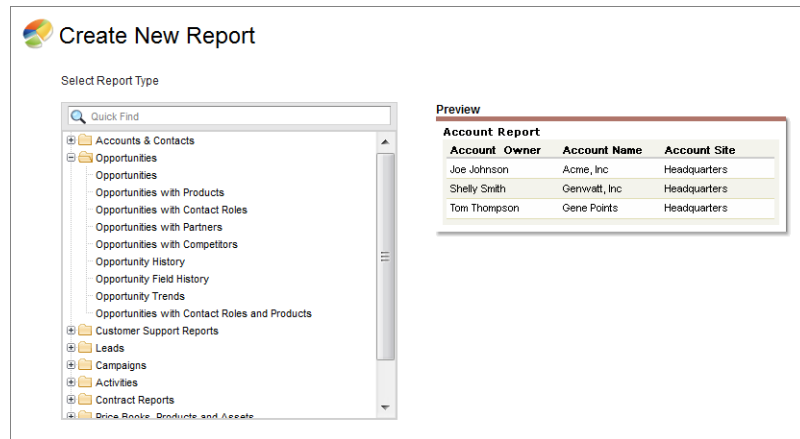
7. Click **OK**.
8. Click **Save**, then close the dashboard.

Step 2: Create the Closed Sales This Quarter Report

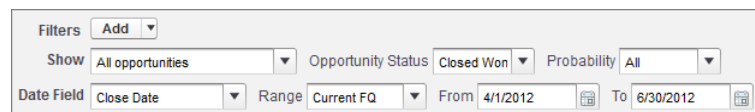
Before we can create dashboard components, we have to create the underlying source reports. In this step, we'll create a simple report showing closed sales for the current quarter. In the next step, we'll create a gauge on our Sales Manager Dashboard that shows these sales against quota.

1. Click the Reports tab.

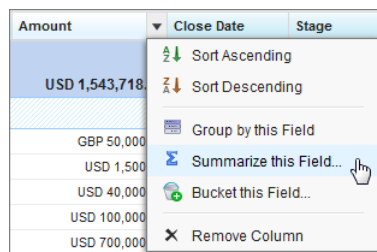
2. Click **New Report**.
3. Choose the *Opportunities* report type. This determines the data available to report on.



4. Click **Create**. This takes you to the report builder, a drag-and-drop visual editor for reports. Notice that a number of fields are already in the report by default.
5. Apply the following filters:
 - Select *All Opportunities* for *Show*.
 - Select *Closed Won* for *Opportunity Status*.
 - Select *Close Date* for *Date Field*.
 - Select *Current FQ* for *Range*.



6. Click **Tabular Format** and change the report format to **Summary**. This allows us to create groupings.
7. Group the report by *Close Month* by dragging that field into the grouping drop zone. You can create secondary groupings and add a chart, but let's keep it simple for now.
8. Click the menu for the *Amount* column and select **Summarize this Field**.




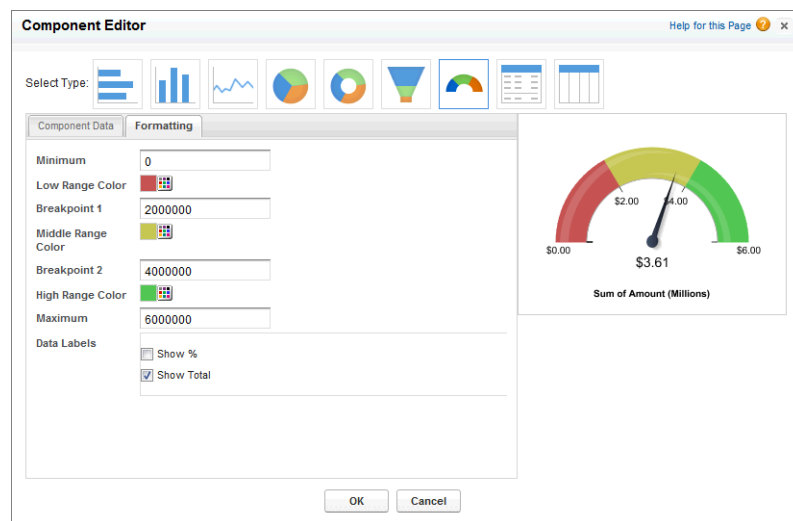
9. Select *Sum* and click **Apply**. This adds the amount for each grouping, as well as the grand total.
10. Click **Save**.
11. Name your report *Closed Sales This Quarter* and accept the auto-generated unique name.


12. Enter a description and choose the *My Personal Custom Reports* folder for now. In a later tutorial, you'll create a shared folder.
13. Click **Save**, then close the report.

Step 3: Create the Gauge Dashboard Component

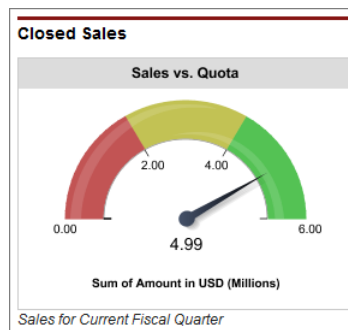
Let's create our first dashboard component, based on the report we just created. Gauge components distill complex information down to a single metric. Gauges are a great way to show how you're doing towards a goal.

1. Go back to the Sales Manager Dashboard we created earlier and click **Edit**.
2. Drag the gauge icon to the first column of the dashboard.
3. Click the Data Sources tab and start typing *Closed Sales This Quarter* into the search box.
4. When you find your report, drag it onto the gauge component. You've just created a dashboard component!
5. Click  to edit the component. You may notice that the default breakpoints don't make good choices, so let's change them in the Formatting tab.
 - Enter breakpoint values that make sense for your data. *Breakpoint 2* should represent the goal for your quota. The *Maximum* value should be a stretch goal. Once you cross the second breakpoint, you're in the green. In this example, the conditional highlighting looks like this:



- Click *Show Total* to display the sum of amount on the gauge.
 - Click **OK**.
6. Add a header, title, and footer by clicking those fields on the component:
 - Enter *Closed Sales* for the header. You can use headers to group components in each column.
 - Enter *Sales vs. Quota* for the title.
 - Enter *Sales for the Current Fiscal Quarter* for the footer.
-  **Note:** When you post a dashboard component snapshot to a Chatter feed, titles are shown, but headers and footers aren't. We'll try this out in a later tutorial.
7. Click **Save**, then close the dashboard.

The end result should look something like this—a gauge of closed sales for the current quarter against projected quota:



Step 4: Filter the Dashboard

Not all your users are interested in the same data. For example, some sales reps may need to see data restricted to the industry sector they work in. You can meet that need with a dashboard filter.

1. Edit the dashboard you just created and click **Add Filter**.
2. For **Field**, type *Industry*.
3. Click the selector and choose the industry categories your users might be interested in.

Edit Filter [Help for this Page](#)

Display Label
Industry

Filter Options

Operator	Value
contains	Banking, Finance, Healthcare, Insurance, Not For Profit, Retail
contains	Apparel, Hospitality, Recreation, Retail
contains	Biotechnology, Chemicals, Construction, Engineering, Environment

[Add Row](#)

OK Cancel

4. Click **OK** and save your dashboard.

Notice that now you have a filter drop-down at the top of the screen. Sales reps working in different market segments can use that to narrow down their view to the opportunities that matter most to them.

Summary

Congratulations! With just a few clicks and drags, you performed some meaningful analysis on your data, and you have the beginnings of a handy dashboard.

In this tutorial, we showed you the interdependency between dashboard components and their source reports, plus how to let users narrow the scope of the dashboard for themselves. In the next tutorial, we'll create a combination chart, plus a custom table of top salespeople—with photos! And we'll see how cross filters can help you flag neglected accounts.

TUTORIAL 2: CREATING COMBINATION CHARTS AND CUSTOM TABLE COMPONENTS

Level: Intermediate

We've got one dashboard component under our belt. The gauge component and its source report in the previous tutorial were pretty basic. In this tutorial, we're going to create two dashboard components from one report! Then we'll enable photos on the table so you can see sales reps' pictures right on the dashboard.

To achieve this, we'll create a source report that has a couple more summaries, as well as a chart, and we'll introduce you to the wonderful world of combination charts and custom table components. Plus, we'll show you how to use cross filters to keep an eye out for neglected accounts.

Step 1: Create the Sales Manager Leader Board Report

As in the previous tutorial, we first have to create the underlying source report. In this step, we'll create a summary report showing closed sales for the current month.

1. Click the Reports tab.
2. Click **New Report**.
3. Choose the *Opportunities* report type and click **Create**.
4. Apply the following filters:
 - Select *All Opportunities* for *Show*.
 - Select *Closed* for *Opportunity Status*.
 - Select *Close Date* for *Date Field*.
 - Select *This Month* for *Range*.
5. Click **Tabular Format** and change the report format to **Summary**.
6. Group the report by *Opportunity Owner* by dragging that field into the grouping drop zone.
7. Click the menu for the *Amount* column and select **Summarize this Field**. Select *Sum* and *Average* and click **Apply** to add these summaries to each grouping, as well as at the grand total level.
8. Click **Show** and deselect **Details**. Also click **Hide** in the unused secondary grouping drop zone to clean up the view. The report preview should look something like this:

Preview		Summary Format	Show	Add Chart	Remove All Columns
		Amount			
Opportunity Owner: Anna Bressan (4 Records)		USD 685,243.67 avg USD 171,310.92			
Drop a field here to create a grouping. Hide					
Opportunity Owner: Jin Chang (3 Records)		USD 568,350.00 avg USD 189,450.00			
Opportunity Owner: Karen Adams (1 Record)		USD 150,000.00 avg USD 150,000.00			
Opportunity Owner: Matt Wilson (3 Records)		USD 41,000.00 avg USD 13,666.67			
Opportunity Owner: Phil Smith (1 Record)		USD 383,250.00 avg USD 383,250.00			
Grand Totals (12 records)		USD 1,827,843.67 avg USD 152,320.31			

Note that the report builder preview only shows a limited number of records, so your report may look different when you run it.

9. Click **Save**.
10. Name your report *Sales Manager Leader Board* and accept the auto-generated unique name.
11. Enter a description and choose the *My Personal Custom Reports* folder.
12. Click **Save**.

Step 2: Add a Combination Chart on the Report

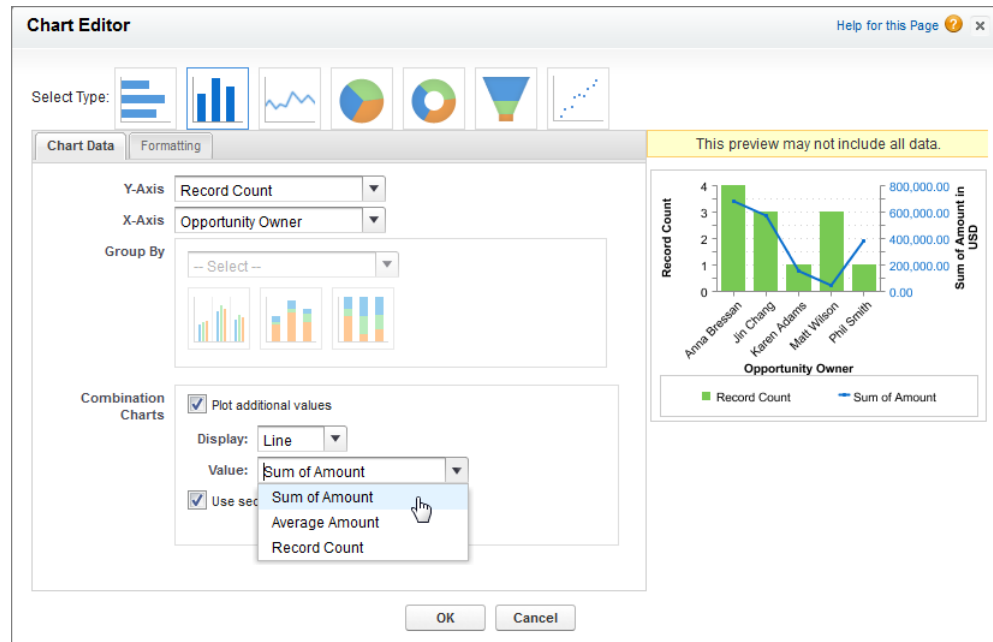
We're close to wrapping up this report. For most dashboard components, the source can be any summary or matrix report with at least one grouping, or tabular report with a row limit and sort order applied. But to create a custom table component on a dashboard, you first have to add a chart to the source report. Columns shown in the dashboard table come from the report chart. That is, if they're not in the chart, you can't use them in the table.

Tip: Creating a combination vertical column chart is a great way to make more values available in custom table components. You can add up to three more columns, or one line.

1. Go back to the Sales Manager Leader Board report we just created and click **Edit**, then **Add Chart**.
2. Select the Vertical Bar Chart type (also known as a column chart).
3. Choose *Record Count* for the *Y-Axis*.
4. Choose *Opportunity Owner* for the *X-Axis*.
5. Select *Plot additional values* to create a combination chart.
6. Select *Line* for *Display*. This adds a line to your vertical column chart.

Notice that the only option under *Value* is Record Count. This is because we chose Record Count for the *Y-Axis*. With a single axis, it doesn't make sense to show other types of data on the chart.

7. Now, select *Use second axis* and look at the *Value* drop-down. Aha! Now you see both Record Count and the two summaries you created on the report: Sum of Amount and Average Amount.
8. Select *Sum of Amount* for *Value*. The chart editor should look like this:



9. Click **OK**

10. Click **Save** and close the report.

Step 3: Create the Top Five Salespeople Custom Table Component

The report we created is valuable on its own: It shows closed sales by opportunity owner, with the sum and average of the amount. It also presents two completely different types of data on a single chart by using a two-axis combination chart! Data mavens will love it. They can refine the filters, drill into smaller segments, and do some deep analysis.

But what about the people who just want quick views into their team or organization? The power and versatility of dashboard components is that you can present multiple views of the same data, for different purposes. In this step, we'll create a custom table component showing the top five salespeople for the current month.

1. Go back to the Sales Manager Dashboard we created earlier and click **Edit**.
2. Drag the table icon to the second column of the dashboard.
3. Click the Data Sources tab and start typing *Sales Manager Leader Board* into the search box.
4. When you find your report, drag it onto the table component. Notice that the table has two columns showing the first grouping and summarized field used in the chart of the source report.
5. Click to edit the component.
6. Click **Customize table**. Now you get to choose values for up to four columns, show totals, and set the sort order! Remember that custom tables require the source report to have a chart, because chart values determine the available column choices.
 - For the first column, choose *Opportunity Owner*.
 - For the second column, choose *Sum of Amount*. Select *Sort Descending* and *Show Total*. The table will show salespeople with the highest sales at the top, and will calculate the total for all salespeople in the table.
 - For the third column, choose *Record Count*. This shows the number of sales for each opportunity owner. Select *Show Total* as well.


- For the fourth column... wait, where's *Average Amount*? Remember that we only added the sum to the chart—not the average—so it's not available to use in the custom table. If you want to use a value as a column, figure out a way to add it to the chart.
7. Enter *5* for *Maximum Values Displayed*. This will show only the top five salespeople based on the sort order you chose.
 8. Select *Show Chatter Photos*. This displays each opportunity owner's profile picture next to his or her name. This option is available only on tables and horizontal bar charts.
 9. Optionally, set conditional highlighting to mark ranges by color.
 10. On the Component Data tab, set *Drill Down to* to *Record Detail Page*. Each salesperson's name becomes a link that dashboard viewers can click to go directly to that salesperson's record detail page. If you don't set the drill-down option, clicking the dashboard component takes you to the source report.
 11. Click **OK**.
 12. Add a header, title, and footer by clicking those fields on the component:
 - Enter *Performance* for the header.
 - Enter *Closed Deals Leader Board* for the title.
 - Enter *Top 5 Salespeople - Current Month* for the footer.
 13. Click **Save**.

The end result will look something like this—a three-column table of sales leaders:

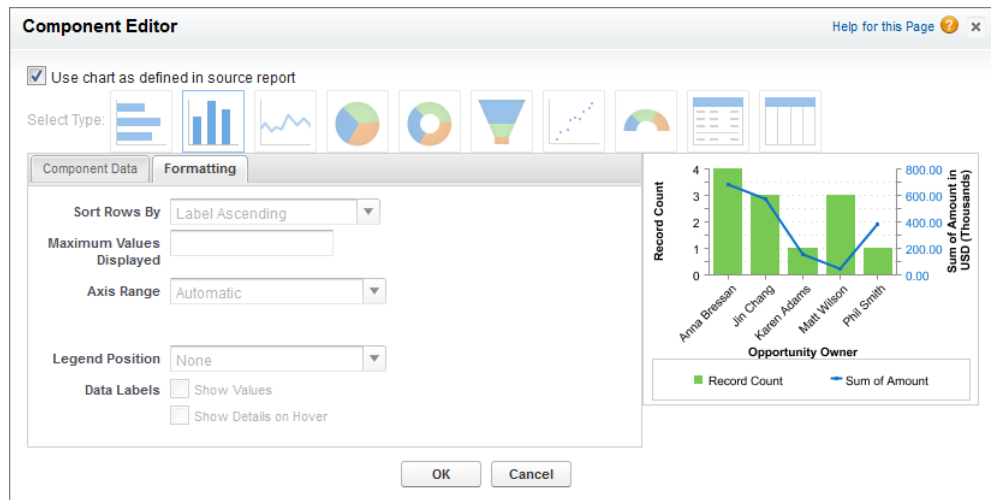
Performance		
Closed Deals Leader Board		
Opportunity Owner	Sum of Amount	Record Count
 Anna Bressan	USD 685K	4
 Jin Chang	USD 568K	3
 Phil Smith	USD 383K	1
 Karen Adams	USD 150K	1
 Matt Wilson	USD 41K	3
Total	USD 1.8M	12
Top 5 Salespeople - Current Month		

Step 4: Add the Combination Chart to the Dashboard

This step is easy, since we've already done the heavy lifting in the report. If you've created exactly the chart you want in the source report, there's an option to reuse that chart in your dashboard component. This is especially handy for joined reports, which you can't show in dashboard components any other way.

1. Edit the Sales Manager Dashboard we created earlier.
2. Drag any component type to the second column of the dashboard, just below the Closed Deals Leader Board component we just created.
3. Click the Data Sources tab and start typing *Sales Manager Leader Board* into the search box. Notice that we're using the same report as we did for the table of top five salespeople.
4. When you find your report, drag it onto the component.
5. Click  to edit the component.

6. Select the `Use chart as defined in source report` option. All of the choices in the component editor are disabled except the display units and drill behavior. We could leave that option unchecked and configure the component data right here with the dashboard builder, but for now we'll stick with what we've already set up in the report builder.



7. Click **OK**.
8. Add a header, title, and footer by clicking those fields on the component:
 - Enter *Closed Sales* for the header.
 - Enter *Deals/Amounts by Owner* for the title.
 - Enter *# of Deals and Sum of Amount - Current Month* for the footer.

9. Click **Save**, then close the dashboard. That's it!

The end result will look something like this—a two-axis combination chart of deals and amounts by owner:



Step 5: Create a Neglected Accounts Report With Cross Filters

Sometimes it's useful to remind sales reps of deals they should be paying attention to. You can find those quickly with a cross filter that returns open opportunities that don't have any pending activities. Add that to the dashboard so the sales team can avoid neglecting potential deals.

1. Save a new copy of the Sales Manager Leader Board report by clicking **Save As**. Name the new report *Sales Manager Neglected Accounts Report*, enter a description and put it in the *My Personal Custom Reports* folder.

2. Set Opportunity Status to *Open*. For the Date Field, select *Close Date*, and set the Range to *Current and Next FQ*. (The From and To date fields populate themselves automatically to reflect the date range you choose.)
3. Click **Filters > Add > Cross Filter**. Configure the cross filter to find Opportunities without Activities.
4. Without what kind of activities? Let's narrow our query down with a subfilter. Click **Add Activities Filter** and set it to *Date equals*. In the date field, type in *Next 30 days*.



Tip: You can type in almost any date range in plain English and get results. For a full list of the phrases that work, search for “special date values” in the online help.

5. Click **OK** and run the report. You’ve just answered the question, “Which open opportunities don’t have any activities scheduled within the next 30 days?” The report should look something like this:

Filtered By: [Edit](#)
 Opportunities without Activities [Clear](#)
 Date equals next 30 days

Opportunity Name	Type	Lead Source	Amount	Close Date	Next Step	Stage	Probability (%)
Opportunity Owner: Anjana Shah (3 records)							
			USD 4,565,250.00 avg USD 1,521,750.00				
Ecotech Germany - 10 Spider 3 Series Laptops	New Business	Trade Show	USD 240,000.00	5/28/2012	-	Proposal/Price Quote	80%
TMI - 40 free memory upgrade	Existing Business	-	USD 3,504,000.00	6/21/2012	-	Prospecting	10%
HomeNet Technologies - Laptops	Existing Business	Advertisement	USD 821,250.00	6/23/2012	-	Qualification	10%
Opportunity Owner: Anna Bressan (1 record)							
			USD 20,000.00 avg USD 20,000.00				
Starlux Hotels & Resorts - Desktops	Existing Business	Partner	USD 20,000.00	6/3/2012	-	Value Proposition	50%
Opportunity Owner: David Hudson (12 records)							
			USD 12,792,875.00 avg USD 1,066,072.92				
ZiffCorp - Laptops	Existing Business	Partner	USD 20,000.00	5/23/2012	-	Proposal/Price Quote	80%
House Central - Laptops	Existing Business	Word of mouth	USD 20,000.00	5/30/2012	-	Proposal/Price Quote	80%
Ecotech - Printers	Existing Business	Word of mouth	USD 2,750,000.00	6/16/2012	-	Needs Analysis	20%
EarthNet - 20 Desktops	New Business	-	USD 1,125,000.00	6/30/2012	-	Prospecting	10%
Vandelay - Printers	Existing Business	Employee Referral	USD 16,000.00	6/28/2012	-	Prospecting	10%
Starfish Publishing - Desktops	New Business	Advertisement	USD 1,650,000.00	6/12/2012	-	Needs Analysis	20%
Jam Television - free memory upgrade	Existing Business	Trade Show	USD 20,000.00	6/9/2012	-	Value Proposition	50%

Now sales reps can check the report for any pending opportunities for which they might want to schedule an email or phone call. To make them really happy, try creating a table component on the dashboard as you did in the previous step.

Summary

By spending the time to get the report just right, you make creating dashboard components that much easier. The steps in this tutorial showed how intertwined reports and dashboards are. For custom table components, the dependencies are even tighter; you can only choose what you’ve put in the chart. And if you just want to use the chart from the report, as is, you can.

We showed that dashboard components can slice and dice the same data from a single report to yield more than one meaningful view into your data. We also illustrated the power of cross filters to find records that don’t have a related record. In the next tutorial, we’ll create a matrix report and dive into custom summary formulas that perform calculations using summary values for groupings. You won’t know how you did without them.

TUTORIAL 3: CREATING A MATRIX REPORT WITH CUSTOM SUMMARY FORMULAS

Level: Intermediate/Advanced

In previous tutorials, we based our work on summary reports, which are great, since they let you group rows of data by a specific field. Matrix reports let you group by row *and* column, giving you an extra dimension to your data analysis. Use them to see how one dimension behaves against another. For example, you can analyze sales by month and location to spot seasonal sales trends in different locales.

In this tutorial, we'll create a matrix report that shows the monthly trend in revenue by type of business. We'll also create formulas that use report summary functions to calculate the relative size of each month's business against the total, and compare each month's total sales against the previous month. Finally, we'll create a dashboard component showing a cumulative line chart of each business type, and another dashboard component based on a report using buckets, showing where each type of business is coming from.

Step 1: Create the Revenue Trend by Type Matrix Report

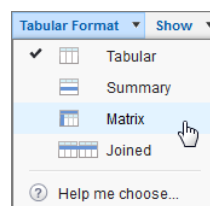
Let's start by creating the basic report. In this step, we'll create a matrix report showing sales by type for each month.

1. On the Reports tab, click **New Report**, choose the *Opportunities* report type, and click **Create**.
2. Apply the following filters:
 - Select *All Opportunities for Show*.
 - Select *Closed Won* for *Opportunity Status*.
 - Select *Close Date* for *Date Field*.
 - Select *Current FY* for *Range*. Choose the range that best fits the data you want to analyze.

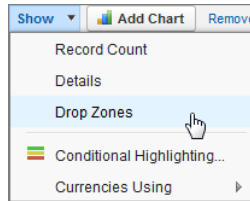


Tip: For the fastest results, always set the smallest date range you can. If your report has to sift through a great many dates, it can take longer to show the information you've asked for.

3. Click **Tabular Format** and change the report format to **Matrix**.



4. Group the report by *Type* by dragging that field into the column grouping drop zone.
5. Group the report by *Close Month* by dragging that field into the row grouping drop zone.
6. Click the menu for the *Amount* column and select **Summarize this Field**.
7. Select *Sum* and click **Apply**.
8. Click **Show** and deselect **Record Count**, **Details** and **Drop Zones** to clean up the view.



9. Click **Save**. The report preview should look something like this:

Preview

Matrix Format

Show

Add Chart

Remove All Columns

		Type	Existing Business	New Business	Grand Total
		Drop a field here to create a column grouping.			
Close Month	Drop a field here to create a row grouping.	Drop summarizable fields into the matrix.			
March 2012		Sum of Amount	USD 800,000.00	USD 40,000.00	USD 840,000.00
April 2012		Sum of Amount	USD 596,000.00	USD 947,718.35	USD 1,543,718.35
May 2012		Sum of Amount	USD 595,000.00	USD 998,500.00	USD 1,593,500.00
Grand Total		Sum of Amount	USD 1,991,000.00	USD 1,986,218.35	USD 3,977,218.35

10. Name your report *Revenue Trend by Type* and accept the auto-generated unique name.
11. Enter a description and choose the *My Personal Custom Reports* folder.
12. Click **Save**.

Step 2: Create a Formula Using the PARENTGROUPVAL Summary Function

Formulas are a bit more advanced, but don't let them scare you. They provide lots of value for a little work. Let's add our first formula to the matrix report we just created. The PARENTGROUPVAL function will help us calculate the percentage of business that each type contributes.

- Go back to the Revenue Trend by Type report we just created and click **Customize**.
- In the Formulas folder of the Fields pane, double-click **Add Formula**. This opens the Custom Summary Formula dialog.
- Define your formula:
 - Enter *Percentage of Total* for Column Name.
 - Enter *Calculates the % each type contributes to the total* for Description.
 - Select *Percent* for Format.
 - Select *0* for Decimal Places.
- Choose where to show your formula results:
 - Select *At a specific row/column grouping level*. For formulas containing summary functions, you have to select a specific grouping level.
 - Select *Row Grand Summary* for option A. This says to display the results of the formula at the bottom of the report, along with the subtotals.
 - Select *Type* for option B. This says to calculate the formula for each of the different business types.

5. Now, let's write the formula. We want to calculate the percentage that the total of each business type contributes to the grand total. In other words, divide the subtotal for each type by the grand total. Since the total is a calculated value, we have to create a formula using the PARENTGROUPVAL summary function.
 - a. Under Formula, click **Summary Fields > Amount > Sum**. This adds `AMOUNT : SUM` to the formula. This is the sum at the grouping level you specified above—that is, the subtotal for each type.
 - b. Click **Operators > / Divide**. We want to divide each subtotal by the grand total, which we'll enter as a summary function.
 - c. Under Functions, select *Summary*, then *PARENTGROUPVAL*.
 - d. Choose *Row Grand Summary* and *Column Grand Summary* for the options, then click **Insert**. Your formula should look like this so far:

```
AMOUNT:SUM/PARENTGROUPVAL(summary_field, ROW_GRAND_SUMMARY, COLUMN_GRAND_SUMMARY)
```

- e. The `summary_field` text is just a placeholder. Replace it with `AMOUNT : SUM`. This defines which value to use at which grouping level. That is, the grand total for amount.

Custom Summary Formula

Column Name:

Description:

Format: Decimal Places:

Where will this formula be displayed?

☐ At all summary levels

☒ At a specific row/column grouping level...

A Row Grand Summary

B Type

Formula **Summary Fields** **Operators** **Check Syntax**

AMOUNT:SUM/PARENTGROUPVAL(AMOUNT:SUM, ROW_GRAND_SUMMARY, COLUMN_GRAND_SUMMARY)

Functions **Summary** **PARENTGROUPVAL**

Select parent grouping level

Row Grand Summary

Column Grand Summary


PARENTGROUPVAL(summary_field, parent_row_grouping, parent_column_grouping)

< Insert

Help on this function

OK Cancel

6. Click **OK**. We've created our first formula! You'll see the results in the bottom row. Your report preview should look something like this:

Preview		Matrix Format	Show	 Add Chart	Remove All Columns		
		Type	Existing Business	New Business	Grand Total		
		Drop a field here to create a column grouping.					
Close Month	Drop a field here to create a row grouping.	Drop summarizable fields into the matrix.					
March 2012		Sum of Amount	USD 800,000.00	USD 40,000.00	USD 840,000.00		
April 2012		Sum of Amount	USD 596,000.00	USD 947,718.35	USD 1,543,718.35		
May 2012		Sum of Amount	USD 595,000.00	USD 998,500.00	USD 1,593,500.00		
Grand Total		Sum of Amount	USD 1,991,000.00	USD 1,986,218.35	USD 3,977,218.35		
		Percentage of Total	50%	50%			

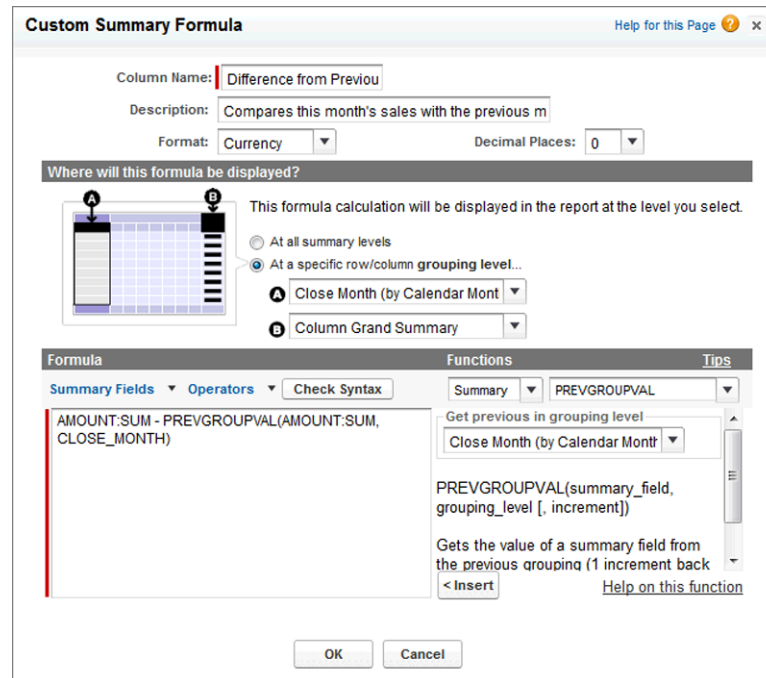
7. Click **Save**.

Step 3: Create a Formula Using the PREVGROUPVAL Summary Function

Now, let's add our second formula. The PREVGROUPVAL function will help us calculate the change in sales for each month versus the previous.

1. Go back to the Revenue Trend by Type report and click **Customize**.
2. In the Formulas folder of the Fields pane, double-click **Add Formula**.
3. Define your formula:
 - Enter *Difference From Previous Month* for *Column Name*.
 - Enter *Compares this month's sales with the previous month* for *Description*.
 - Select *Currency* for *Format*.
 - Select *0* for *Decimal Places*.
4. Choose where to show your formula results:
 - Select *At a specific row/column grouping level*.
 - Select *Close Month (by Calendar Month)* for option A. This says to calculate the formula for each month.
 - Select *Column Grand Summary* for option B. This says to display the results of the formula at the right of the report, along with the monthly totals.
5. Now, let's write the formula. We want to calculate the difference between the subtotals for each month and the month before it. In other words, subtract the previous month's total from the current month's. We'll do this using the PREVGROUPVAL summary function.
 - a. Under Formula, click **Summary Fields > Amount > Sum**. This adds `AMOUNT : SUM` to the formula. This is the sum at the grouping level you specified above—that is, the subtotal for each month.
 - b. Click **Operators > - Subtract**.
 - c. Under Functions, select *Summary*, then *PREVGROUPVAL*.
 - d. Choose *Close Month (by Calendar Month)*, then click **Insert**. Your formula should look like this so far:

`AMOUNT : SUM-PREVGROUPVAL(summary_field, CLOSE_MONTH)`
 - e. The `summary_field` text is just a placeholder. Replace it with `AMOUNT : SUM`. You can specify an increment if you want to compare to a value more than one month prior. The default increment is one, and the maximum is 12. For this step, we want the default increment.



Custom Summary Formula

Column Name:

Description:

Format: Decimal Places:

Where will this formula be displayed?

☐ At all summary levels

☒ At a specific row/column grouping level...

A

B

Formula:

Functions:

Get previous in grouping level:

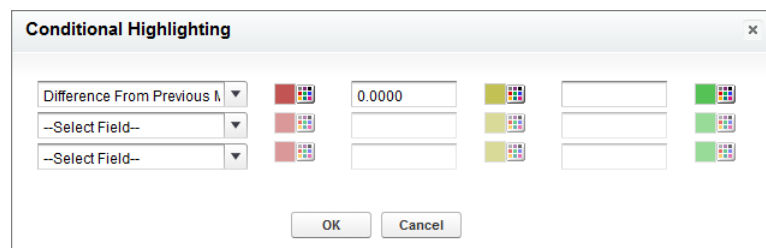
PREVGROUPVAL(summary_field, grouping_level [, increment])

Gets the value of a summary field from the previous grouping (1 increment back)

< Insert Help on this function

OK Cancel

6. Click **OK**.
7. Click **Show > Conditional Highlighting**.
8. Let's make it easy to see positive and negative values by assigning them colors:
 - Select the formula field you just created.
 - Enter 0 for the first breakpoint.
 - Select the color red to display numbers below zero.
 - Select the color green to display numbers above zero.



Conditional Highlighting

OK Cancel

9. Click **OK**. Your report preview should look something like this:

Preview

Matrix Format

Show

Add Chart


Remove All Columns

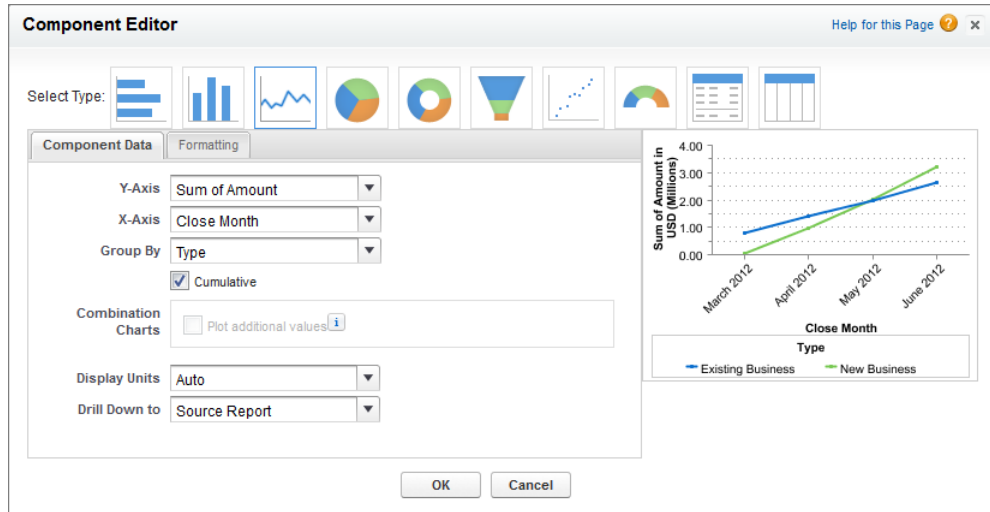
		Type	Existing Business	New Business	Grand Total
		Drop a field here to create a column grouping.			
Close Month	Drop a field here to create a row grouping.	Drop summarizable fields into the matrix.			
March 2012		Sum of Amount	USD 800,000.00	USD 40,000.00	USD 840,000.00
		Difference From Previous Month			
April 2012		Sum of Amount	USD 596,000.00	USD 947,718.35	USD 1,543,718.35
		Difference From Previous Month			USD 703,718.35
May 2012		Sum of Amount	USD 595,000.00	USD 998,500.00	USD 1,593,500.00
		Difference From Previous Month			USD 49,781.65
Grand Total		Sum of Amount	USD 1,991,000.00	USD 1,986,218.35	USD 3,977,218.35
		Percentage of Total	50%	50%	

10. Click **Save**.

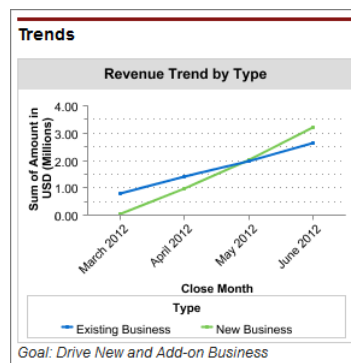
Step 4: Add a Cumulative Line Chart Component to the Dashboard

Now that we've finished creating the matrix report, let's add a custom chart component to the Sales Manager Dashboard that shows the trend in sales for each business type. We'll use a cumulative line chart, grouped by type.

1. Go back to the Sales Manager Dashboard we created earlier and click **Edit**.
2. Drag the line chart icon to the third column of the dashboard.
3. Click the Data Sources tab and start typing *Revenue Trend by Type* into the search box.
4. When you find your report, drag it onto the line chart component. Add a header, title, and footer by clicking those fields on the component:
 - Enter *Trends* for the header.
 - Enter *Revenue Trend by Type* for the title.
 - Enter *Goal: Drive New and Add-on Business* for the footer.
5. Click  to edit the component. On the Component Data tab, select the parameters you need.
 - Choose *Sum of Amount* for the *Y-Axis*. Notice how the summary field and summary formula you created in the report are available in the drop-down besides record count.
 - Choose *Close Month* for the *X-Axis*. The choices here are column and row groupings from the report.
 - Choose *Type* for *Group By*.
 - Click *Cumulative* to show the monthly amounts added together through time. The chart editor should look like this:



6. On the Formatting tab, set *Legend Position* to *Bottom*. This gives you more room to display your data.
7. Click **Save**. Your dashboard component should look something like this:



Step 5: Add a Bucket Field and Show It as a Pie Chart

So far, we've reported on how our team members are doing compared with each other and with their goals, and we've sorted out how much new business is in the pipeline compared with additions to existing business. But where is all this business coming from?

Let's use a bucket field to get a quick sense of our opportunities by market segment. A bucket is a grouping you create on the fly to bring together things that the fields in the report can't show you directly.


1. Go back and edit the Revenue Trend by Type report.
2. From the Bucket Fields folder of the Fields pane, double-click **Add Bucket Field**. This opens the Bucket Field dialog.
 - Select *Industry* for *Source Column*. We'll use that as a quick-and-dirty indicator of a customer's market segment.
 - Enter *Market Segment* for the bucket field name.

The bucket field dialog automatically shows you a detailed list of pre-loaded industry categories. We just want to let our executives see the overall picture at a glance, so we're going to group these into bigger, easier-to-manage categories.


3. Click **New Bucket** and create a bucket named *Manufacturing*, another bucket named *Technology*, and a third bucket named *Financial*.

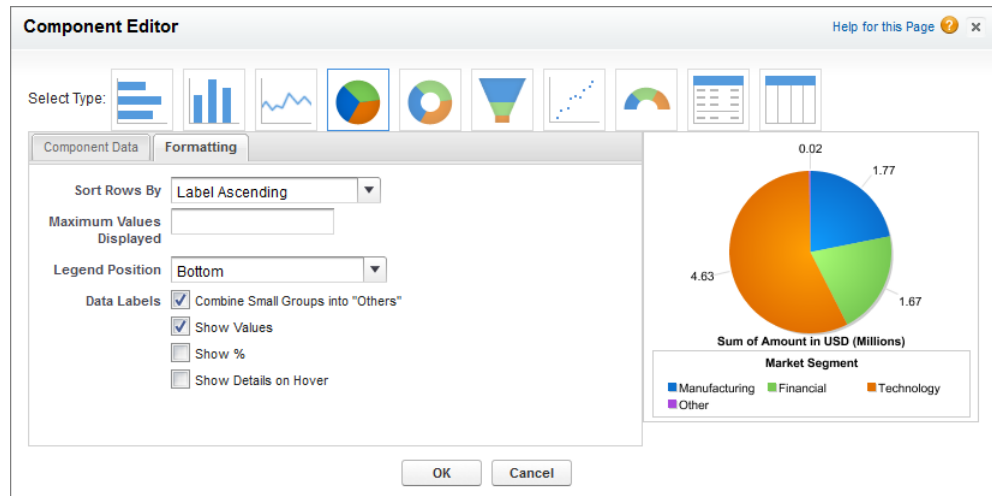
4. Drag each industry from the detailed list into the appropriate bucket. For example, "Banking," "Finance," and "Insurance" will all go in the Financial bucket. Select *Show unbucketed values as "Other."*
5. Drag your new bucket field into the column grouping drop zone, and save the report. Your *New Business* and *Existing Business* columns are now split into three columns each, showing how much of each type of business is coming from each industrial category.

Type		Existing Business				New Business		
Close Month	Market Segment	Manufacturing	Financial	Technology	Subtotal	Manufacturing	Financial	Technology
<input type="checkbox"/> March 2012	Sum of Amount Difference From Previous Month	USD 0.00	USD 0.00	USD 800,000.00	USD 800,000.00	USD 0.00	USD 0.00	USD 40,000.00
<input type="checkbox"/> April 2012	Sum of Amount Difference From Previous Month	USD 95,000.00	USD 200,000.00	USD 396,000.00	USD 691,000.00	USD 1,008,500.00	USD 570,500.00	USD 866,218.33
<input type="checkbox"/> May 2012	Sum of Amount Difference From Previous Month	USD 0.00	USD 500,000.00	USD 0.00	USD 500,000.00	USD 0.00	USD 0.00	USD 1,020,000.00
<input type="checkbox"/> June 2012	Sum of Amount Difference From Previous Month	USD 235,000.00	USD 25,000.00	USD 349,850.00	USD 609,850.00	USD 430,000.00	USD 389,500.00	USD 1,160,677.17
Grand Total	Sum of Amount Percentage of Total	USD 330,000.00	USD 725,000.00	USD 1,545,850.00	USD 2,600,850.00 32%	USD 1,438,500.00	USD 960,000.00	USD 3,886,885.50

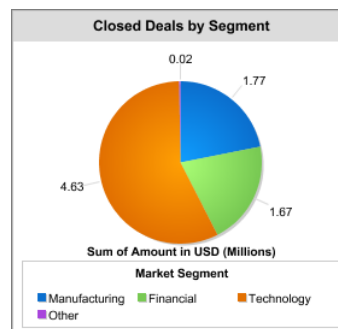
 **Tip:** You could have dragged the bucket field into the row grouping drop zone instead of the column grouping drop zone. Then you'd see how much each industry contributed to each month's revenue, rather than to each type of business. The overall totals would be the same.

Now let's put a graphical picture of this information on the dashboard for users to see.

1. Go back to our Sales Manager Dashboard and click **Edit**.
2. Drag the pie chart icon onto the dashboard, then drag the *Revenue Trend by Type* report from the Data Sources tab and drop it on the pie chart component.
3. Add a header, title, and footer by clicking those fields on the component:
 - Enter *Industry Groups* for the header.
 - Enter *Close Deals by Segment* for the title.
4. Click  to edit the component. On the Component Data tab, select the parameters you need.
 - Choose *Sum of Amount* for *Values*.
 - Choose *Market Segment* for the *Wedges*. (Your bucket field works just like any other grouping.)
 - Leave *Auto* for *Display Units*.
 - For *Drill Down To*, choose *Filtered Source Report*, so that users who click an area of the pie chart will go to the report filtered for that area. The dashboard component editor should look like this:



5. Click **Save**. Your dashboard component should look something like this:



You've just seen how to group data on the fly with a bucket field, and show it off as a pie chart on your dashboard. The data we bucketed here was from a picklist field; you can also bucket data by numerical ranges or text strings.

Summary

In this tutorial, we showed how the matrix format can take reporting to the next level. Matrix reports are great for showing summaries down columns and across groupings at the same time—like the sum of sales for each type and for each month. Though we didn't do it here, you can also add secondary column and row groupings to matrix reports.

We showed how summary functions offer a powerful way to use grouping values in report formulas. The `PARENTGROUPVAL` function lets you calculate values relative to a parent grouping, and `PREVGROUPVAL` relative to a peer grouping.

The formulas we used were fairly simple, but there's so much more you can do with these summary functions. In the next tutorial, we'll show you how you can use them to pull together data from multiple different reports.

And finally, we showed how the little things, like grouping by date, hiding details, and combining like fields into buckets on the fly, can make a report easier to understand.

TUTORIAL 4: CREATING A JOINED REPORT WITH CUSTOM SUMMARY FORMULAS

Level: Intermediate/Advanced

We've seen how you can answer complex questions about Salesforce data by summarizing down columns and across groupings at the same time. But what if the information you need doesn't fit in a single report?

In a joined report, you have what amounts to several different reports in one. Each report is called a "block." You can filter each block separately, just like a standalone report. But you can also compare values *across* blocks, to look at your data from multiple angles. Best of all, the blocks can look at different objects!

Step 1: Create the Pipeline Predictor Joined Report

To try this out, let's build a simple tool to track how well sales reps are capitalizing on their opportunities. We'll create a basic Opportunities report, add three blocks, filter each block by the appropriate status, and then group the results by sales rep.

1. Create a new report with the name Pipeline Predictor, selecting **Opportunities** as the report type.
2. Click **Tabular Format** and select **Joined**. Notice that your report data is now set off in a box with a colored border. That's your first block.
3. Create a new block by dragging the *Opportunity Name* field onto the preview pane somewhere to the right of the first block, and drag the *Amount* and *Account Name* fields onto the new block.

Did you notice a second filter panel appearing in the filter pane above the blocks? Each block in our joined report has its own independent set of filters.

4. Create a third block in the same way. Now we have three different Opportunities reports sitting next to each other in the same container. They're all identical right now, but we're going to use each one to tell us something unique.
5. We want all the blocks to give us comparable information, so prepare each block the same way:
 - a. Remove all the fields except *Opportunity Name*, *Amount*, and *Account Name*, by dragging them back over to the Fields pane. That'll help us focus on what's important here.
 - b. In the filter panel, set **Show** to **All Opportunities**.
 - c. Click the dropdown at the top right of each block and make sure **Record Count** is checked.
 - d. Click the dropdown next to *Amount*, click **Summarize this Field**, and select **Sum**.

6. Now let's configure each block to tell a different part of the story.

- We'll use the first block to show the opportunities that got away. Click the title (right now it just says "Opportunities block 1"), and rename it *Closed Lost*. In the **Closed Lost** filter panel above, set these filters:

Opportunity Status equals Closed

Date Field equals Close Date

Range equals Current and Previous FY

Stage equals Closed Lost

- We'll use the second block to show the opportunities we landed. Rename the block *Closed Won* and set these filters:
`Opportunity Status equals Closed Won`
`Date Field equals Close Date`
`Range equals Current and Previous FY`

- The third block will show us opportunities that are getting close to their anticipated closing. Rename the block *Closing Next Month* and set these filters:
`Date Field equals Close Date`
`Range equals Next Month`
`Opportunity Status equals Open`

Tip: For the fastest results, always filter as narrowly as you can. Filters that use `equals` will often return information more quickly than filters that use `contains`, `does not contain`, or `not equal to`. In complex reports, this can make a difference.

- Now we'll group all three blocks by sales rep. Drag the `Opportunity Owner` field to the horizontal bar that says, "Drop a field here to group across report blocks." (This is called the grouping drop zone.)
- Click **Save**, then **Run Report**.

We've just created a handy resource that sales reps can use to monitor their track record and to stay on top of deals that are heating up. Next, let's see if we can wring a little more information out of this data with custom formulas.

Step 2: Add a Custom Cross-Block Formula

We've seen how to track each sales rep's performance with a simple scorecard. How can we use this data to understand our overall pipeline progress? Let's turbocharge our joined report with a win ratio for each rep. Then we can use that to offer predictions about pending deals based on past performance.

- Go back to the Pipeline Predictor joined report we just created and click **Customize**.
- In the Formulas folder of the Fields pane, double-click **Add Cross-Block Formula**.
- Name your new formula field *Win Ratio* and configure it to divide the number of deals won by the number of deals lost. The formula will look like this:

```
[Closed Won]RowCount / ([Closed Lost]RowCount + [Closed Won]RowCount)
```

- Add the formula to each of the blocks.
- Double-click **Add Cross-Block Formula** again. Name this formula field *Predicted Revenue* and configure it to multiply the total value of the deals closing next month by the overall ratio of deals won to deals lost. That will look like this:

```
[Closing Next Month]AMOUNT:SUM * ([Closed Won]RowCount / ([Closed Lost]RowCount + [Closed Won]RowCount))
```

- Add the formula to the last block.



Tip: These custom summary formulas are just examples. Now that you know how, you'll probably want to adjust them to work best with the data you're analyzing—or create new ones altogether.

7. Click **Run Report**.

Report
Opportunity Pipeline Predictor Help for this Page ?

« Go to Report List

Show Chatter Follow

Run Report Show Details Report Properties Save Save As Customize Delete Report generation complete.

	OPPORTUNITIES Closed - Lost		OPPORTUNITIES Closed - Won			OPPORTUNITIES Closing Next Month		
	Record Count	Amount	Record Count	Amount	Oppty Won Ratio	Record Count	Amount	Predicted Revenue
Opportunity Owner: Anjana Shah	1	USD 75,000.00	1	USD 225,000.00	0.50	2	USD 4,325,250.00	USD 2,162,625.00
Opportunity Owner: Anna Bressan			2	USD 10,833.33	1.00	3	USD 37,000.00	USD 37,000.00
Opportunity Owner: David Hudson	2	USD 2,329,500.00	4	USD 290,000.00	0.67	5	USD 6,357,375.00	USD 4,238,250.00
Opportunity Owner: Karen Adams	1	USD 4,268,227.62	0	0.00		1	USD 511,000.00	USD 0.00
Opportunity Owner: Matt Wilson	7	USD 689,935.17	4	USD 1,955,000.00	0.36	5	USD 4,169,325.00	USD 1,516,118.18
Opportunity Owner: Phil Smith	1	USD 850,000.00	6	USD 241,500.00	0.86	3	USD 531,243.67	USD 455,351.72
Grand Totals	12	USD 8,212,662.78	17	USD 2,722,333.33		19	USD 15,931,193.67	USD 9,338,975.60

We've just created two custom formulas that look at data in different blocks of a joined report and produce information we can use to help reps monitor their performance and drive more revenue! Now we'll see how we can extend our view to include data from multiple report types.

Step 3: Build a Sales Rep Scorecard

The real power of joined reports emerges when you want to take one element of your organization and find out everything you can about it. For example, you might want to know everything that's going on for each individual sales rep on your team.

To do this, let's create a report that looks at all the different kinds of records sales reps work with. We'll bring together the opportunities in their pipeline, the activities they've scheduled, and the open support cases for accounts they're working with. With one report, we'll know at a glance what's on everyone's plate.

1. Create a new report, selecting **Opportunities** as the report type, and call it "Sales Rep Scorecard."
2. Click **Tabular Format** and select *Joined*.
3. This time we want our blocks to look at entirely different types of information.
 - a. Click **Add Report Type** and select **Tasks** and **Events** from the **Activities** folder, and click **OK**.
 - b. Click **Add Report Type** again and select **Cases** from the **Customer Support Reports** folder.

Notice that each of your three blocks is marked off with a border of a different color, to remind you that you're looking at three different sources of information. And as before, each block has its own filter panel above.

4. For convenience, let's clean up each block by removing fields we don't need. For example, in the Opportunities block, you might want to remove all fields except **Opportunity Name**, **Amount**, and **Account Name**.
5. To reduce distraction, let's filter out old data.
 - a. In the filter panel for Opportunities, set **Opportunity Status** to *Open*.
 - b. In the Tasks and Events filter panel, set **Show** to *Open Activities*.

- c. In the Cases filter panel, click **Add** and specify *Status does not contain*, and select all the statuses that include "Closed."
- d. For Opportunities and for Tasks and Events, set **Range** to *Current FQ*.
6. Click the dropdown at the top right of each block and make sure **Record Count** is checked.
7. In the Opportunities block, click the dropdown next to *Amount*, click *Summarize this Field*, and select **Sum**.


Now we're set up to see the power of a joined report. Notice how the **Fields** panel is divided into sets of fields that belong to each of the Salesforce objects our report is looking at. The **Common Fields** section lists the fields that make sense for all the objects in the reports. We can use any of these fields to group across all of our blocks.

8. Drag the *Account Owner* field to the grouping drop zone.

Now each block shows who is working with the records in that block. Users can scan across and get a full picture of what's happening for each person listed.

Preview Joined Format Show Add Chart								
OPPORTUNITIES Opportunities block 1			TASKS AND EVENTS Tasks and Events block 2			CASES Cases block 3		
Opportunity Name	Amount	Account Name	Subject	Date *	Company / Account	Case Owner	Subject	Account Name
Drop a field here to group across report blocks.								
Data Mart - 45K	\$12,000.00	Data Mart	Other	8/2/2012	S & S Canopies & Camper Mfg	Brenda Support	Browser settings	Associated Supply Co.
Data Mart - 27K	\$10,000.00	Data Mart	Schedule Discovery Call	8/3/2012	Harmon Consulting	Admin User	Taking a long time to login	Net Inc.
UlyssesNet - 3K	\$3,750.00	UlyssesNet	Follow up on status	8/3/2012	Advanced Interconnections Corp*	Linda Support	Need guidance getting started with my installation	Employnet
Value America - 25K	\$25,000.00	Value America	Travel time	8/3/2012	Southern Research Co.	Jay Support	Updating data	Tyconet
DFC Inc. - 12K	\$12,000.00	DFC Inc.	Customer Feedback	8/4/2012	-	Jay Support	Can't understand installation instructions	Optos Inc.
UlyssesNet - 10K	\$10,000.00	UlyssesNet	Manager deal update	8/4/2012	United Partners*	Steven Support	Username does not appear	Steam Power Company
Q2 2015 Oppty	\$600,000.00	Haven Enterprises	2nd Call	8/4/2012	Associated Supply Co.	Jay Support	Billing issue address	Valley Supply Inc.
Q1 2015 Oppty	\$600,000.00	Haven Enterprises	Left voicemail	8/4/2012	DFC Inc.	Admin User	Can't connect to the server	Towson Inc.
Q3 2015 Oppty	\$550,000.00	Haven Enterprises	Touch base	8/4/2012	Employnet	Admin User	Can't connect to the server	Towson Inc.
Optos Inc. - 23K	\$23,000.00	Optos Inc.	New notes	8/4/2012	Homeland Product Co.	Admin User	-	Towson Inc.
Value America - 10K	\$10,000.00	Value America	Review Case	8/5/2012	Advanced Interconnections Corp*	Admin User	-	Towson Inc.
Valley Supply Inc. - 10K	\$10,000.00	Valley Supply Inc.	Send Thank You	8/5/2012	UlyssesNet	Admin User	-	Open Source Inc.
Larry Baxter - 7K	\$6,499.00	Larry Baxter	Schedule Negotiation Call	8/5/2012	Gusto Distributing	Steven Support	-	Associated Supply Co.
United Partners* - 61K*	\$61,229.00	United Partners*	New notes	8/5/2012	Open Source Inc.	Linda Support	-	United Partners*
Data Mart - 1K	\$12,000.00	Data Mart	Follow up to my message	8/5/2012	Advanced Interconnections Corp*	Admin User	-	United Partners*
Data Mart - 45K	\$11,000.00	Data Mart	Send new pricing	8/5/2012	Vand Enterprises, Inc.	Steven Support	-	Larry Baxter
Data Mart - 1K	\$5,000.00	Data Mart	Qualification notes	8/5/2012	Optos Inc.	Admin User	-	Haven Enterprises
Data Mart - 27K	\$16,000.00	Data Mart	Product demo with team	8/5/2012	United Partners (San Francisco)	Brenda Support	-	Net Inc.
Advanced Interconnections Corp* - 49K*	\$49,300.00	Advanced Interconnections Corp*	Needs list from Prospect	8/5/2012	Value America	Admin User	-	Employnet
United Partners (VA) - 22K	\$22,500.00	United Partners (VA)	Manager deal update	8/5/2012	Big Sky & Sons	Jay Support	-	Universal Services

9. This report might be even easier to read if we could also tell which tasks and cases are related to which opportunities. We can do that with a sub-grouping. Drag the *Parent Account* field from the **Fields** pane to the secondary grouping drop zone.

 **Tip:** Try different cross-block groupings and sub-groupings to highlight the information you're most interested in. For example, if you wanted to compare what's going on in different accounts, you could use *Account Name* instead of *Account Owner* for your top-level grouping.

10. Click **Run Report**.

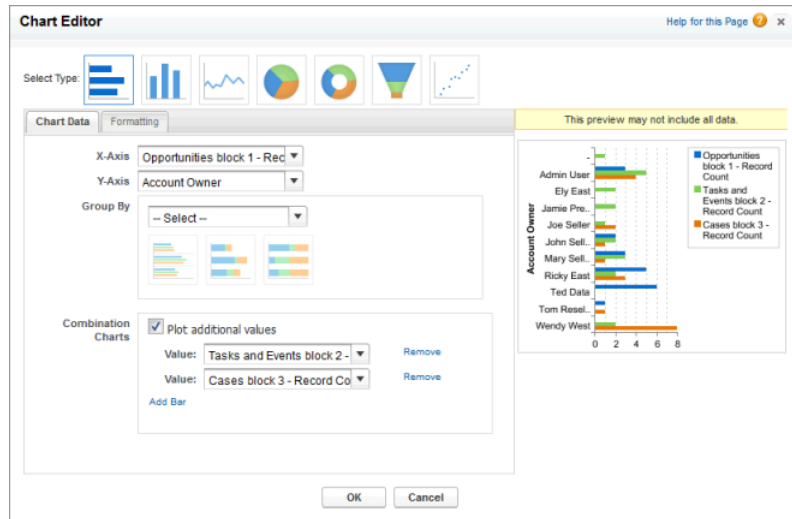
We've just created a kind of bulletin board that joins multiple types of information in useful ways. People can tell at a glance how their pipeline is looking, what upcoming tasks they should be preparing for, and what support activities might affect their ability to close deals.

Next, we'll share this with the team in the form of a chart on our sales dashboard, which they can follow in their Chatter feed for regular updates.

Step 4: Add the Sales Rep Scorecard to the Dashboard

We've seen in the last two tutorials how you can present chart data on a dashboard for easy access. A joined report can appear as a dashboard component too, but you design its appearance in the report builder, not on the dashboard.


1. Back in the Sales Rep Scorecard report that we just created, click **Customize > Add Chart**.
2. On the **Chart Data** tab, observe that the X (horizontal) axis measures the number of open opportunities for each of our sales reps, who are listed on the Y (vertical) axis. We can go with those defaults in this case.
3. Select **Plot additional values**. For the first additional value, select **Tasks** and **Events**.
4. Click **Add Bar** to choose yet another value. For this one, select **Cases**.



5. Click **OK**. you've just created a chart that sketches out the relative health of each sales rep's pipeline.

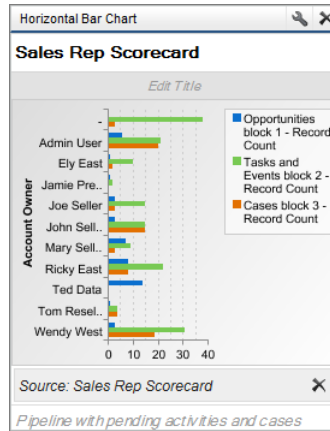


Tip: Other ways of showing this information graphically may serve better for the particular data you're working with. Just click **Edit Chart** and try different data and formatting combinations until you have exactly the representation you want.

6. Now let's share the report as a dashboard component. Go back to the sales manager dashboard that we created in [Tutorial 1](#) and click **Edit**.
7. Drag the **Sales Rep Scorecard** onto the dashboard from the Data Sources tab in the Reports pane.
8. Switch to the Components tab in the Reports pane and drag any of the component types onto the source report that you just added. An error message advises you that a dashboard component showing a joined report must use the chart that you designed in the report itself. (Other report types can be designed right on the dashboard.)
Click  and check that box on the Component Data tab. Now your bar chart appears as a dashboard component.

9. Let's add one more detail to help users make the most of this dashboard component. On the Component Data tab, find the **Drill Down To** dropdown list, select **Filtered Source Report**, and click **OK**.

Now sales reps can click their own names on the dashboard and go directly to a view of the underlying report that shows only their own need-to-know information!



Summary

We've seen how you can analyze complex situations by packaging several different reports as blocks of one joined report. We learned how to filter within blocks and group the results in useful ways. We did some advanced tricks with custom formulas, and we put the results on a dashboard for easy access.

In the next tutorial, we'll use historical trend reporting to look at changes in your data over time.

TUTORIAL 5: CREATING A PIPELINE PUSH REPORT

Level: Intermediate

Analyzing your latest data is great, but what if you need to know how things are changing over time? For example, suppose you want to monitor your company's sales pipeline to catch deals that keep getting pushed out.

To do that, you'll need to compare past and current values of key Opportunity attributes, such as dates, amounts, and status changes. You may want to focus on deals whose value grew or shrank, or deals that moved into or out of a given target period.

Step 1: Create a Simple Historical Report

Let's begin with a report that compares two dates. This report shows you the total value of your pipeline today compared with its value yesterday.


1. Create an opportunity history report.

- a. On the Reports tab, click **New Report**.
- b. Under **Select Report Type > Opportunities**, select *Opportunities with Historical Trending*.
- c. Click **Create**.

 **Note:** If *Opportunities with Historical Trending* doesn't appear, ask your administrator to enable historical trend reporting for your organization and set up historical trend reporting for the Opportunities object.

2. For Show, select *All Opportunities*.

The default value in the **Historical Date Field** is *Yesterday*, and the **Amount (Historical)** field shows yesterday's date.


 **Tip:** "Yesterday" is a *rolling* date value, meaning that it points to a date that is relative to today's date. If you run this same report tomorrow, **Amount (Historical)** will show today's date.

3. Click the down arrow in the headers of the **Amount (Historical)** and the **Amount—Today** columns. Click *Summarize This Field*, then select *Sum*.
The total amount appears at the bottom of each column.

4. Click the down arrow in the header of the **Amount** column and select *Show Changes*.

5. Click **Run Report**.

6. In the **Change** column, observe the difference between the total value of the **Amount (Historical)** column and that of the **Amount—Today** column.

 **Tip:** You can see changes at a glance by looking for values that are colored green or red.

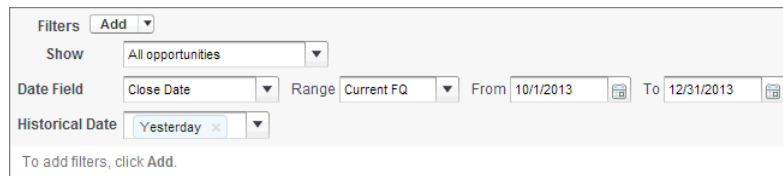
You've created a simple historical trending report to help you analyze your pipeline's behavior. Now you're ready to dig a little deeper into your historical data.

Step 2: Find Deals that Have Been Pushed Out


To focus on business that's taking more time to close than expected, add a historical filter that finds deals in your pipeline that have had their close dates moved to a later date.

1. Click **Add** and select *Historical Field Filter*.
2. Set `Close Date (Historical)` to yesterday's date, and click **OK**.

 **Tip:** Use the calendar under **Fixed Days** to select the date



3. Click the down arrow in the header of the `Close Date` column and select *Show Changes*.
4. Click **Run Report**.
5. For each deal in the report, compare the date in the `Close Date - Historical` column with the date in the `Close Date - Today` column.

 **Tip:** You can see changes at a glance by looking for values that are colored green or red.

You've created a report that can help you detect pushed opportunities. Now try turning it into a matrix report (as we saw in [Tutorial 3](#)) and adding it to your dashboard ([Tutorial 1](#))!

Summary

We've just built a simple report that shows how our pipeline is changing in value, and can help us identify deals that are being pushed or otherwise manipulated. In the next tutorial, we'll set up a simple sharing model so that the right people can see and edit our reports and dashboards. And we'll see how linking up a report or dashboard with a Chatter feed can multiply its value by helping others respond quickly to changes in our data.

TUTORIAL 6: SHARING AND COLLABORATION

Level: Beginner

In the previous tutorials, we created some pretty cool reports and dashboards. But how do we make sure the right people are seeing them? Creating your own security model for shared dashboard and report folders is a safe and easy way to protect access to sensitive data.


In this tutorial, we'll cover how to create folders and manage access to them, then we'll highlight Chatter features for reports and dashboards, like feed tracking, following, alerts, and posting dashboard component snapshots. And we'll embed a report chart right in a record detail page, so users can find valuable context without leaving the page they're working on.

Another way to share information is to schedule report and dashboard refreshes and select people to receive notifications. Those people get an email with an image of the report or dashboard. A nice option, but a far cry from the true, real-time collaboration that Chatter provides.

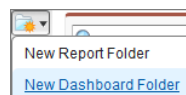
Step 1: Create a Shared Dashboard Folder

Before you begin, make sure that your administrator has enabled folder sharing for you.

In the first tutorial, you saved the dashboard to your personal folder. In this step, we're going to create a shared folder for our dashboard. That way we can make sure the right people have access to this dashboard—without giving the wrong people access to sensitive data.

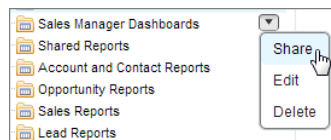
 **Note:** See [Permissions You Need](#) on page 2. If you can't create dashboard folders, skip this step. You can always save the dashboard to your private folder, or a public one.

1. Click the Dashboards tab. You'll see your most recently viewed dashboard.
2. Click **Go to Dashboard List**.
3. Open the drop-down in the Folders pane and create a new dashboard folder.

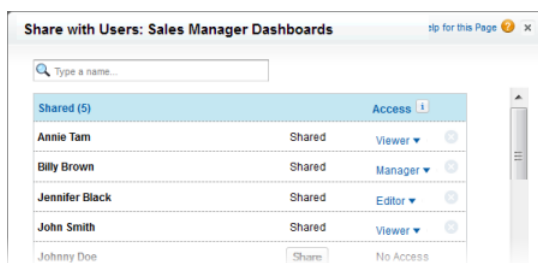


Enter *Sales Manager Dashboards* for the label, accept the auto-generated unique name, and click **Save**.

4. Click **Go to Dashboard List** again. In the dropdown list next to your new folder, click **Share**.



5. You'll probably want to share this folder with users who are sales managers. Click **Share with: Users** and pick the people you want. (To search for a user, just start entering a name.)



If your organization has a Sales Manager role set up, you can share the folder with that role. That's an easy way to give multiple people the same access level all at once. Just click **Share with: Roles** instead of **Users** and pick the role name from the list.

6. Give the right access level to each person (or role) you've shared this folder with. Your access level determines what you can do with dashboards in this folder.
 - With Viewer access you can see the data in a report or dashboard, but you can't make any changes, except by cloning it into a new report or dashboard. All users have at least Viewer access to report and dashboard folders that have been shared with them. (Some users may have administrative user permissions that give them greater access.)
 - When you are an Editor on a folder, you can view and modify the reports and dashboards it contains, and move them to and from any other folders you have Editor or Manager access to.
 - With Manager access, you can do everything Viewers and Editors can do, plus control other users' access to it, change its properties, or delete it.
7. Click **Save**. Now let's put our Sales Manager Dashboard into this new folder. On the Reports tab, drag the dashboard over to the Folders pane and drop it on the Sales Manager Dashboards folder.
8. If someone has access to a dashboard folder, but not to the report folder containing the source reports, they won't be able to see the data in the dashboard. Follow the steps on this page again, this time on the Reports tab, so you end up with a Sales Manager Reports folder shared with the same people. Then drag Closed Sales This Quarter, Sales Manager Leader Board, Revenue Trend by Type, Neglected Opportunities, Pipeline Predictor, and Sales Rep Scorecard onto the Sales Manager Reports folder in the Folders pane.

Congratulations! You've just created a space where selected users can work with data you share with them, in the ways that you've specified.

Step 2: Embed a Chart in a Record

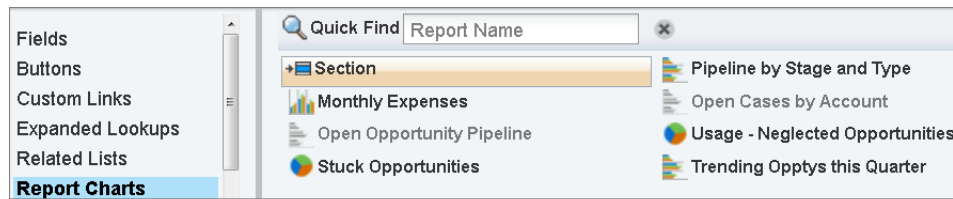
The Reports and Dashboard tabs are great one-stop shopping spots for data. But users don't have to go there every time they need to know something. You can set it up so people can see a report chart right on the page where they're working. For example, when updating an opportunity, they can see a chart that's filtered to show the latest data about the account that opportunity is associated with, without leaving the Opportunity detail page.

1. Pick the report you want to display in your record page. Make sure it already has a chart, and that it's in a folder that's shared with users who need access.
(We saw how to do this in [Step 1: Create a Shared Dashboard Folder](#) on page 31.)
2. Go to the page layout editor for the object you're adding a chart to. Let's use Opportunities for this example. From the object management settings for opportunities, go to Page Layouts.



Tip: You can do this with a custom object too. From the object management settings, select the custom object, and then scroll down to Page Layouts.

3. Click **Edit** next to the page layout.
4. Click **Report Charts**.



5. In the Quick Find box, type the name of the report and click to find and select the report chart. You can browse up to 200 recently viewed reports by chart type in the Report Charts palette.
6. Drag the chart onto the layout.
7. Click on the chart to customize it.
8. Click **Save** and go look at a record on the Opportunities tab. It will look something like this:



Congratulations! You've just saved some time and effort for everyone who works with Opportunities in your organization. You've made it so they can see data about an opportunity right on the page they're working on, without having to switch over to another tab.

Step 3: Turn on Feed Tracking for Reports and Dashboards

Once you enable Chatter for your organization, you can turn on feed tracking for reports and dashboards. When the value changes for a tracked field, Chatter posts an update to the feed, instantly alerting everyone following that report or dashboard.

1. From Setup, enter *Chatter Settings* in the Quick Find box, then select **Chatter Settings**.
2. Click **Edit**.

3. Select the `Enable` checkbox.
4. Click **Save**.

Chatter Settings [Help for this Page](#)

Chatter is a corporate network that lets your users work together, talk to each other, and share information, all in real time.

[Save](#) [Cancel](#)

Chatter Settings ! = Required Information

Turn on Chatter and Global Search features. We have given you a head start - your users may auto-follow a few people or records by default and your search box is in the header. [Learn More...](#)

Enable ☒

Email Notifications

Allow users to receive personal Chatter email notifications.

Allow Emails ☒

Now let's turn on feed tracking for reports and dashboards.

1. From Setup, enter *Feed Tracking* in the `Quick Find` box, then select **Feed Tracking**.
2. Select the `Dashboard` object from the list.
3. Select the *Enable Feed Tracking* checkbox.
4. Optionally, choose fields to track, like *Dashboard Running User*, *Title*, *Description*, and *Dashboard Unique Name*. If you select a field, and its value changes, that object's feed is updated, alerting everyone who's following it.
5. Click **Save**.
6. Repeat the steps to enable feed tracking for the Reports object.

Feed Tracking [Help for this Page](#)

Enable feed tracking for objects so users can follow records of that object type. Select fields to track so users can see feed updates when those fields are changed on records they follow.

Object	Tracked
Account	2 Fields
Applicant	
Asset	
Campaign	
Case	
Chatter Group	0 Fields
Contact	
Content	0 Fields
Contract	
Dashboard	
Event	

Fields in dashboards

[Save](#) [Cancel](#) ☒ **Enable Feed Tracking** [Restore Defaults](#)

You can select up to 4 fields.


Dashboard Running User	<input checked="" type="checkbox"/>	Dashboard Unique Name	<input type="checkbox"/>
Description	<input checked="" type="checkbox"/>	Title	<input checked="" type="checkbox"/>

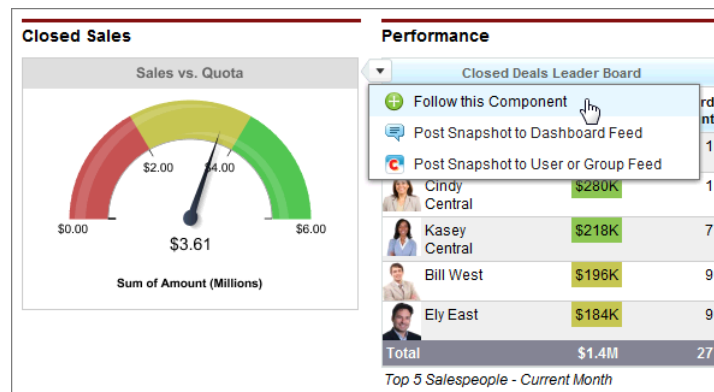
Step 4: Follow Your Favorite Dashboards and Get Alerts on Components

Now that you have Chatter feeds on dashboards, it becomes a meeting place where your team can post messages and comments and share relevant files, links, and other information. In this step, we'll follow the Sales Manager Dashboard so that updates are brought into your own feed, and walk through the steps for following a dashboard component.

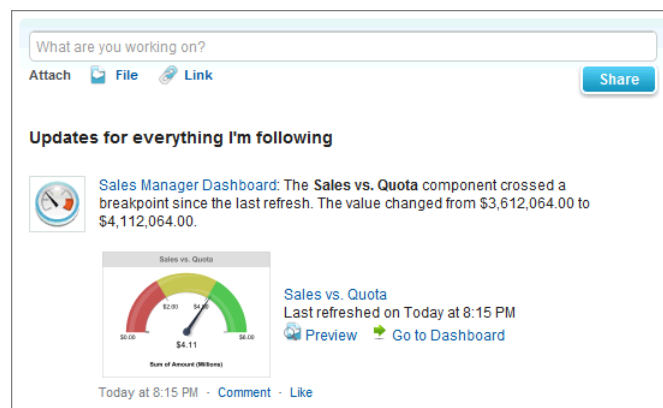
Following a dashboard is simple. Just find the one you want, and click **+ Follow**. When you follow a dashboard, you automatically follow all eligible components. When the dashboard is refreshed, if the value for a followed component crosses a breakpoint, an alert is posted to your Chatter feed. Following components isn't supported for dynamic dashboards.

Let's follow the first dashboard component we created—the gauge of closed sales—to receive alerts when the sales total crosses one of the conditional highlighting breakpoints.

1. Hover over a component to display the  menu. To clear the hover menu, click an empty part of the screen.
2. Click **Follow this Component**.



When an alert is triggered, you'll receive a Chatter post similar to this one:



Summary

An analysis is only as good as the data it's based on. Very true. But no matter how good your analysis, it's worthless if it doesn't prompt action, change behavior, or improve understanding. And you can't achieve any of those if the right people aren't seeing the results of your analysis—your reports and dashboards.

Folders give you complete control over access to their contents. If you store dashboards and reports in appropriately shared folders, your data is safe from inappropriate exposure. Chatter gives you feed tracking, following (for reports, dashboards, and components), alerts on metric and gauge components, and dashboard component snapshot posts. Embedding report charts right on record detail pages gives people a look at the broader picture without making them switch tabs.

Reports and dashboards make analysis easy. Folders, embedded charts, and Chatter make sharing and collaboration a cinch.

TUTORIAL 7: BUILDING CUSTOM APPS WITH THE SALESFORCE REPORTS AND DASHBOARDS API

Level: Advanced

You've learned how to do some pretty sophisticated things with reports in this workbook so far. But what if you need more? What if your users can't live without a custom app specially tailored to their own unique business requirements?

No problem! If you can code a little, or have access to someone who can, Salesforce provides an API that can handle almost anything you can do with reports and dashboards through the standard web interface.


For example, say you've used Visualforce to build a custom app, and you want that app to display report data. Or your users need a special kind of chart that isn't one of the out-of-the-box options when they build a dashboard. In this tutorial, we'll take a look at some ways you might give your users what they want.

To use the API, you have your app send a request to a URL that's based on the instance where your Salesforce org is running. For example, if your org is hosted on `na1.salesforce.com`, you could get a list of all the reports you have by sending a request to `https://na1.salesforce.com/analytics/reports`.

Here are the basic operations you can undertake with the Salesforce Reports and Dashboards API. We'll be using some of these in the next few steps.

Action	URL	Method	Request Body
List all recently used, supported reports.	<code>/analytics/reports</code>	GET	N/A
Retrieve report, report type, and related metadata for the specified report.	<code>/analytics/reports/<reportId>/describe</code>	GET	N/A
Run the specified report.	<code>/analytics/reports/<reportId></code>	GET	N/A
Run the specified report with dynamic filters.	<code>/analytics/reports/<reportId></code>	POST	Report Metadata
Run the specified report asynchronously.	<code>/analytics/reports/<reportId>/instances</code>	POST	N/A
Run the specified report asynchronously with filters.	<code>/analytics/reports/<reportId>/instances</code>	POST	Report Metadata
List the 200 most recent run instances of the specified report.	<code>/analytics/reports/<reportId>/instances</code>	GET	N/A
Fetch the specified run instance of the specified report.	<code>/analytics/reports/<reportId>/instances/<instanceId></code>	GET	N/A
Get a list of recently used dashboards.	<code>/analytics/dashboards</code>	GET	N/A
Retrieve metadata, data, and status for the specified dashboard.	<code>/analytics/dashboards/<dashboardID></code>	GET	N/A

Trigger a dashboard refresh.	/analytics/dashboards/<dashboardID>	PUT	N/A
Get the status for the specified dashboard.	/analytics/dashboards/<dashboardID>/status	GET	N/A

 **Note:** This is just a brief overview to show you some of the cool kinds of things you can do with reports and dashboards using the Salesforce Reports and Dashboards REST API. For full instructions and a detailed reference, check out the [Salesforce Reports and Dashboards API Developer Guide](#).

Step 1: Run a Report Synchronously


If speed is what you need, synchronous execution is the way to go. Your report runs afresh every time the user looks at it, and feeds it right back to your app. If your users need to track hour-by-hour changes, you may want to run your report synchronously.

Let's get acquainted with the Reports and Dashboards REST API by running a saved report with the ID 00OD0000001ZbP7MAK. We'll run it synchronously this time, and without any filters.

- Kick off the report by sending a GET command with the ID of the report you want to run.

The command will look like this:

```
curl -s -H 'Authorization: OAuth token ...'
https://na1.salesforce.com/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK
```

 **Note:** We're using NA1 as the instance for this example. Substitute the instance where your Salesforce organization is hosted.

You've just run your first report via the API! Don't worry about reading the results yet. You'll get to that in the next few pages.

Step 2: Run a Report Asynchronously

Running a report asynchronously means sending the request, then getting the results back at some later time. There are a few advantages to running reports asynchronously through the API.

- When you run asynchronously, the results are kept around in a cache that you can use any time during the next 24 hours. And the API commands for reusing cached results don't count against the 1200-requests-per-hour limit. (General API request limits still count, though.)
- Asynchronous reports have a longer time-out interval. So if you know your report is looking at a very large data set, and you don't want to risk timing out, you might want to run asynchronously.
- You can run up to 1,200 asynchronous reports per hour, which is over twice the limit for synchronous reports. So if you expect a lot of users to be looking at your app, asynchronous runs might be for you.

- Kick off your asynchronous report by sending a POST command to `https://<instance>/analytics/reports/<reportId>/instances`.

The command will look like this:

```
curl -s -H 'Authorization: OAuth token ...'
https://na1.salesforce.com/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/instances
-X POST -d ''
```

- To get the results of your asynchronous run, poll the report run instance with GET.

A specific asynchronous run of a report is called an instance. Each instance has an ID. To get the data set that an instance contains, you send a request to the system, identifying the instance you want by its ID. This is called polling the instance. If the report has finished running, the response to your poll request is the data set you asked for. (If it's not finished, you get an "in progress" response.)

```
curl -s -H 'Authorization: OAuth token ...'
https://na1.salesforce.com/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK/instances/instance_id
```

Now we've run a report synchronously and asynchronously. Next, we'll make our data more useful by narrowing down our results.

Step 3: Filter Report Data

A report is most useful when you use filters to narrow down the data it returns.

You learned how to set filters on the fly, using the standard web interface, in [Step 3: Create the Gauge Dashboard Component](#) on page 5. You can filter a report via the API as well. The API has commands to add filters, edit them, or remove them.

For example, say you've just run a saved report that is filtered to show only items that you have more than a dozen of. Now you want to filter for smaller quantities, without changing the saved report. To do this, send back the report metadata object with edited filters.

1. Here's some typical metadata that your report run might have returned:

```
{ "reportMetadata": { "name": "PipelineReport", "id": "00OD0000001ZbP7MAK", "developerName": "PipelineReport",
  "reportType": { "type": "MerchandiseList", "label": "Merchandise" }, "reportFormat": "MATRIX",
  "reportBooleanFilter": null, "reportFilters": [ { "column": "AMOUNT", "operator": "greaterThan",
    "value": "12" } ], "detailColumns": [ "ACCOUNT.NAME", "CREATED_DATE", "AMOUNT" ],
  "currency": null, "aggregates": [ "RowCount" ], "groupingsDown": [ { "name": "CONTACT2.COUNTRY_CODE",
    "sortOrder": "Asc", "dateGranularity": "None" } ], "groupingsAcross": [ { "name":
    "OWNER", "sortOrder": "Asc", "dateGranularity": "None" } ] } }
```

2. Change the filter and run the report. It will look something like this, with the edited filter shown in bold type. (This example is synchronous, but an asynchronous run works the same way.)

```
curl -s -H 'Authorization: OAuth token ...'
https://na1.salesforce.com/services/data/v29.0/analytics/reports/00OD0000001ZbP7MAK -X
POST -d '{ "reportMetadata": { "name": "PipelineReport", "id": "00OD0000001ZbP7MAK",
  "developerName": "PipelineReport", "reportType": { "type": "CaseList", "label": "Cases" },
  "reportFormat": "MATRIX", "reportBooleanFilter": null, "reportFilters": [ { "column": "AMOUNT",
    "operator": "lessThan", "value": "12" } ],
  "detailColumns": [ "ACCOUNT.NAME", "CREATED_DATE", "AMOUNT" ],
  "currency": null, "aggregates": [ "RowCount" ], "groupingsDown": [ { "name": "CONTACT2.COUNTRY_CODE",
    "sortOrder": "Asc", "dateGranularity": "None" } ], "groupingsAcross": [ { "name": "OWNER",
    "sortOrder": "Asc", "dateGranularity": "None" } ] } }
```

You've just run a filtered report and retrieved the data. You're ready to do some cool tricks with it! For some ideas, along with full instructions and detailed reference information, check out the [Salesforce Reports and Dashboards REST API Developer Guide](#).

Step 4: Find, Show, and Refresh Dashboards

Many users interact with reports mainly through dashboards. You can use the Reports and Dashboards REST API to access and refresh dashboards just as easily as you can with reports.

For example, suppose your users are tired of paging through screens in search of the dashboards they need. You can use the Reports and Dashboards REST API to let them choose from among the dashboards they've looked at recently.

1. To help your users find their dashboards easily, use a GET request on the Dashboard List resource to retrieve a list of recently used dashboards.

```
/services/data/v31.0/analytics/dashboards
```

For each dashboard, the Dashboard List resource sends you back something like this. The URL handle stores the status or results for the dashboard. The list is sorted by the date when the dashboard was last refreshed.

```
[ {
  "id" : "01ZD00000007QeuMAE",
  "name" : "Adoption Dashboard",
  "statusUrl" : "/services/data/v31.0/analytics/dashboards/01ZD00000007QeuMAE/status",
  "url" : "/services/data/v31.0/analytics/dashboards/01ZD00000007QeuMAE"
}]
```

2. You may want to show users their dashboard data in different ways, depending on the platform or device where they're using your app. You can pull the data from the dashboard with a GET request to the Dashboard Results resource.

```
/services/data/v31.0/analytics/dashboards/01ZD00000007S89MAE
```

What you get back is the actual data in the dashboard, plus its metadata (the dashboard ID, name, component metadata, and any filters) and its refresh status. The result will look like this:

```
{
  {
    "componentData" : [ {
      "componentId" : "01aD0000000a36LIAQ",
      "reportResult" : {
        // Report result data omitted for brevity.
      },
      "status" : {
        "dataStatus" : "DATA",
        "errorCode" : null,
        "errorMessage" : null,
        "errorSeverity" : null,
        "refreshDate" : "2014-04-10T20:37:43.000+0000",
        "refreshStatus" : "IDLE"
      }
    } ],
    "dashboardMetadata" : {
      "attributes" : {
        "dashboardId" : "01ZD00000007S89MAE",
        "dashboardName" : "Simple Dashboard",
        "statusUrl" : "/services/data/v31.0/analytics/dashboards/01ZD00000007S89MAE/status",

        "type" : "Dashboard"
      },
      "canChangeRunningUser" : false,
      "components" : [ {
        "componentData" : 0,
        "footer" : null,
        "header" : null,
        "id" : "01aD0000000a36LIAQ",
        "properties" : {
          "aggregateName" : "s!AMOUNT",
          "maxRows" : null,

```

```

        "sort" : {
          "column" : "TYPE",
          "sortOrder" : "asc"
        },
        "visualizationProperties" : { },
        "visualizationType" : "Bar"
      },
      "reportId" : "00OD0000001g2nWMAQ",
      "title" : null,
      "type" : "Report"
    } ],
    "description" : null,
    "developerName" : "Simple_Dashboard",
    "filters" : [ {
      "name" : "Amount",
      "options" : [ {
        "alias" : null,
        "endValue" : null,
        "id" : "0ICD00000004CBioAM",
        "operation" : "greaterThan",
        "startValue" : null,
        "value" : "USD 2000000"
      } ],
      "selectedOption" : null
    } ],
    "id" : "01ZD000000007S89MAE",
    "layout" : {
      "columns" : [ {
        "components" : [ 0 ]
      } ]
    },
    "name" : "Simple Dashboard",
    "runningUser" : {
      "displayName" : "Allison Wheeler",
      "id" : "005D000000016V2qIAE"
    }
  }
}

```

3. If you're concerned that users might not be seeing the latest data, you can refresh a dashboard remotely by sending a PUT Dashboard Results request, specifying the dashboard you want to refresh by its ID.

```
/services/data/v31.0/analytics/dashboards/01ZD000000007S89MAE
```

The response contains the status URL for the refreshed dashboard:

```

{
  "statusUrl" : "/services/data/v31.0/analytics/dashboards/01ZD000000007S89MAE/status"
}

```

Summary

You've learned some nifty tricks in the workbook, from filtering report data to combining multiple reports in one. Now you've also got the basics for doing it all remotely, from any app on any platform. With the Salesforce Reports and Dashboards REST API, you can help users stay on top of their critical data wherever they are.

FINAL SUMMARY

We've come a long way since the beginning of the book. We've covered how to build detailed reports, how to visualize your data with report charts, and how to leverage the data in a report with dashboard components. We've seen the importance of sharing your analysis with the right people, and how to control access using folders.

We've also covered a lot of powerful reporting features, like joined reports, conditional highlighting, cross filters, and two-axis combination charts, and a whole lot more. We even jumped into some advanced skills by creating formulas using summary functions.

We discussed the importance of sharing and collaboration around data. Ultimately, the goal of analysis is action. By analyzing trends, comparing numbers, and gauging performance, you identify problem areas, set expectations, and recognize value. Chatter brings data and people together.

In the final tutorial, we saw how you can design and deliver reports and dashboards without any need for the standard Salesforce web interface. We sketched out ways to create, filter and read report data using the Reports and Dashboards REST API. You can find real-life apps using the API on the [App Exchange](#). Try it yourself!

Your finished dashboard will look something like this, using your own data. Each dashboard can have up to 20 components. What will you create with the other fourteen?





<http://developer.force.com/workbooks>