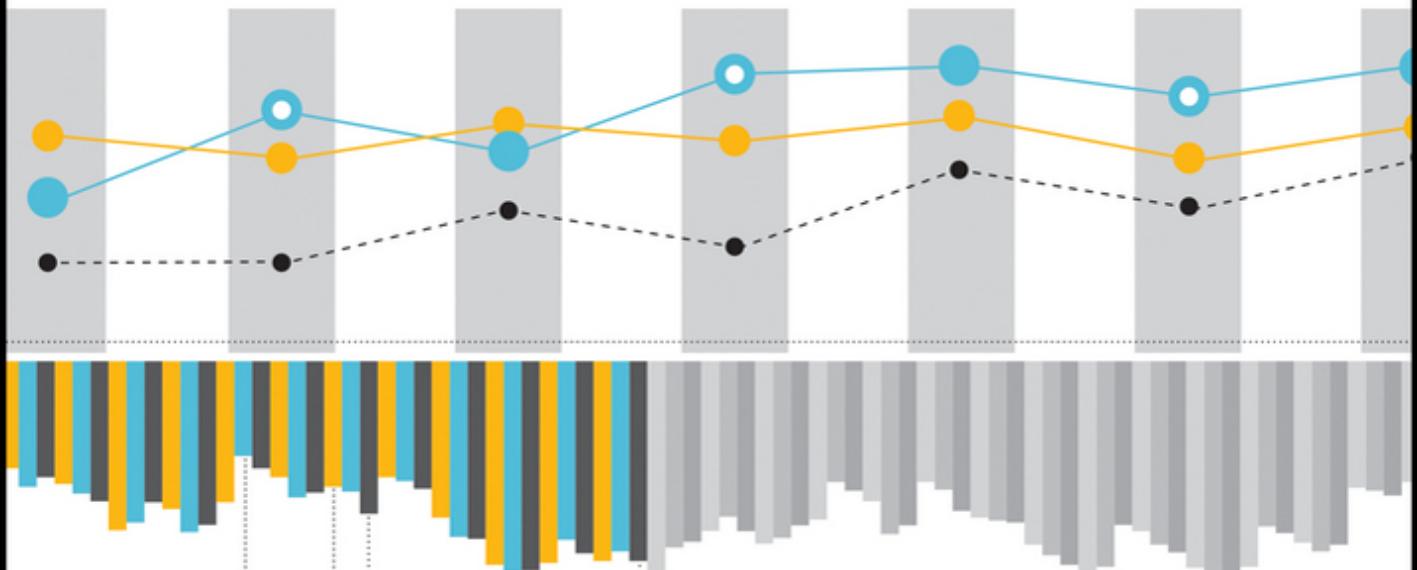




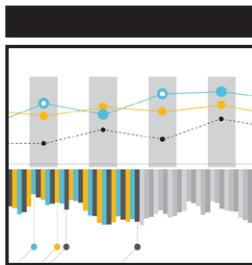
Christopher Price  
Adam Jorgensen  
Devin Knight

# Building Performance Dashboards and Balanced Scorecards

with SQL Server® Reporting Services



WILEY



# Building Performance Dashboards and Balanced Scorecards with SQL Server® Reporting Services

---

Christopher Price  
Adam Jorgensen  
Devin Knight

WILEY

# Contents at a Glance

<b>Introduction</b>	<b>xix</b>
<b>Part I      Foundations of Dashboards and Balanced Scorecards</b>	<b>1</b>
<b>Chapter 1    Introduction to Balanced Scorecards</b>	<b>3</b>
<b>Chapter 2    Planning Your Dashboard and Scorecard Initiative</b>	<b>21</b>
<b>Part II     Building and Extending Dashboards and Balanced Scorecards</b>	<b>33</b>
<b>Chapter 3    Building Your First Dashboard and Balanced Scorecard</b>	<b>35</b>
<b>Chapter 4    Enhancing your Dashboard and Balanced Scorecard</b>	<b>51</b>
<b>Chapter 5    Interactive Dashboards and Balanced Scorecards</b>	<b>67</b>
<b>Chapter 6    Visual Dashboards and Balanced Scorecards</b>	<b>89</b>
<b>Chapter 7    Mobile Dashboards and Balanced Scorecards</b>	<b>125</b>
<b>Chapter 8    Extending Dashboards and Balanced Scorecards with Custom Code</b>	<b>139</b>
<b>Chapter 9    Publishing Dashboards and Balanced Scorecards</b>	<b>155</b>
<b>Part III    Dashboards and Balanced Scorecards in Action</b>	<b>173</b>
<b>Chapter 10   Dashboards</b>	<b>175</b>
<b>Chapter 11   Strategy and Process Maps</b>	<b>195</b>
<b>Chapter 12   Balanced Scorecards Case Study</b>	<b>219</b>
<b>Chapter 13   Decision-making Charts</b>	<b>233</b>
<b>Chapter 14   Forecasting Trends and Data Smoothing</b>	<b>253</b>
<b>Index</b>	<b>277</b>



# Contents

<b>Introduction</b>	<b>xix</b>
<b>Part I      Foundations of Dashboards and Balanced Scorecards</b>	<b>1</b>
<b>Chapter 1    Introduction to Balanced Scorecards</b>	<b>3</b>
Origins of Strategic Management Systems	3
Tools of Today's Management	4
Balanced Scorecards	4
Strategy Maps	5
Dashboards	7
Benefits of Today's Scorecards	7
Team Strategy	8
Information Technology Alignment	8
Data Strategy	8
Organizational Alignment	8
Strategy to Action Conversion	9
Choosing the Right Technology for the Task	9
Designing for Audience Usability	10
Executives	10
Sales and Marketing	10
Field and Global Workforce	11
Operations and Production	11
Organizing Your Balanced Scorecard	11
Defining Your Strategic Function	12
Why is the Strategy Map so Important?	17
How Do You Know It's Working?	19
Moving Forward with your Scorecard Initiative	19
Summary	19

<b>Chapter 2</b>	<b>Planning Your Dashboard and Scorecard Initiative</b>	<b>21</b>
Defining Scope and Gathering Requirements		22
Executive Involvement		22
Defining Measures		25
Selecting the Right Performance Metrics		28
Planning Development and Implementation		29
Managing Expectations and Scope		30
Summary		31
<b>Part II</b>	<b>Building and Extending Dashboards and Balanced Scorecards</b>	<b>33</b>
<b>Chapter 3</b>	<b>Building Your First Dashboard and Balanced Scorecard</b>	<b>35</b>
Anatomy of a Reporting Services Project		36
Introduction to Data Sources, Datasets, and Controls		37
Design and Preview		41
Building Your First Dashboard		46
Scoping and Requirements for Your Dashboard		46
Considerations Before You Get Started		47
Identifying Metrics and Gathering Data		47
Working with Metrics and KPIs		48
Designing the Dashboard		49
Summary		50
<b>Chapter 4</b>	<b>Enhancing Your Dashboard and Balanced Scorecard</b>	<b>51</b>
Organizing and Achieving Visual Consistency		51
Placement and Layouts		52
Dashboard Headers and Footers		53
Reports, Subreports, and Paging		54
Formatting Dashboard Elements		55
Fonts and Text Boxes		55
Borders, Lines, Color, and Images		56
Text, Numbers, and Dates		58
Conditional Formatting		60
Expressions		60
Conditional Formatting Step-by-step		61
Custom and Data-driven Labels and Titles		62
Creating Custom Labels		63
Working with Dates and Times		64
Summary		66
<b>Chapter 5</b>	<b>Interactive Dashboards and Balanced Scorecards</b>	<b>67</b>
Common Interactive Features		67
Drillthrough		68
Expand / Collapse Groups		69
Interactive Sorting		71
Bookmarks		72

---

Parameterization	74
Introduction to Reporting Services Parameters	75
Data-driven Parameters	79
Working with Cascading Parameters	82
Step-by-step Drillthrough Report	83
Summary	88
<b>Chapter 6 Visual Dashboards and Balanced Scorecards</b>	<b>89</b>
General Guidelines for Data Visualizations	89
Know your Audience	90
Select the Right Medium	90
Kill Cute	90
Use Colors Carefully	91
Working with Charts and Graphs	92
Getting Started with Charts	93
Extending the Reporting Services Chart	102
Working with Sparklines and Data Bars	107
Sparklines	107
Data Bars	109
Working with Gauges and Bullet Graphs	110
Gauges	110
Bullet Graphs	112
Working with Indicators	115
Working with Maps	118
Summary	123
<b>Chapter 7 Mobile Dashboards and Balanced Scorecards</b>	<b>125</b>
Planning for the Right Audience	125
Browser or Native?	126
Choosing a Mobile Platform	127
The Pros and Cons of Reporting Services	127
The Pros and Cons of PerformancePoint	128
General Considerations for Mobile Dashboards	134
Performance	135
Security	136
Design Considerations	137
Summary	137
<b>Chapter 8 Extending Dashboards and Balanced Scorecards with Custom Code</b>	<b>139</b>
Choosing to Embed Custom Code	140
Direct Inputting Code	141
Writing a Function	142
Creating Assemblies for Reusability	146
Building a .NET Assembly	147
Deploying an Assembly	150
Using an Assembly in a Report	151
Summary	153

<b>Chapter 9</b>	<b>Publishing Dashboards and Balanced Scorecards</b>	<b>155</b>
	Choosing a Deployment Location	155
	Configuring and Publishing to a Native Report Server	156
	Setting Up and Configuring Reporting Services	156
	Planning for Growth and Scalability	158
	Deploying to SQL Server Reporting Services	161
	Managing Security	162
	Configuring and Publishing to SharePoint	165
	Configuring Reporting Services for SharePoint	166
	Deploying to SharePoint	170
	Data Alerts	170
	Summary	172
<b>Part III</b>	<b>Dashboards and Balanced Scorecards in Action</b>	<b>173</b>
<b>Chapter 10</b>	<b>Dashboards</b>	<b>175</b>
	Case Study: Adventure Works	175
	What to Measure and When	176
	Developing the Dashboard	176
	Summary	194
<b>Chapter 11</b>	<b>Strategy and Process Maps</b>	<b>195</b>
	Strategy Maps	195
	Perspectives	197
	Strategic Themes	198
	Objectives and Casual Links	200
	Designing a Strategy Map	201
	Process Maps	202
	Process Flow Map	202
	SIPOC Map	206
	Value Stream Map	207
	Case Study: Contoso Plastics	210
	Background	211
	The Situation	211
	Process Improvement	212
	Design Sessions	212
	Anecdotal Results	213
	Implementing Strategy and Process Maps	214
	Working in Microsoft Visio	214
	Alternative Implementation Options	217
	Summary	218
<b>Chapter 12</b>	<b>Balanced Scorecards Case Study</b>	<b>219</b>
	Designing a Balanced Scorecard for a Marketing Firm	220
	Defining Objectives	220
	Selecting Metrics	222
	Choosing Appropriate Targets	223

---

Bringing the Balanced Scorecard Together	225
Wrangling the Data	226
Building the Balanced Scorecard	226
Summary	231
<b>Chapter 13 Decision-making Charts</b>	<b>233</b>
Box-and-whisker Charts	233
Box-and-whisker by Example	234
Building a Box-and-whisker Chart	235
Pareto Charts	240
80-20 Rule	240
Pareto Chart	240
Building a Pareto Chart	241
Variance Charts	244
Gantt Charts	248
Summary	251
<b>Chapter 14 Forecasting Trends and Data Smoothing</b>	<b>253</b>
Smoothing Data	253
Simple Moving Average	254
Weighted Moving Average	256
Exponential Weighted Moving Average	257
Forecasting with Moving Averages	259
Forecasting	259
Forecasting with Microsoft Excel	259
Forecasting with SQL Server Analysis Services	262
Case Study: Health Care Forecasting	273
Gathering and Preparing the Data	273
Building the Time Series Model	274
Making the Forecast	276
Summary	276
<b>Index</b>	<b>277</b>



# Introduction

Balanced scorecards and performance dashboards are high value, high visibility tools for driving and monitoring organizational performance in today's hypercompetitive business environment. These business critical reports can be reliably delivered using Reporting Services on Microsoft's SQL Server platform.

This book focuses on understanding and successfully delivering performance management reports using SQL Server Reporting Services by introducing the benefits, concepts, and techniques behind designing, building, and delivering dashboards, balanced scorecards, and other performance management reports.

## How This Book Is Organized

---

This book is organized into three distinct sections:

### **Part I: Foundations of Dashboards and Balanced Scorecards**

The first section of the book takes a holistic look at performance management, common metrics, and the steps necessary to successfully implement balanced scorecards, dashboards, and other similar reports. Chapter 1 is foundational in covering the business drivers and metrics involved, while Chapter 2 discusses gathering business requirements, scope, project management, and implementation.

***Chapter 1: Introduction to Balanced Scorecards***

This chapter introduces the basic premise behind both dashboards and balanced scorecards from a manager's or executive's perspective and answers the question of why these analytical tools are important, diving into the fundamentals of balanced scorecards, explaining the scorecard in the business process context, including defining a strategic foundation while developing an overall vision.

***Chapter 2: Planning Your Scorecard Initiative***

This chapter outlines a proven project approach to implementing dashboards and balanced scorecards in the enterprise, including getting executive buy-ins and defining managing scope, requirements, and expectations.

**Part II: Building and Extending Dashboards and Balanced Scorecards**

The second section of the book targets report designers and developers, as you are introduced to the skills and techniques needed to design, develop, and deliver both a dashboard and balanced scorecard using Reporting Services. This section assumes only limited Reporting Services knowledge, walking you through building a progressively more advanced solution.

***Chapter 3: Building Your First Dashboard and Balanced Scorecard***

This chapter guides you through the building of both a simple dashboard and a balanced scorecard. The examples presented introduce important Reporting Services features and objects including: reports, data sources, datasets, tables, and matrices.

***Chapter 4: Enhancing Your Dashboard and Balanced Scorecard***

This chapter introduces you to techniques for enhancing the dashboard and scorecard. Topics include advanced formatting controls and labels, paging, and conditional formatting.

***Chapter 5: Interactive Dashboards and Balanced Scorecards***

Chapter 5 introduces the skills necessary to make Reporting Services dashboards and scorecards interactive, including sorting, filtering, parameterization, and report drillthrough.

### ***Chapter 6: Visual Dashboards and Balanced Scorecards***

The focus of Chapter 6 is on the visual aspect of dashboards and scorecards in Reporting Services. Topics include charts, gauges, sparklines, graphs, and map integration. You are also introduced to more advanced topics, such as customizing visual elements, setting thresholds, and alerting.

### ***Chapter 7: Mobile Dashboards and Balanced Scorecards***

Chapter 7 discusses topics and considerations that are important when designing and implementing dashboards for mobile devices.

### ***Chapter 8: Extending Dashboards and Balanced Scorecards with Custom Code***

Custom code within Reporting Services is introduced as a special topic in Chapter 8. You will walk through the implementation of a few examples, such as alternating row color and advanced formatting in both embedded code and in external assemblies.

### ***Chapter 9: Publishing Dashboards and Balanced Scorecards***

Chapter 9 closes the loop and discusses publishing and deploying your dashboard and/or scorecard including administration, security, scheduling, and report subscriptions.

## **Part III: Dashboards and Balanced Scorecards in Action**

The third section of the book consists of case studies to highlight common performance management report implementations across different industry segments. This section targets executives, managers, and developers, as each chapter discusses the rationale, techniques, and implementation of each performance management report.

### ***Chapter 10: Dashboards***

This chapter uses a fictional retail sales company to illustrate the techniques and skills learned up to this point to design and implement a typical dashboard in Reporting Services.

### ***Chapter 11: Process and Strategy Maps***

Chapter 11 introduces the techniques and fundamentals behind process and strategy maps. A manufacturing case study is presented to walk you through the design and implementation process for both.

### ***Chapter 12: Balanced Scorecards***

This chapter uses a retail and marketing scenario as a functional example for designing and developing a balance scorecard using Reporting Services.

### ***Chapter 13: Decision-making Charts***

Chapter 13 introduces decision-making tools, including box-and-whisker, Pareto, and Gantt charts.

### ***Chapter 14: Forecasting Trends and Data Smoothing***

The final chapter discusses forecasting and data smoothing techniques to highlight trends. A financial services case study is used to demonstrate a common implementation of the techniques discussed.

## **Who Should Read This Book**

---

This book was written for a very diverse audience and as such it is possible to skip around and get the information you need or want.

Managers, decision makers, and executives who are trying to get up to speed or are ready to take on a dashboard project should start with Part I. They will also find Part III useful in bridging theory with practice.

Developers and practitioners who are already familiar with analytical report fundamentals can start with Part II and dive directly into the skills needed to design, build, and implement dashboards and balanced scorecards.

## **Tools You Will Need**

---

The primary focus of this book is Microsoft SQL Server Reporting Services. To follow along and apply the skills and techniques demonstrated, you will need to have either Microsoft SQL Server 2008 R2 or Microsoft SQL Server 2012 Database Engine and Reporting Services installed.

Additionally, you will need to download and install the Adventure Works sample databases from: <http://msftdbprodsamples.codeplex.com/>.

Periodically, other tools such as Microsoft SQL Server Analysis Services, SharePoint, PerformancePoint, and PowerView are mentioned or discussed. Having this software available is useful but not required.

## **What's on the Website?**

---

The companion website for this book can be found at [www.wiley.com/go/performancedashboards](http://www.wiley.com/go/performancedashboards) and it contains all completed sample code and Visual Studio projects demonstrated for this book. To use the samples provided, open each solution and then update the data source for your local machine instance.

## **Summary**

---

This book will help you build a solid foundation on which you will be able to deliver meaningful and powerful analytical reports. The skillsets and techniques are intended to help make your dashboards and performance scorecards not only actionable and easy to use but also easier to build and manage.

# Foundations of Dashboards and Balanced Scorecards

---

## In This Part

---

This section of the book introduces the components and best practices for getting your scorecard and dashboard initiatives underway. Chapter 1 discusses the concepts behind the scorecard and how to get started planning the development of your first one. Chapter 2 helps you understand all the required steps for planning and executing your balanced scorecard initiative within your organization.

**Chapter 1:** Introduction to Balanced Scorecards

**Chapter 2:** Planning Your Dashboard and Scorecard Initiative



# Introduction to Balanced Scorecards

In today's economy and business climate, tools are needed that enable management teams and executives to adapt to the demands of changing markets and rapidly moving conditions while working across the organization to drive success through strategic alignment. Balanced scorecards and dashboards, along with strategy maps, are organizational tools that help ensure this cross-business alignment and strategy implementation. This chapter covers where these tools originated, how management teams first began leveraging them, and how firms are using them today to be successful in their implementations of strategy.

## Origins of Strategic Management Systems

---

A challenge for many growing firms in the 80s and 90s was turning deep strategic goals into actionable behavior across an organization. To fill this need, two experts from Harvard, Kaplan and Norton, put forth the concepts and practices behind the balanced scorecard in an article in *Harvard Business Journal* in 1992. Their new system of managing priorities and aligning responsibilities with profitability focused on converting strategy into action. A careful balance was needed with these proposed scorecard systems because some firms were succeeding with them and others viewed them as administrative overhead and

were getting bogged down in their implementation. To help streamline this overhead and get all the stakeholders on the same page, Kaplan and Norton developed a template driven system called a strategy map that sums up the critical measurements an organization needs to execute on its strategic goals.

## Tools of Today's Management

---

The organizational model has grown and matured since the 90s, and the balanced scorecard is now in its third generation. Since its original development, its components have undergone some changes to produce more actionable deliverables. For instance, strategy maps have been improved by adding categories and areas for business types, such as online-only businesses. These improvements help the management team to more effectively communicate strategy to organizations. These improvements have built on the already significant adoption and success of this scorecard framework by many firms. Because it focuses on a problem that is consistent across industries, namely turning strategy into actions and measuring the success of those actions, the benefits are numerous for every participant. Today's model comes in three basic parts and is used by organizations whose industry did not even exist when the original technique was established.

### Balanced Scorecards

A balanced scorecard is a tried and true method for turning organizational strategy into action. It is a specific format and organizational technique used to align goals to owners and strategy. Organizations use balanced scorecards to add clarity to their goals. The timetable and execution may vary between organizations, but the scorecard is flexible to accept these new criteria and requirements.

The balanced scorecard has a couple components. The first is a *destination statement*. The destination statement is a one- to two-page description of your organization at a point in the future, typically three to five years out, describing a desired end state.

The second component is the perspective map that shows how those goals align with different perspectives of the organization. This is the thing most people think of as the “balanced scorecard” since it is the most viewed artifact of the process. Traditionally, it's separated into perspectives for each consumer group: financial stakeholders, employees, customers, partners, and suppliers. It includes any strategic view that you think is important as you look at your organization on a three- to five-year landscape.

See an example of a high level scorecard in Figure 1-1.

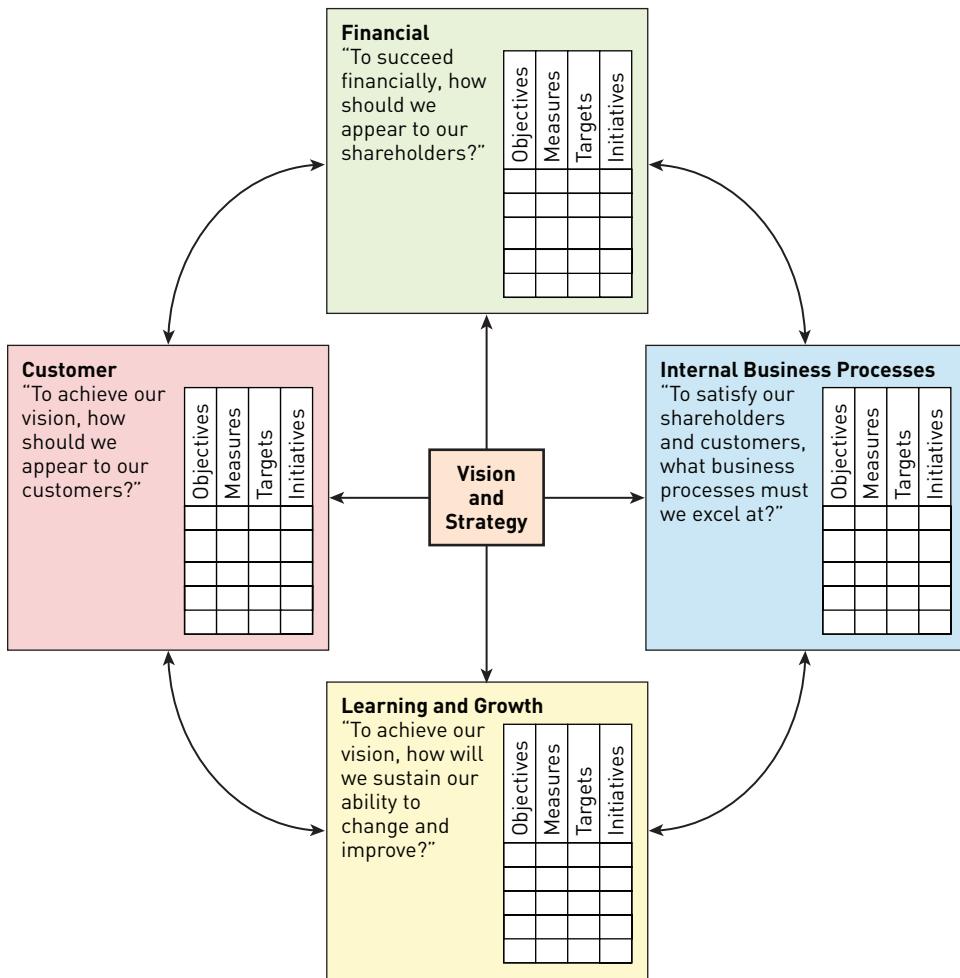


Figure 1-1

## Strategy Maps

A *strategic linkage model*, commonly known as a strategy map, is a component within a balanced scorecard that displays the strategy of the entire organization. It shows how each major area applies a strategy and offers clear metrics for measuring performance through the fiscal or operational year to ensure attainment of the goals established in the balanced scorecard. The strategy map is important because it defines every person's role in driving success for the organization.

The strategic linkage model contains 12 to 24 objectives and is broken into two perspectives: activities and outcomes. Activities could include adding new product lines, increasing safety margins, or other tasks. Outcomes could be winning a safety award pertinent to the industry or being in the top five in a particular product line as measured by a predefined metric.

It also includes definitions for all key objectives. These definitions are critical, because they enable each part of the organization to determine if it is aligned to the objectives and prepared to focus on what needs to be done to meet them. Your strategy map should also include a list and definitions for all the measures you've selected to monitor the strategic objectives. This includes target numbers for the measures as well as sourcing information. For example, if you have a sales measure and there's a particular system, query, or column within a system from which you want to pull information, you need to document that metadata in your strategic linkage model.

An example strategy map is illustrated in Figure 1-2.

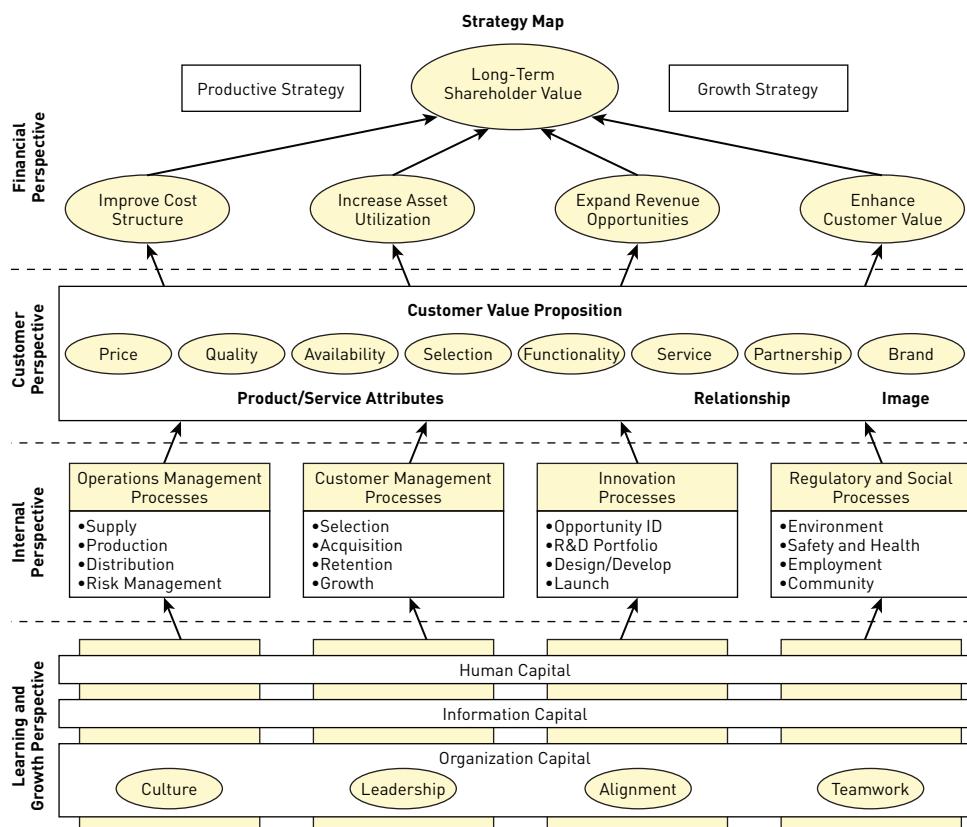


Figure 1-2

## Dashboards

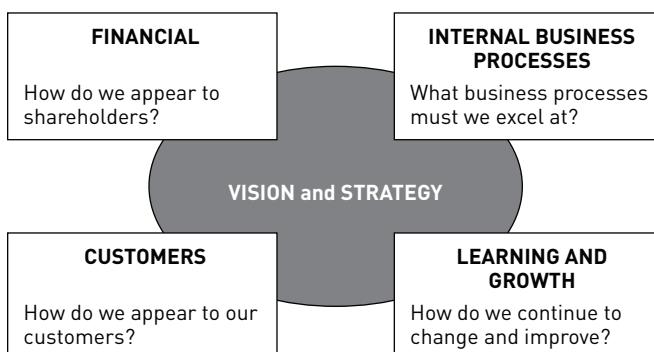
Dashboards enable you to track, roll up, and provide accountability for all of the different areas you are highlighting in your strategy map and balanced scorecard. Dashboards are generally created in Reporting Services, SharePoint, or in a third-party tool. This book focuses on those created in Reporting Services. Figure 1-3 shows an example dashboard.



**Figure 1-3**

## Benefits of Today's Scorecards

These tools continue to benefit organizations in their strategizing, communication, and many other areas. Figure 1-4 illustrates some of the questions organizations are able to answer using this scorecard and strategy map framework.



**Figure 1-4**

Dashboards and strategy maps also help with more specific initiatives such as:

- Team Strategy
- Information Technology Alignment
- Data Strategy
- Organizational Alignment
- Strategy to Action Conversion

Take a look at each of these initiatives in more detail to understand how dashboards and strategy maps work to aid executive leadership.

## Team Strategy

In a fast moving organization, having teams that are working in alignment is critical for success. An effective balanced scorecard takes into account your teams' strategies and how they roll up into the overall success of the organization. Chapter 2, "Planning Your Dashboard and Scorecard Initiative," discusses this in more detail.

## Information Technology Alignment

In a growing organization, many projects and strategic efforts are driven by or focused on information technology. These projects could include core infrastructure changes or upgrades, such as the implementation of new servers or hosting equipment, or data-focused solutions, such as business intelligence or analytics applications.

## Data Strategy

Data strategy is a key part of today's balanced scorecard. With the advent of big data solutions, more advanced scale-out, and cloud options, now more than ever companies rely on data outside of the traditional IT chain. The dashboard tools for these solutions are the same ones businesses have been using. PerformancePoint, Reporting Services, and other third-party tools continue to make strides to improve their performance and functionality to support today's business priorities. New functionality in Excel and other common desktop applications are opening windows into initiatives that in previous years were much more labor intensive or, in some cases, not feasible.

## Organizational Alignment

As organizations grow, misalignment poses challenges. Even the best executive leadership and communication can fall short within a rapidly growing

organization. Tools like the balanced scorecard, strategy map, and related solutions provide the foundation from which decisions tie back and keep executives and managers aligned and speaking the same language throughout the year, as they work toward pre-established goals.

## Strategy to Action Conversion

All of these initiatives are important, but they can't do much good until the leadership team plans out the steps needed to turn strategy into actionable goals. The strategy map is the primary tool that the team uses to communicate goals and actionable steps to the team at large. It creates visibility to the overall plan and each individual's role in the execution of it.

## Choosing the Right Technology for the Task

---

To build dashboards and balanced scorecards and get the team to buy into them, the organization needs access to data and analysis tools to track progress. This is usually where investments in reporting technologies, business intelligence tools, and training come into play. Organizations use these technology components to program their metrics, measures, strategy maps, and scorecards and to make them accessible and dynamic across the organization.

When working with balanced scorecards and dashboards, it's important to understand the different tools available and then choose the right tool for the task. The technology tools that most people use to bring together the data needed for reporting, creating and publishing reports and scorecards, and analyzing metrics' success include Excel, Reporting Services from Microsoft SQL Server, SharePoint, Performance Point, and a number of other third-party solutions. These tools are incredibly powerful and give the user and the developer a lot of flexibility in building scorecards, strategy maps, dashboards, and other types of reporting.

Reporting Services is often a favored tool because it's more easily consumed across the enterprise. It's web-accessible, and many other self-service tools that people use to do additional analysis can import data from Reporting Services. This enables users to look at just their performance or their particular area of the scorecard or dashboard. Typically, people prototype these tools with their individual teams and then create reports based on the goals that are established through the balanced scorecard process. Strategy maps can be created using Visio or Reporting Services. Those, too, can be pushed out to the organization through data-driven subscriptions and other tools.

SQL Server, SharePoint, Performance Point and Excel are other tools from Microsoft that make up their data platform ecosystem. SQL Server is a database

platform that stores the data, SharePoint is the portal and collaboration tool that provides sharing and hosting of reports, Performance Point is a visualization tool sometimes used to create certain components of analytics reports, and Excel is the world's leading spreadsheet and analysis program. Excel is the other tool most often used in creating and prototyping reports that will eventually wind up in other systems. Third party tools are sometimes used for advanced visualizations like info graphics, but this is less common in a scorecard project.

**NOTE** Most users get these tools from their IT department, but you can use Excel and other desktop solutions to prototype fantastic reports that are much more easily shared through an enterprise solution like Reporting Services.

Because of its ease of use, comprehensive features and wide adoption rate, this book focuses on using Reporting Services in the SQL Server environment to create balanced scorecards, dashboards, and strategy maps and how to tie them together. Other functionality from Reporting Services includes SharePoint integration, which enables collaborative commenting and work across the different deliverables, such as strategy maps, individual reports, and drill-down balanced scorecard implementations.

## Designing for Audience Usability

---

The various technology tools available enable you to break down the individual goals for different audiences and create a strategy that contains consumable pieces of information for each of your major audience groups. You can group your audiences into any number of categories, but common major audience groups include executives, sales and marketing, field and global workforce, and operations and production.

### Executives

Executives are primarily focused on high-level instrumentation that tells them whether their key strategies are successful. Generally, a well-targeted dashboard or a series of dashboards will deliver the value that executives are looking for. Departmental executives want their individual metrics, so it's important to enable them to drill through the different reports in an interactive way, so they can see exactly what's going on in their portion of the organization.

### Sales and Marketing

Sales and marketing have goals largely focused on driving new sales, sales growth, and additional market penetration. A variety of reports can be used to

track performance against individual, team, and organizational goals. A balanced scorecard and certainly a dashboard enable this drill-down visibility to the various levels of performance. Teams and individuals can then cross-reference to a strategy map to see exactly where their performance is tracking against their peers and how much they are contributing to the overall success of the organization.

## Field and Global Workforce

The field and global workforce needs to be able to see at a glance where their performance stands against that of their peers and the organization, because often their performance is compensation-driven. This is true especially for global sales teams, field account teams, or global technical teams who need to make sure they are reaching certain goals. A personal, customizable dashboard gives them a lot of flexibility. Reporting Services even enables you to drive content in a particular dashboard based on a user's login or security credentials. These features become handy as you develop personalized scorecard and dashboard interactions for individual employees, roles, and functions within a global workforce.

## Operations and Production

Operations is a major group when it comes to metrics, consumable information, and details like the ones discussed in this chapter. Operations is focused on not just dashboards, scorecards, and strategy maps, but also on individual reporting. This group uses dashboards to view information at an aggregate level, but will likely want very detailed reports from multiple levels that roll up to get to that high level. The group wants to drill down into logistics, financials, and other details that drive success.

Production or manufacturing groups, like operations, want details and data to be very clean. This data will come from both people and machines, and there can be a lot of it. This need for detailed, clean data is becoming more and more important today as technology enables organizations to capture, analyze, and track vast amounts of information to make decisions about everything from production quality, machine utilization, and employee and process efficiency. All of this information can be surfaced through one dashboard or scorecard with drill-down capabilities to other, third-party reporting solutions.

## Organizing Your Balanced Scorecard

---

Now that you understand the benefits of these organizational tools and the importance of choosing your technology and knowing your audience, you are ready to begin laying the groundwork for your first scorecard and dashboard. Overall, the focus of your balanced scorecard initiative should be to make strategy

a continual process. This section will get you started, walking you through the steps necessary to define your strategic foundation, develop a vision, and begin planning and putting process steps in place for your scorecard.

## Defining Your Strategic Function

To develop your vision and strategy, the Balanced Scorecard Institute, the organization that manages balanced scorecard frameworks, recommends a nine-step approach for building a scorecard similar to the one described in the following sections and shown in Figure 1-5.

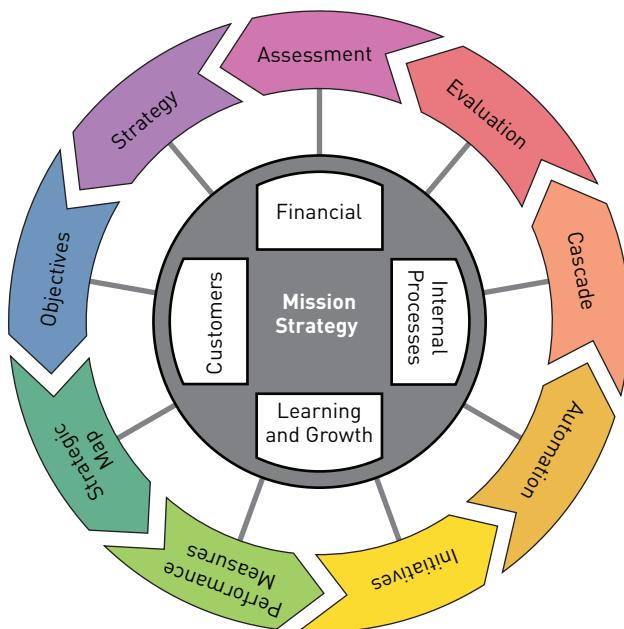


Figure 1-5

**NOTE** The next chapter, "Planning Your Dashboard and Scorecard Initiative," focuses on interviewing executives and gathering scope and requirements. It is important to complete the following vision and strategy exercise with some of those executives before you get into a detailed interview process.

### Step 1: Assessment

Step 1 is where you create your destination statement. Remember to keep this to 1-2 pages and have it describe where you would like your company to be in 3-5

years. Here you should also assess your mission and vision, making sure to call out any challenges, enablers, and values. This assessment should include preparing a change management plan. You should identify key messages and any media outlets or venues that you have to communicate this new updated mission and vision, as well as messengers or people who are allowed to speak on behalf of the strategy to both internal and external customers. External customers, in particular, will evaluate your strategy as you continue to grow it, and will validate your vision in the market. Some will follow you, and some may question you, so it's important to anticipate these challenges. List potential alliances between key customers and suppliers as enablers and values for your organization, and ensure that you're all aligned and going in the correct direction. Figure 1-6 outlines the assessment's recommended steps as they stack up in the order of strategy to action.

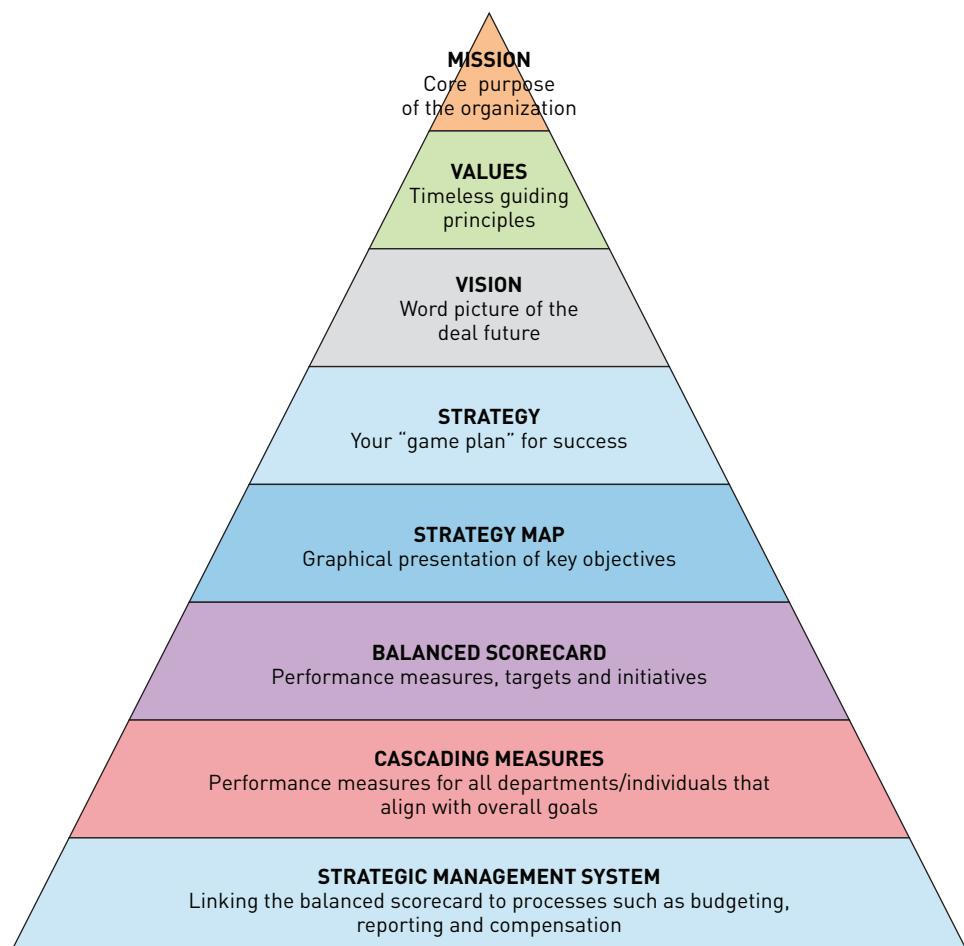


Figure 1-6

### Step 2: Themes and Perspectives

Step 2 involves defining the themes and perspectives within your strategies that must be developed to understand and focus on the organization's value proposition. Ask yourself the following questions:

- What results are you looking for out of this exercise?
- What results will be required over the next three to five years for you to be successful?
- Into what themes can you categorize those results—improved financial performance, increased operational efficiency, successful new product launches?
- What perspectives are most important—financial, external customers (suppliers, alliances, etc.), or internal customers (employee development, improving operational performance, etc.)?

All of these questions are critical to understanding your organization's value proposition. Common themes and perspectives are shown in Figure 1-7.

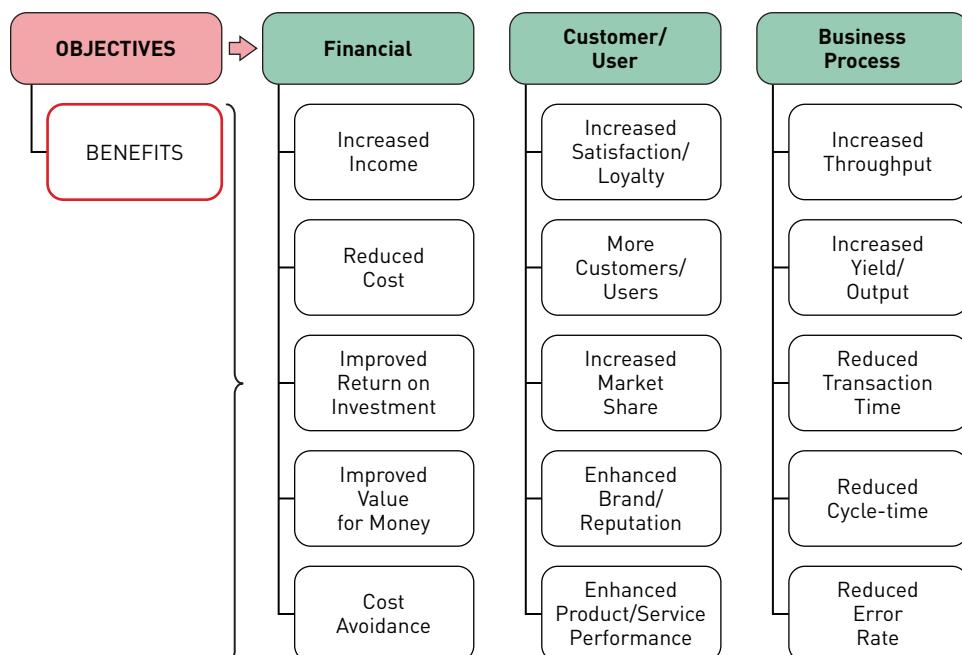


Figure 1-7

### *Step 3: Organizational Objectives*

Step 3 is when you break down the items from Steps 1 and 2 into real objectives. These are the basic building blocks of your strategy, and they define the strategic intent of your organization. The objectives are then initiated and categorized into the themes you established in Step 2. You'll need to further categorize those themes and objectives by perspective. These perspectives are then linked in strategy maps for each theme and merged to produce one set of strategic objectives for the entire organization. This process of breaking the strategies down and then pulling them back together through the different levels of themes and perspectives is critical for successfully navigating the next steps.

### *Step 4: Cause-and-effect Links*

In Step 4, cause-and-effect linkages between the enterprise-wide strategic objectives are formalized into an enterprise-wide strategy map. The previously constructed thematic strategy maps are then combined into the overall strategy map that shows how the organization creates value for customers and stakeholders. Understanding these cause-and-effect linkages is how initiatives are communicated to the individual employees to help them see their role in team and organizational performance. Relating cause and effect linkages creates a structure for motivating employees and driving the appropriate behaviors and focus.

### *Step 5: Defining Performance Measures*

Step 5 is defining leading and lagging performance measures for each strategic objective, and you'll also need to define expected targets, thresholds, and baselines. This information is needed to create good benchmarking data. To gather this data, each measure is compared against its expected target and marked as being within a particular threshold; make sure that it's not too far out from the baseline. You can compare the measure against the baseline if you expect it to increase over your roadmap of three to five years or decrease as a result of an operational efficiency detractor.

### *Step 6: Identify Strategic Initiatives*

In Step 6, you develop strategic initiatives to support each strategic objective. Ownership of the performance measures and the strategic initiatives is assigned and documented in data definition tables. The people assigned to these initiatives should be well aligned and positioned to drive the expected targets and baseline

improvements that you established in Step 5. They should be volunteers or people who are passionate about the focus area or have the knowledge, qualifications, and expertise needed to drive the improvement. They must take ownership of these performance measures seriously and be willing to be accountable for the success or failure of their particular initiative.

### *Step 7: Prototype an Implementation*

Implementation begins in Step 7 by utilizing performance measurement software to get the right performance information communicated to the right people at the correct time. SQL Server Reporting Services, including its associated tools and ecosystem, is an excellent software for the task. Adding this automation enables you to fill in structure and discipline throughout the system, giving people a framework within which to make better business decisions. As they align their decisions, personal strategy, and approach, they have the ability, through this software, to immediately see if they're focused on the correct things and aligned with the organization's continual strategy process and performance objectives.

### *Step 8: Define Multi-tier Scorecards*

Step 8 is when the enterprise scorecard data gets cascaded down into individual business and support unit scorecards and then translated into what's called *tier 3 scorecards*, or individual or team performance scorecards. Pushing the information all the way down to the individual performer is key for organizational alignment around strategy. Steps 7 and 8 enable you to deliver performance information directly from the CEO down to a line worker, salesperson, or individual performer in any department.

### *Step 9: Perform Scorecard Evaluation*

Step 9, the final step, involves a complete evaluation of the scorecard. You need to ask questions regarding effectiveness, such as the following:

- **Are we measuring the right things?** What you measure is critical to the success of the scorecard. If you don't have agreement on what to measure and good data to back it up, then the scorecard will not add value.
- **Have any of our environmental factors or external industry factors changed?** With the pace of change in many industries today, you may have factors change while you are implementing the scorecards. This happens more often than you might think. If you're working through an

implementation and it takes 4-6 months, much can change for an organization in that time. Make sure you have a process to re-validate assumptions through your implementation.

- **Are our finances being budgeted and executed on strategically?** Based on the information provided in the scorecard, are you using your cash and financial instruments to the best level of effectiveness? This will likely come from data on the executive scorecard view.
- **Is employee morale where it needs to be?** Managers need to be able to see employee morale through metrics, such as project success, quality, and attendance. These metrics can be surfaced as part of the multi-tier approach, and a higher level indicator could be used for executive level dashboards to keep them in the loop. Just make sure the actionable data is surfaced to the managers.
- **Is the organization invested in this process?** Are you getting feedback on the process and the scorecards? If no one is talking about them, they are probably not using them. Make sure you are checking in to find out how to make the scorecards as valuable as possible.
- **What improvements to that investment can be made?** Based on the feedback, what adjustments or improvements can be made? Sometimes advanced development techniques can even enable the management team to adjust metric calculations themselves each quarter in accordance with policies. You don't want them changed too often of course.
- **Have any of the data systems that are driving your critical metrics changed over time?** Should you be pulling information from a different system or is there now better data somewhere that would simplify the calculation or metric evaluation process? Take advantage of these opportunities as they arise. If any of the data systems have changed, you must re-validate the data.

## Why is the Strategy Map so Important?

As you learned, development of the strategy map begins early in the nine-step scorecard development life cycle. It's integral to cascading strategy through the multiple levels of the organization, and without it the scorecard loses its actionable focus. Figure 1-8 shows an example of a strategy map that could be used in conjunction with a balanced scorecard.

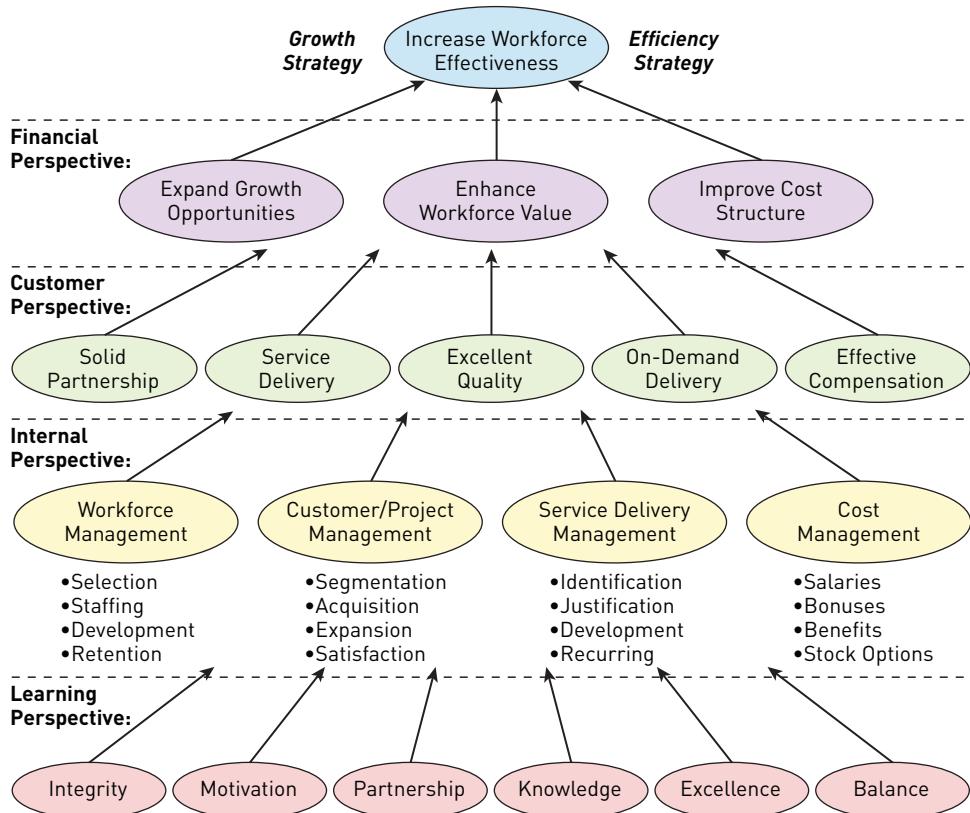


Figure 1-8

The strategy map provides a simple, visual representation of the individual strategic goals and metrics; it indicates to whom they are aligned, whether that's an individual resource, department, business unit, or perhaps even an industry, supply partner, or alliance. It's primarily used to communicate context and enable the entire organization to see the reasons why the executive team chose those metrics in the first place.

Although your goal is to use internal metrics to develop a competitive advantage, you should avoid using industry standard metrics as your key drivers. Industry standard metrics are valuable if there are things that other players within your industry use as part of standardized industry reporting, but make sure you understand your organization's metrics as they relate to those industry standard values. You should only be using industry values if there is a number you need for specific correlation—usually for public consumption and less for strategic analysis. Management must use this information to dig in and find where your organization could be innovative and competitive.

## How Do You Know It's Working?

How do you know if your balanced scorecards and dashboards are working? Hopefully, you're seeing business metrics that are increasing and decreasing in the right places, but the trick is implementing a formal process for constantly reviewing and adjusting the tools. With a formal process in place, teams will individually review and adjust their own metrics and roll them up through their management chain to an organization level.

This roll-up of metrics requires a key executive review process and visibility all the way up to the stakeholders to ensure that they understand why the changes are being made and agree that they are best for the organization.

## Moving Forward with your Scorecard Initiative

There are many steps involved in developing a balanced scorecard, but none should be skipped. Management must invest the time to prepare for and participate in the development sessions. Set the time aside and don't expect to accomplish everything the first time through. Be agile in your implementation and focus on one perspective at a time, driving all the way through that perspective to define the value brought to the organization. Be iterative and walk through the process with different business units. You'll continue to get better and more efficient as you do each one. Perhaps first tackle the ones that you believe are the most prepared for this type of organizational change.

It's also important to have fun with it! You're working on growing your business, making it more agile, or saving money—all are very important to the success of your organization over three to five years and should be positive goals to be working towards. The balanced scorecard, strategy, and vision exercises all intertwine to help you look forward and follow a defined path.

## Summary

---

Balanced scorecards and dashboards can be incredibly beneficial to you and your organization. Some of the key technology aspects include scorecards, dashboards, strategy maps, and how they are used in Reporting Services, SharePoint and PerformancePoint. The development process interrelates with those individual technologies and is therefore important to follow closely. Stakeholder involvement and early collaborative leadership to develop first draft metrics are very important. Data review and validation, continued metric viability assessment, as well as key executive review and visibility exercises remain critical as you go forward.



## Planning Your Dashboard and Scorecard Initiative

This chapter covers the processes and best practices for planning and initiating your scorecard project. These include getting sponsors, defining your scope and requirements, and then planning your iterative, sprint-based execution and development timeline. Scope and requirements for this project mean the measures and metrics from a business perspective and the systems involved. The requirements will include requisite and preferred views of the data, as well as specific methods of calculation the firm or organization desires from the end deliverables.

It's critical to define scope and requirements appropriately, as they will drive your project's success. Take an agile approach, make sure that the scope and requirements are attainable and deliver value quickly. This becomes more and more of a priority as you push for adoption among your user base. It's also important to conduct several iterations of the process, making sure you're adding value with each step. Scope and requirements can also help enforce accountability, which is vital in a scorecard development process, because you have a lot of people who are responsible for the various metrics—lots of different pieces of a puzzle coming together.

## Defining Scope and Gathering Requirements

---

Scorecards can get out of control easily. Too many metrics and measures can lead to confusion and unnecessary work. A lot of effort can be placed in areas that do not add value or reinforce the key metrics. You need to make sure you have the right people involved, the right metrics and measures, the right decisions being made, and the process needs to show value iteratively.

### Executive Involvement

Keeping people invested in the value of your dashboard or balanced scorecard over time is critical, and they're only going to remain invested if they can continue to see value as the process and the project move forward. If you manage the scope appropriately, and potentially aggressively, those people will stay invested, because they will believe that they are going to get value in the end.

#### *Getting Executive Commitment*

Choosing your executives and getting their commitment are the first steps in defining your scope and requirements. As you begin to outline your thoughts regarding which executives you would like to involve in this particular effort, be sure to select good candidates. Your selection will depend on the area of business and level of the company you're assessing, as well as the scope you are defining. Remember from Chapter 1 that dashboard and balanced scorecard perspectives equal stakeholders, and those stakeholders understand what is important to the team within their area of operation; therefore they will likely make good candidates.

As you select executive stakeholders, there are several things to keep in mind.

- Understand what's driving their priorities.
- Do the research and learn what metrics, measures, and operational success factors are critical for them to see value in the project. Common examples, while dependent on organizational strategic objectives include:
  - Revenue or gross profit versus prior period
  - Customer or employee satisfaction scores
  - Inventory turnover rate
  - Customer churn rate
  - Website visits/traffic metrics (unique visitors, page views, time on page)
  - Marketing campaign ROI

- Think through a pre-process question and answer session and conduct that session with a few of these executives to get their feedback on what they understand, what they may still have questions about, and where they may perceive extra value or lack of value for each part of the process.
- Consider conducting an "analytics town hall" session. Bring everybody into the same room, talk through the challenges and the solutions that you're composing, and look for themes in their responses to begin to drive priority.

The right stakeholders will continue to drive success for your organization. They have the right relationships to remove roadblocks when you encounter them. Having those relationships in place enables them to accomplish with a phone call or a casual conversation what it would likely take you hours to overcome.

The right stakeholders can also assist with accountability. They can ensure that the people who have committed to getting things done actually accomplish what they said they would. In other words, they can hold people accountable at their peer level and below. They can also help you understand how to better communicate with people above them in the management chain. This is crucial because being able to communicate effectively both up, down, and across an organization will give you a lot more flexibility, power, and influence as you are moving through this process.

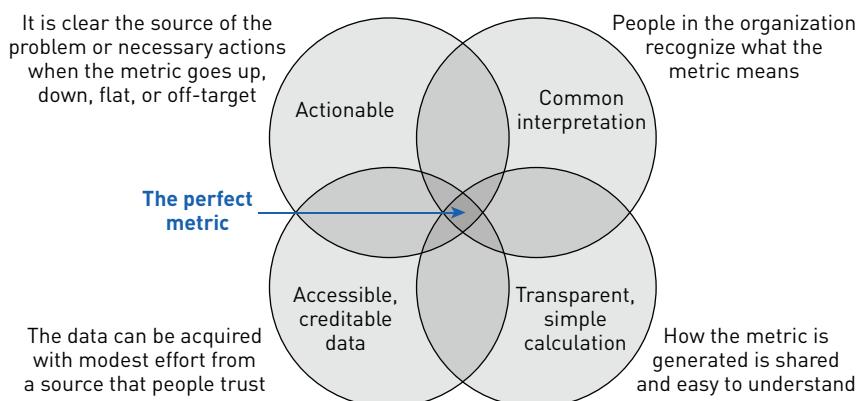
Furthermore, financial executives will have more insight into finance; operational executives will have more insight into operational metrics and management, and so on. Choose the first batch of executives wisely, ensuring they are focused on the right perspectives, information, and set of metrics and measures that you would like as your first priority.

### *Interviewing Executives*

Next, you need to interview the executives you want to commit. Once you've chosen the right executives for your first round of interviews, be sure to create a good set of questions to ask. Talk to your peers and people who have worked on these types of projects before to help you prepare the questions that will enable you to glean the information you need. With questions in hand, you are ready to start your interviews. The following are a few tips to help ensure success:

- Prepare a sample agenda.
- Complete introductions.
- Bring along an ambassador. For example, if you don't have a direct relationship with the executive you're interviewing, bring one of your peers who does. Again, relationships remove roadblocks.

- Clearly explain your goals. A clear explanation of goals helps disarm any resistance to the process and helps people begin to see the value right away.
- Ask for feedback on perceived value of the project. The executive's perceived value directly correlates to how much help he or she is willing to give you.
- Explain the scorecard process. Walk the executive through the perspectives and all of the individual steps that you're going to take to make this process and project a success.
- Educate about metric selection. Metrics should be actionable, have a common interpretation, be composed of accessible and creditable data, and have a transparent and simple calculation. Figure 2-1 might be useful in this regard. Metric selection is discussed in further detail in the section, "Selecting the Right Metrics."



**Figure 2-1**

- Explain the executive's proposed involvement. Ensure that the executive understands exactly what will be expected of him.
- Explain any action items or deliverables that you will need from the executive. Also clarify the schedule on which you'll need those items completed.
- Clarify the outcome of the interview. When you leave the first meeting, make sure that your and the executive's ideas and general topics are aired out appropriately. Also, communicate if there are any other people or other relationships with which you need the executive's help to be successful.
- Discuss internal project logistics or budgetary arrangements, if appropriate. For example, does a new cost center need to be set up or a new project created in the internal project management system? Do there need to be default resources assigned? Is there a budget amount, budget calculator, or budget presentation that needs to be created internally or presented

to any particular committee before you can move forward? Many large organizations have project review committees and other such groups that review and approve these projects as they begin to spin up, and you will need to be accountable for this process. Do your homework before you get too far down the line and make sure that you have the right people lined up to support you.

- Review action items for all parties in the meeting and then adjourn.
- Follow up the meeting with an e-mail, reiterating what was agreed to, and politely thank the executive for his support on the project.

It's crucial to nail down commitments from the executive team early. Their involvement and commitment are the key elements needed to keep the ball rolling, particularly when they are out of town or focused on other things. You should also follow up with each executive periodically throughout the project to make sure that they understand the progress that's being made and the value that they and their teams are delivering.

## Defining Measures

Now it's time to gather your first round of requirements. This involves a few steps, including: collecting the requirements, identifying the measures and filtering them to get a list that's appropriate—since you're being aggressive with managing expectations and scope—and then validating that those measures are accessible through valid and creditable data. Basic building blocks for requirements are shown in Figure 2-2.

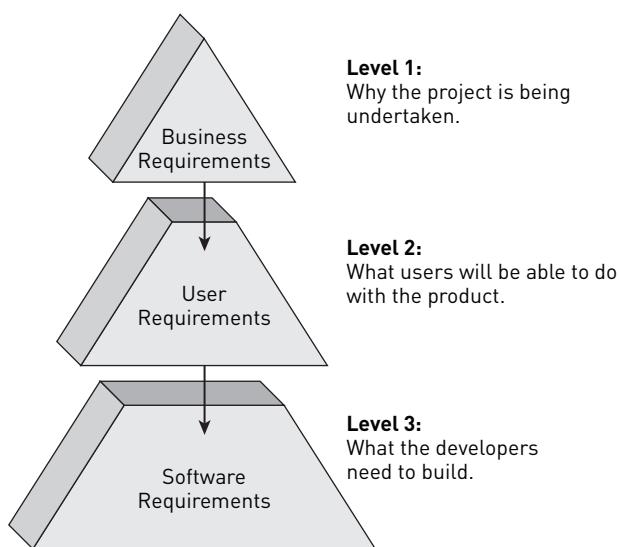


Figure 2-2

## *Collecting*

As you start collecting your first round of measures, it's critical to identify target measures early. What makes a good measure? This is a question you should ask yourself over and over again. A measure is highly contextual and should provide an accurate and timely picture of the organizational objective. Table 2-1 shows how the strategic objective is focused on customer satisfaction. Note that four core measures have been identified, each with a varying degree of timeliness and source.

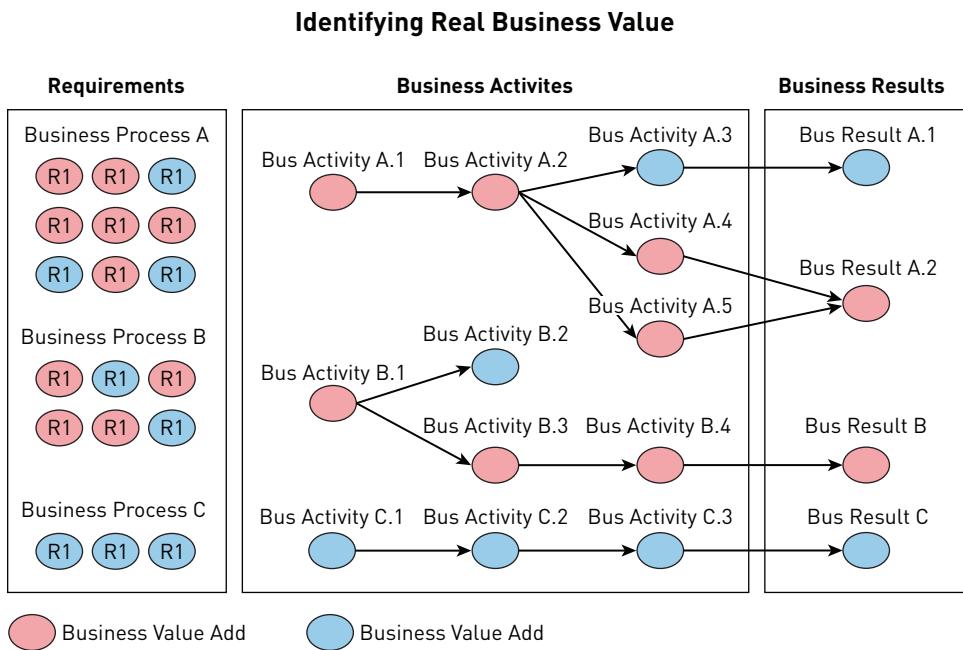
**Table 2-1**

MEASURE/ METRIC	TARGET/GOAL	SOURCE	TIMELINESS
Customer Retention	25% of customers have repeat purchases within 90 days	Sales/ERP	Daily/Rolling Time Period
Customer Loyalty	50% of customers have previously ordered within 180 days	Sales/ERP	Daily/Rolling Time Period
Customer Satisfaction	Average customer satisfaction survey score improves by 0.1 point versus quarter	Customer Satisfaction Surveys	Weekly/Quarter to Date
Customer Complaints	Reduce customer complaints by 5% versus prior quarter	CRM	Daily/Quarter to Date

As you review your target measures, make sure that they come from good, solid data within your organization or from an external source. Of course, don't put the cart before the horse. If there are items that you need to pay attention to that are external to the organization, the data needs to be evaluated. Review any compliance and regulatory requirements or markers that your organization may have and include them on your scorecard as good strategic items.

Think about the business that you're in. Are you in a well-established industry with well-established performance metrics? These can provide a great foundation for re-thinking, re-tooling, and becoming more innovative as your organization grows.

Consider also any key scope that needs to be included. Have any executive stakeholders or financial stakeholders said very clearly that this project must include X, Y, or Z to be considered a success? Those stakeholders, or more specifically their preferences, should be evaluated thoroughly to maintain their support of the project. Review the business requirements carefully. It may help to diagram them as illustrated in Figure 2-3.

**Figure 2-3**

### *Filtering and Prioritizing*

Once you're done collecting a raw group of measures, you should filter them down into a manageable list. Enlist key people to review and cut down the first round of measures. This group can include your stakeholders, peers, or a team of executives and managers. What you're looking for is validity and impact to the organization. Some may believe that particular measures are not as valid as other ones, and their opinions should be taken into account. If they believe that some metrics are more impactful than others, then you should prioritize them higher using the feedback you've gathered. The filtering stage is your opportunity to let the organization help you drive what is a realistic and reasonable scope for your scorecard.

### *Validating*

Once you've gone through the filtering process, you need to take the measures that were prioritized by your group and begin validating them. That means you need to source the data and the calculations to calculate and solve for the different measures. This might be the time to get IT or perhaps a business analyst involved, or whoever you need to access the data, show the steps for

calculating it, surface the data, and make sure that the process is repeatable and pragmatic in a way that can be displayed and shared with the organization. It's important that the data validation process becomes transparent after the initial discovery, because many people will ask, "How did you get that answer?" The data validation phase is critical because it establishes most of the credibility for the measures and the values that you're sharing with the organization.

## Selecting the Right Performance Metrics

---

There is a common saying in the business community, "You are what you measure." Incorrect, misleading, or just flat out wrong metrics can lead to issues such as poor strategic objectives or goal alignment, or a false sense of achievement. This type of issue is fairly common. One such example could be the result of focusing too much on a metric like profitability while losing sight of bigger picture metrics such as customer service or experience.

Selecting the correct performance metric, then, becomes a crucial step in the process of a successful dashboard or balance scorecard initiative. To get started, you first need to understand the two fundamental metric types:

- Metrics that identify problems or opportunities
- Metrics that measure performance

Both metrics types share four common dimensions:

- **Useful and Actionable:** Metrics that are too broad are neither useful nor actionable. For example, metrics such as employee satisfaction or customer satisfaction, while useful, are both very broad and typically leave your audience members with an unclear path on how their actions can improve the metric. Useful metrics provide a clear, actionable path for improvement within the context of the metric. As seen in Table 2-1, a more useful measure or metric would tie to customer complaints or more directly to sales because both are clear and the path to improvement more intuitive and actionable.
- **Relevant:** Metrics that are obscure or have multiple interpretations can lead to confusion when placed on a dashboard. Sales opportunities is a good example of a metric for which relevance can be an issue. If your Sales department has one definition of an opportunity, and your Marketing or Finance department has a competing definition, then a metric measuring sales opportunities would not be relevant for an executive dashboard. For a metric to be relevant, it should be clearly and universally understood by the intended audience without explanation.

- **Accessible and Credible:** The data used to produce a metric is often difficult to acquire, difficult to process, and occasionally even unreliable. Website usage data and the metrics they produce are notorious examples of this. To demonstrate, try to obtain key metrics, such as the number of unique visitors or the average time a user spends on your website. You will almost always find that competing analytical packages return widely varying measures produced from the same set of data. The metrics you choose (and the data from which they are derived) should be not only accessible to your audience but also credible.
- **Transparent:** When a metric is arbitrarily complex, it is typically not well understood. This can occur when multiple, more atomic metrics are combined together using a custom or proprietary formula to produce some new measure. Metrics that you include on your dashboard or balance scorecard should be transparent and easily understood.

## Planning Development and Implementation

---

Now that you have executive buy-in, you've gathered and documented your requirements, and selected the metrics that will be used to build your dashboard or balanced scorecard, it's time to consider the development and implementation process.

If your initiative is small, it could potentially be handled in a single project or phase. For most dashboard and balance scorecard projects, however, it's necessary to divide the larger overall project into smaller more manageable projects. These projects should be developed and implemented using an iterative or sprint-based development and implementation techniques based on the *Agile Development Methodology*.

The foundation of an Agile Development Methodology is an iterative or incremental approach where requirements and development are handled through collaboration of self-forming, cross-functional teams focused on continuous delivery. There are four tenets that can be derived from the Agile Manifesto (*Manifesto for Agile Software Development*, 2001):

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

This methodology fits well with dashboard and balanced scorecard projects, since these projects almost always require cross-functional teams and demand flexibility as the business and its strategies continue to evolve.

To identify the boundaries for these projects, most initiatives can be divided along naturally occurring lines, such as those between strategic objectives or departments. Once you have identified each project's boundaries, use the following considerations to prioritize the development and subsequent implementation of each:

- Is the project funded and are resources scheduled?
- Is there a sequence or are there dependencies among the projects identified?
- Are the data and required development and implementation resources available?
- Are there external projects or initiatives that will either impact the project or be dependent on the project?

After you've categorized and prioritized your projects, you can begin working on project management. An imperative part of your dashboard and balance scorecard initiative will fall to your project management team. Using a project management technique, such as a Gantt chart (see Chapter 13, "Decision-making Charts") enables your executive committee or sponsors to monitor the progress of the project without requiring a time commitment in their already busy schedule.

Regardless of the technique used for monitoring, communication among the entire project team is critical during the course of development and implementation.

## Managing Expectations and Scope

---

Managing expectations and scope throughout the project is very important. You must set and communicate expectations every time you deal with somebody on this particular effort, whether it's an executive stakeholder, a project participant, an IT person who's giving you resources and development time, or a business stakeholder who's giving you information about metrics and measures. If the project begins to get off track, people will lose confidence in the overall effort, and in turn, you and your efforts. Generally, there are many people involved in making this type of project a success, and so it's important to manage expectations and scope appropriately to reward their hard work.

Scorecards show strategy internally and externally and communicate confidence and direction of the organization. Therefore, it is essential that they're aligned. Managing expectations and scope helps keep the different scorecards

and projects aligned and moving in the right direction, and helps avoid missed goals. Missing goals creates reasonable doubt that the scorecard project will be a success, and when the project is focused on varying strategy for an organization, it could have a serious impact. Not setting expectations and scope correctly could result in a miscommunication of company strategy, which could result in disappointment for internal and external customers and, for some organizations, have a negative financial impact.

Best practice communication is also critical. A good chain of communication, a solid understanding of when problems should be escalated, and a vetted project management methodology should be in place. Project management methodologies are out of scope for this book, but you probably have an experienced project manager somewhere in your organization or could find this information very easily in another text.

Finally, it's essential to manage delays. Roadblocks often occur, but they usually only block one lane of the road, leaving the rest open for business. Many other tasks can be managed and moved forward and expectations can continue to be met. For those deliverables that cannot be achieved because of a particular roadblock, the issue needs to be communicated and escalated to the right people. You may not be able to get as much done as originally anticipated, but scope can be adjusted based on a delay. Often projects go very smoothly, but when delays occur, it's important to reevaluate your scope and make sure your expectations are realistic. If you continue to do this throughout the process, your first round of scorecards through your last will continue to be a reasonable and manageable process.

---

## Summary

---

There are a lot of pieces involved in managing, gathering, and defining the scope for your balanced scorecard implementation. You've selected your first round of executives, being careful to pick the right executives for your project. You've identified your ambassadors, the people who will help you build relationships with executives that you may not have already. Ambassadors may also be people who are friendly to your project and work on teams with which you may not have a strong relationship. You'll need somebody to bring you in and help sell the value of the project.

You've scheduled your first interviews and considered doing "analytics town hall" sessions. You've brought together the people who are in charge of driving the different stakeholder operations, as well as the various validation operations. You've ensured that they understand what's coming through the pipeline.

You've also looked at any project requirements necessary to get this effort going internally, such as cost centers, budget approval, project management

time—all of the things that can come into play in an organization as you kick-off new projects. Remember that your stakeholders' time is valuable, and it will be a lot easier to get traction with certain people if they know that their information and time is going to a specific project.

You've planned out your first agenda for those meetings, including specific items you need to discuss with the different individuals you'll meet. You are now in a great position to move forward to actual development. The next chapters take you through the actual development steps for the different scorecards and help you plan out the sprint timelines and individual deliverables based on what your scorecard goals really are.



# **Building and Extending Dashboards and Balanced Scorecards**

---

## **In This Part**

---

This section of the book targets report developers and introduces developing and delivering both a dashboard and balanced scorecard using Microsoft SQL Server 2012 Reporting Services. It assumes some knowledge of Reporting Services. You will walk through building a progressively more advanced solution. This section covers how to build a dashboard and a scorecard and shows some examples of dashboards. It does not go into detail on every feature in Reporting Services; however, it does cover how to enhance your dashboards with interactivity and visualizations and how to deploy dashboards.

- Chapter 3:** Building Your First Dashboard and Balanced Scorecard
- Chapter 4:** Enhancing Your Dashboard and Balanced Scorecard
- Chapter 5:** Interactive Dashboards and Balanced Scorecards
- Chapter 6:** Visual Dashboards and Balanced Scorecards
- Chapter 7:** Mobile Dashboards and Balanced Scorecards
- Chapter 8:** Extending Dashboards and Balanced Scorecards  
with Custom Code
- Chapter 9:** Publishing Dashboards and Balanced Scorecards



# Building Your First Dashboard and Balanced Scorecard

This chapter covers building reports in Reporting Services from the ground up to build a dashboard and a balanced scorecard. No previous Reporting Services knowledge is assumed, so the basics are covered first. If you already have a working knowledge of Reporting Services, you are free to skip ahead to Chapter 5. Keep in mind, the information in this chapter is helpful in completing any knowledge gaps. It assumes that SQL Server 2012 Reporting Services is installed and configured. Installation is not covered and can be researched by checking SQL Server books online at [http://msdn.microsoft.com/en-us/library/ms130214\(v=sql.110\).aspx](http://msdn.microsoft.com/en-us/library/ms130214(v=sql.110).aspx).

This chapter covers how to build your first dashboard in a step by step example. The chapter explains:

- What is Reporting Services?
- What is a Data Source?
- What is a Dataset?
- How to build a Dashboard

## Anatomy of a Reporting Services Project

---

Microsoft SQL Server 2012 Reporting Services is a powerful and large tool that enables you to create reports, scorecards, and dashboards with great detail. Reporting Services gives you complete control of every item on-screen down to the pixel and color.

A Reporting Services project typically contains these items:

- **Reports:** The RDL file that is the actual report.
- **Data sources:** These typically consist of a connection string to a database.
- **Datasets:** These include the query or stored procedure used to select the appropriate data from the data source for a report.

These items are all housed within a Reporting Services project, which is created in SQL Server Data Tools (formerly known as Business Intelligence Development Studio or BIDS).

To get started on a report project, open SQL Server Data Tools (SSDT). This is found on the Start menu under SQL Server 2012. If you are running Windows 8, press the Windows key on the keyboard, type **SQL**, and click SSDT in the list that appears. This opens SSDT at the home screen. To create a Reporting Services project, perform the following steps:

1. Click File in the top-left corner of SSDT and select New > Project, as shown in Figure 3-1. The New Project screen will open at this point. Select Report Server Project under the Installed Templates list on the left.

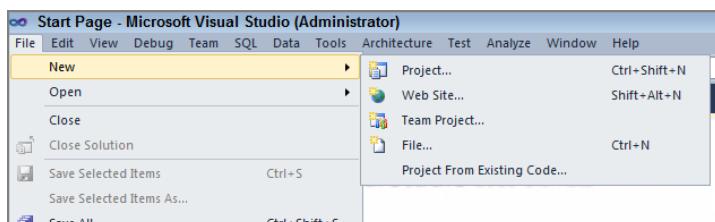


Figure 3-1

2. In the center of the New Project window, there are two Report Server options, as shown in Figure 3-2. The first option runs a wizard to walk you through creating a report. The second option starts a blank Report Services Project. Select the second option to create a blank project. At the bottom of the window type **My First SSRS Project** in the name box, then click OK.

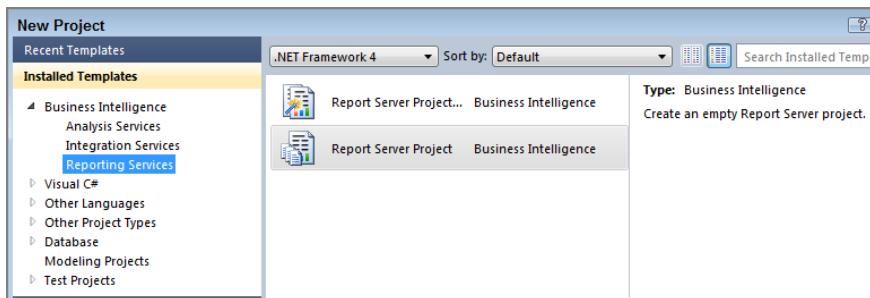


Figure 3-2

3. Ensure the Solution Explorer appears on the right-hand side of SSDT, as shown in Figure 3-3. The Solution Explorer is the window that contains all the parts of a project. This is used to add or delete items from the project.

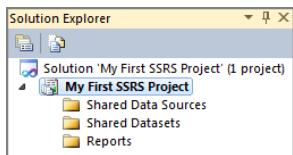


Figure 3-3

The first folder you see in the Solution Explorer window is Shared Data Sources, which contains all the connections the reports in this project use to connect to a database, an Analysis Services Cube, or some other data source. The second folder is Shared Datasets. It contains all of the datasets that have been set to shared. A *shared dataset* can be used by multiple reports, even ones not in the same project. A *dataset* contains a query or stored procedure used to pull data from a data source. The third folder in the Solution Explorer is the Reports folder. It contains all the reports in the project. The next section explains the Solution Explorer items in detail.

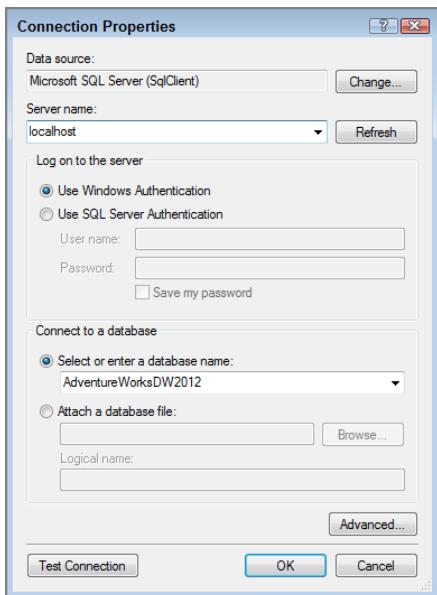
**NOTE** If you don't see Solution Explorer, click View and select Solution Explorer from the text header menu in the project.

## Introduction to Data Sources, Datasets, and Controls

With the project created, it is time to start building the items needed in the Solution Explorer. The first item to create is the data source.

1. Right-click the Shared Data Sources folder in the Solution Explorer and select Add New Data Source.

2. Name the data source **AWDW** and make sure Microsoft SQL Server is selected in the type drop-down menu.
3. Click the Edit button to open the Connection Properties. The Server name and Database name are needed to create a connection.
4. Enter your Server name. If you are using a local instance then you can enter **localhost**. Then select **AdventureWorksDW2012** for the database name, as shown in Figure 3-4.
5. Test the connection by clicking the Test Connection button to ensure the data source is valid. Then click OK in both windows to close them. The new AWDW connection will now show in the Solution Explorer under the Shared Data Sources folder.



**Figure 3-4**

**NOTE** The following examples use the database **AdventureWorksDW2012** and the Adventure Works SSAS cube database. Both can be downloaded from <http://Codeplex.com>.

The next step is to create a report. A report is created instead of a shared dataset because any dataset can be converted to a shared dataset after the report has been created. Not all datasets need to be shared, only datasets that will be used by multiple reports. A good example of a dataset that should be shared is a list of employees or locations.

1. Right-click the Report folder in the Solution Explorer and select Add, then click New Item.
2. In the Add New Item window, select Report.
3. In the bottom of the window, enter **My First Report** as the name of the report. Then click Add.
4. Ensure the Report Data window appears on the left. If it does not, press **CTRL+ALT+D** to make it appear. This Report Data window is used to add datasets and data sources to the report.
5. Now add the data source to the report from Shared Data Sources. Right-click the Data Sources folder in the Report Data window as shown in Figure 3-5 and select Add Data Source.

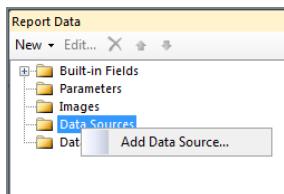


Figure 3-5

6. Change the name of the data source to **AWDW**.
7. Check the radio button to Use shared data source reference.
8. Click the lower drop-down menu and select AWDW, as shown in Figure 3-6, then click OK.

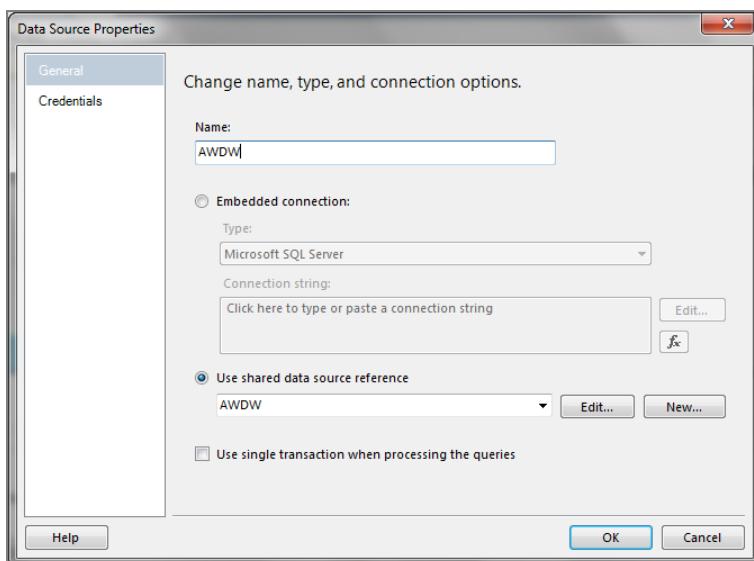


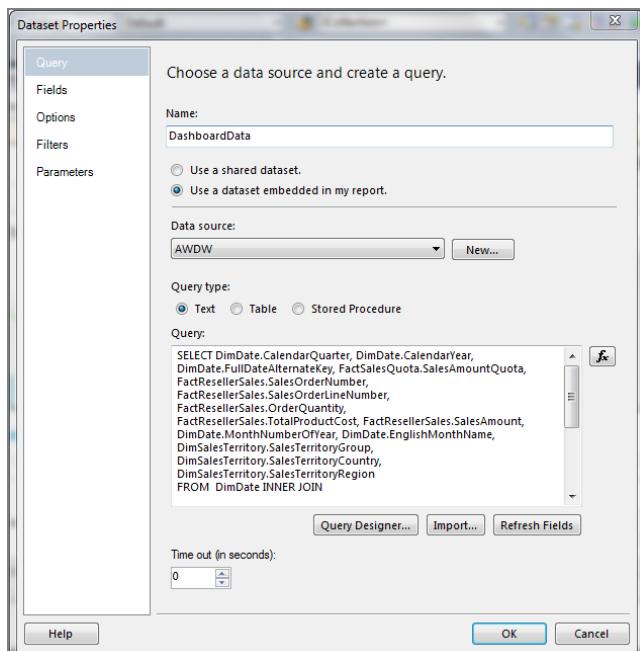
Figure 3-6

9. The last step is to add the dataset to the report. Right-click the Datasets folder in the Report Data window and select Add Dataset.
10. Change the name of the dataset to DashboardData.
11. Check the radio button to set the dataset to embedded.
12. Select AWDW in the Data Source drop-down menu.
13. Enter the following query into the Query window (see Figure 3-7), then click OK.

```

SELECT DimDate.CalendarQuarter, DimDate.CalendarYear,
DimDate.FullDateAlternateKey, FactSalesQuota.SalesAmountQuota,
FactResellerSales.SalesOrderNumber, FactResellerSales.SalesOrderLineNumber,
FactResellerSales.OrderQuantity,
FactResellerSales.TotalProductCost, FactResellerSales.SalesAmount,
DimDate.MonthNumberOfYear, DimDate.EnglishMonthName,
DimSalesTerritory.SalesTerritoryGroup, DimSalesTerritory.SalesTerritoryCountry,
DimSalesTerritory.SalesTerritoryRegion
FROM DimDate INNER JOIN
FactResellerSales ON FactResellerSales.ShipDateKey = DimDate.DateKey INNER JOIN
DimSalesTerritory ON FactResellerSales.SalesTerritoryKey =
DimSalesTerritory.SalesTerritoryKey INNER JOIN
DimEmployee ON FactResellerSales.EmployeeKey = DimEmployee.EmployeeKey inner JOIN
FactSalesQuota ON DimEmployee.EmployeeKey = FactSalesQuota.EmployeeKey and
FactSalesQuota.CalendarYear = DimDate.CalendarYear and
FactSalesQuota.CalendarQuarter = DimDate.CalendarQuarter

```



**Figure 3-7**

The Solution, Project, and Report are created now and the Shared Data Source is a part of the report. The next step is to build a simple report with this data to demonstrate how to design and preview reports.

## Design and Preview

In this section you build a simple table report with the data source and dataset created in the previous section. This will help explain the basics of using the tools in the Toolbox and the overall design process. A discussion of all the design features of Reporting Services is beyond the scope of this book.

On the left side of the screen there is a Toolbox (shown in Figure 3-8) that contains all the components used to build a report. If the Toolbox does not show, click the View menu at the top and select the Toolbox to make it appear. Follow these steps to build a simple table report.

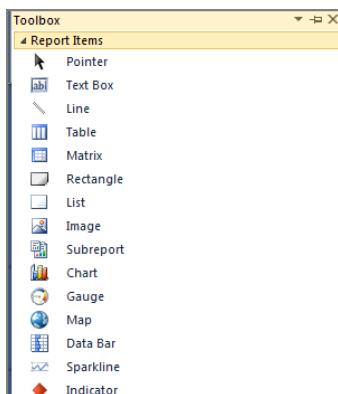


Figure 3-8

1. Drag a Table from the Toolbox onto the Design pane in the center of the screen. The table has a header section across the top and a data section across the bottom.
2. Click the data section of the table in the first column on the left. A small blue icon will appear. This icon will open a drop-down menu of all the columns in the dataset in the project as shown in Figure 3-9.
3. Select `CalendarYear` for the first column.
4. Select `CalendarQuarter` for the second column.
5. Select `SalesAmount` for the third column.

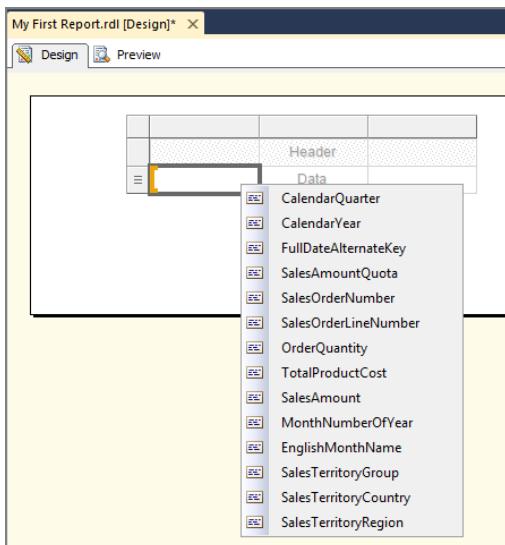


Figure 3-9

6. Click the Preview tab at the top of the Design pane to view the report. The report should look similar to that shown in Figure 3-10.
7. Click the Design tab to return to design mode.

A screenshot of the Microsoft SQL Server Report Designer interface in Preview mode. The title bar says "My First Report.rdl [Design]". Below it is a toolbar with "Design" and "Preview" buttons. The preview area shows a table with three columns: "Calendar Year", "Calendar Quarter", and "Sales Amount". The data is as follows:

Calendar Year	Calendar Quarter	Sales Amount
2005	3	2024.9940
2005	3	6074.9820
2005	3	2024.9940
2005	3	2039.9940
2005	3	2039.9940
2005	3	4079.9880
2005	3	2039.9940
2005	3	86.5212
2005	3	28.8404
2005	3	34.2000
2005	3	10.3730
2005	3	80.7460
2005	3	419.4589
2005	3	874.7940
2005	3	809.7600

Figure 3-10

Notice that the sales amount needs to be formatted and overall the report is just not attractive. This is where the cleanup and design work comes into play. There is almost an unlimited number of formatting options available in Reporting Services. The next series of steps covers the most commonly used formatting options, which you can use to make the report look much cleaner.

1. Click the Report text toolbar and select Add Page Header from the drop-down menu.
2. Drag in a Textbox from the Toolbox into the Report Header.
3. Type **My First Report** in the text box.
4. Right-click the text box and select Textbox Properties. It may be necessary to click away from the text box first to see this menu; otherwise it may select the actual text instead of the text box.
5. In Textbox Properties, select the Font option on the left.
6. Change the font to Calibri, the size to 24, and check the bold option, then click OK.
7. Stretch the text box out to show the entire text and drag it into the top-right corner. The text should look like Figure 3-11 now.

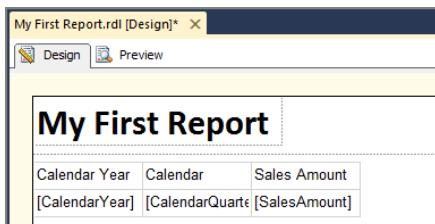


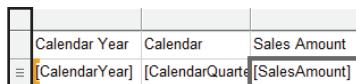
Figure 3-11

The Textbox Properties screen is one of the most common tools used in Reporting Services. It is used to format text in the desired manner. There are also sorting and other actions available that will be covered in Chapter 5, “Interactive Dashboards and Balanced Scorecards.”

The following steps will finish the cleanup of the data on the report.

1. Right-click the Sales Amount data field in the table and select Textbox Properties.
2. Click Number on the left.
3. Click Currency in the Category list.

4. Set the decimal places to 0 and check the Use 1000 separator option. This will show the money in a proper format. Click OK.
5. Click any cell in the header row and a set of boxes will appear on the top and left side of the table, as shown in Figure 3-12.

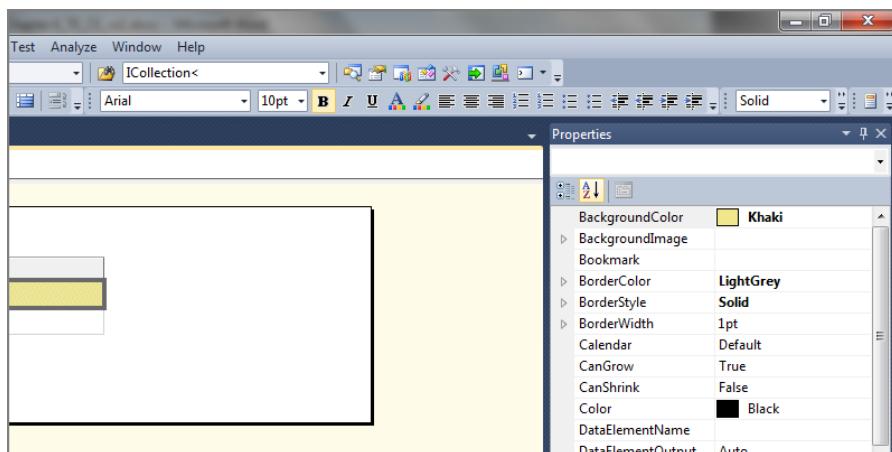


A screenshot of a Microsoft Access table. The table has three columns: 'Calendar Year', 'Calendar', and 'Sales Amount'. The header row is selected, indicated by a thick black border and handles (small squares) at the corners and midpoints of the row. The first column contains the value 'CalendarYear', the second 'CalendarQuarter', and the third 'SalesAmount'.

**Figure 3-12**

The boxes surrounding the table are handles used to control the table and are used to select rows and columns. When an edit is needed on an entire row or column, these handles should be clicked to select the appropriate items.

6. Click the handle on the left of the header row and the entire header row should be selected.
7. Open the Properties window on the right side of the screen. Look for the `BackgroundColor` property in this window, click the drop-down next to it, and select a color for the header background. A light color is best since the header is black text.
8. Use the toolbar at the top of the menu to set the text to bold. Figure 3-13 shows the bold and background color options.
9. Drag the side of the report to match up with the report and the headers.
10. Click the Preview tab to see the finished and formatted report, which should resemble Figure 3-14.



**Figure 3-13**



Figure 3-14

You should now have a complete report with some formatting. Building a dashboard will require many more SQL Server Report Service controls though. Table 3-1 provides an overview for the controls that are provided out of the box.

**Table 3-1: SQL Server Report Services Controls Toolbox**

CONTROL	DESCRIPTION
Textbox	Used to display titles, parameter choices, metrics, and other textual information.
Line	A visual line for which style, weight, and color can be formatted.
Table	Used to display detailed data that can be grouped by one or more columns. The table is formatted in rows and columns.
Matrix	Similar to the table control, displays grouped and summarized data. Commonly referred to as a crosstab or pivot table report.
Rectangle	A visual element used as a container to group controls within a report.
Subreport	A report item that references another report.
List	A data region or container of other controls that repeats for each group or row within a dataset.
Image	An internally embedded or external image.
Chart	The report item capable of rendering bar, line, pie, and other chart types from a dataset.

*Continued*

**Table 3-1** (continued)

CONTROL	DESCRIPTION
Gauge	A visual element used to display a metric. Typically used to show a metric value against a defined goal.
Map	A geographical, geospatial, or ESRI (Environmental System Research Institute) formatted map that can include data layers to display data points.
Data Bar	A visual element useful for showing ratios and correlations between multiple rows within a table.
Sparkline	A visual element useful for showing trends.
Indicator	A set of icons that are used to show metric values and trends.

Next, you will build a complete dashboard using this same dataset and data source used in the previous example. This dashboard will be much more complex and use a lot more of the features in Reporting Services.

## Building Your First Dashboard

The main purpose of a dashboard is to show the operational or tactical health/status of an organization, or part of an organization, at a glance. A user should be able to look at a dashboard for about five seconds and see any problems easily at a high level. There are several ways to accomplish this in Reporting Services and several tools that can be used to build a dashboard. Before you get started building a dashboard though, it is important to decide the scope or purpose of the dashboard. You'll then want to choose among design options and gather data for the dashboard. *Key Performance Indicators (KPIs)* should also be considered when preparing to build your dashboard. The following sections walk you through these important steps and more for building your first dashboard.

## Scoping and Requirements for Your Dashboard

Scoping a dashboard can be thought of as deciding what is going to go on the dashboard. A great way to choose the items that need to be on a dashboard is to ask yourself the following questions:

- What business questions am I trying to answer?
- What are the pain points I need to watch?
- What information would be most helpful?

These questions are used in modeling just about any business intelligence solution and can also help with designing a data warehouse. The Adventure Works data sources used in this chapter already have a data warehouse design and model, but the data warehouse contains a lot of data, and showing all of that on a dashboard would be problematic. So, it is important to narrow down the data to answer specific questions.

## Considerations Before You Get Started

The business intelligence industry is littered with stories about dashboard projects that, although designed in grandiose fashion, failed and ultimately delivered little if any value. That's the bad news. The good news is that if you consider the following tips, your dashboard project will be better positioned for success.

1. **Limit the scope.** It's tempting when you get started on a dashboard project to want to take on the entire organization as your audience. Instead, start small. Choose a manageable scope, such as an existing report that is well defined and will provide a quick return on investment.
2. **Remember, the business owns the dashboard, not the IT department.** It is critical that the business not only own the dashboard but that it leads the dashboard initiative. For a dashboard to be adopted and become an integral part of a company's culture, IT must resist the urge to take ownership. The role of IT should be strictly that of enabler.
3. **Limit the dashboard's metrics to the "critical few."** Identifying and selecting the correct metrics is often one of the most difficult aspects of a dashboard project. Resist the urge to include all your metrics for a given process or business unit. There is a saying that having too many metrics on a dashboard can be as bad as having the wrong metrics. Instead, focus on the "critical few" that are most informative and best aligned with the operational or tactical goals of the dashboard.
4. **Be agile.** Deliver early and often during the dashboard project. Work iteratively, ensuring that each iteration delivers a complete and working dashboard. A good rule of thumb is to keep these iterations short, or around 30 days.

## Identifying Metrics and Gathering Data

Unless you are starting with an existing report, data warehouse, or SQL Server Analysis Service cube, among the first things you need to do is identify metrics and source data, and subsequently gather data. Although only a small section of this book is dedicated to this topic, your dashboard will be built on top of this work, and the bulk of the project time for dashboard projects is spent here.

### *Identifying Metrics*

After you have identified the process or tactical objectives for your dashboard, you are ready to begin the metric identification and selection process. This process usually begins by interviewing subject matter experts and decision makers from within the business. To get started, consider the following questions that are commonly asked during these interviews.

- Which business decisions are made routinely?
- Are there any critical business decisions?
- What information is used to make the previously identified business decisions?
- How do you measure the effect of a business decision?
- Are there trends that are monitored?
- What signals an alert?
- Are there floor/ceiling values that are considered?

From these questions, you should be able to model an initial list of metrics. The next step is to prioritize each metric so that you can arrive at those “critical few” metrics mentioned previously.

### *Gathering Data*

For most dashboard projects the data you need is readily available, and if it is not already in a data warehouse or operational data store (ODS), it can often be found in line of business (LOB) applications, enterprise resource planning (ERP), customer relationship management (CRM), or other systems which are already being reported against.

In cases where the data is not readily available, it may be necessary to design and plan the data collection process, which is often referred to as an *Extract-Transform-Load* (ETL) process. This process can be time-consuming, as it is often necessary to perform tasks such as data profiling, data mapping, and data quality checks to ensure that the data is dashboard ready.

## **Working with Metrics and KPIs**

Metrics and key performance indicators (KPIs) definition are the primary work products derived from the steps described in the previous sections. After you have identified your “critical few” metrics (or KPIs), it’s time to implement them for use in your Reporting Services dashboard.

The implementation method you choose is determined by the environment in which you are working. If you are retrieving data directly from a source system,

such as your ERP or CRM, the easiest implementation method for your metrics and KPIs is to either directly embed them as calculations in your dashboard (see Chapter 4, “Enhancing Your Dashboard and Balanced Scorecard” for more information) or define them as calculated values within your source queries.

Either implementation option is straightforward, but they both have considerations of which you need to be aware.

- Embedding metrics in your dashboard prohibits reuse of the metric in subsequent or alternate view dashboards. This can lead to having multiple definitions for the same metric, which could cause maintenance and versioning issues.
- Depending on the complexity of the metric calculation, you could do one of two things:
  - If embedded in a report, you could slow down the rendering of the report by making the SQL Server Reporting Services engine work harder to resolve the metric.
  - If embedded in a source system query, you could slow down the source system by introducing contention or consuming valuable database resources.

**NOTE** A better option for both your metrics and key performance indicators is to extract them into either your data warehouse or SQL Server Analysis Services cube environment, if one is available.

Both of these options overcome the weaknesses previously identified by enabling the calculation of your metrics to be off-loaded to a business intelligence system that is designed specifically to handle this type of workload; while allowing your metrics and KPIs to be consumed by multiple dashboards, reports, and other applications, all from a single source.

After you have chosen and implemented each metric, it is critical that you test, test, and then re-test to ensure that metrics are being calculated correctly and that they support the goals laid out for the dashboard. Incorrect metrics, whether poorly chosen or created through faulty calculation, render all other work meaningless.

## Designing the Dashboard

At this point, you are ready for the fun part of your dashboard project. This is where all your hard work in requirements gathering, scope definition, and metric identification come together and start to take shape as you define the physical or visual design of the dashboard. While this section doesn’t cover specific visual design rules (more on that in Chapter 6, “Visual Dashboards and Balanced Scorecards”), there are several points to be aware of as you get started.

Your dashboard is not just another report. It's a decision-making tool, and as such there are certain considerations that need to be made in context of the visual design. Use the following guidelines to get your layout going:

- Limit your dashboard to a single page (or screen) so your audience is not required to scroll or flip pages.
- Group metrics and information by decision, process, or tactical goal. Visually, there should be clear delineations between subject areas.
- Emphasize information, not raw data. Make the raw or source data available, but don't include it in your dashboard.
- Ensure that the information included on the dashboard directly supports the "critical few" metrics so that your dashboard is actionable.

With these guidelines in mind you are ready to choose visual representations for your metrics. Broadly, the options you have available within Reporting Services are:

- **Textual:** Numbers, percentages, and descriptions defined in text boxes, tables, and matrices.
- **Indicators:** Typically associated with KPIs and can display both values in terms of a goal as well as a trend.
- **Gauges:** Useful for showing metrics that have a value and a goal.
- **Charts:** Numerous forms of charts are available, including data bars and Sparklines. Typically used on dashboards to identify trends, highlight comparisons, or show correlations.

In Chapters 4 and Chapter 6, you will dive deeper into each of these respective areas to gain a better understanding on when to choose each method and how they are implemented.

---

## Summary

This chapter introduced you to both the SQL Server Reporting Services environment as well as the process needed to plan, design, and develop your first dashboard.

Recall that the foundation of your dashboard in Reporting Services is the data source definition and the report dataset. Report controls such as text boxes, tables, gauges, and charts are the visual implementation of the metrics identified in the second half of this chapter.

In subsequent chapters, you will build on these fundamental concepts as you work toward a complete performance dashboard and balanced scorecard.

# Enhancing Your Dashboard and Balanced Scorecard

Regardless of the company, business requirements, or report type, an analytical report must be well organized, focused, and actionable to be ultimately useful. This chapter discusses techniques that accomplish and implement these high-level objectives, which are available as enhancements to build on the reports you started in Chapter 3, “Building your First Dashboard and Balanced Scorecard.” The techniques discussed in this chapter help you do the following:

- Achieve visual consistency through the use of strategically placed layouts and headers and footers
- Organize reports with paging and grouping
- Apply conditional formatting to report elements
- Build dynamic and data-driven labels and titles

## Organizing and Achieving Visual Consistency

---

The art and science of data visualization is a vast field about which entire books are written. While Chapter 6, “Visual Dashboards and Balanced Scorecards,” is largely devoted to diving deeper into the core principles of this field, this section deals with the delivery of a consistent look and feel either among multiple dashboards or, as you will see soon, in multipage dashboards.

## Placement and Layouts

As you begin creating balanced scorecards and dashboards, you'll be faced with a number of design options. Beyond ensuring that you have selected correct and relevant metrics, one of the first decisions you will inevitably have to make is the organizational placement of each metric or measure within your report.

Special care and consideration at this step is needed because poor layout or organization can be just as disastrous in terms of report effectiveness as selecting the wrong metrics or having bad data.

Some of the issues that can be manifested from poor placement or layout are:

- Unimportant metrics or data obscuring more important ones by being placed more prominently.
- Incorrect layout of metrics and measures implying association between data points that aren't connected or, alternatively, disconnecting metrics that are connected.
- Excessive (or not enough) white space distracting or cluttering the report.

To avoid these issues, start by taking a moment to understand abstractly how human beings see and process information.

### *Visual Cognition*

As human beings, we all visually see and process data in a similar manner. We have a built-in ability to recognize both simple and complex patterns, infer groups even when no borders or containers exists, and derive information by interpreting colors and shapes. While the science behind how this process occurs is beyond the scope of both this chapter and this book, understanding how we read and interpret data will help you arrive at an effective layout.

### *Left-to-right*

Although this varies by culture, the order in which your dashboard or other report is read infers a precedence or level of importance. Because western cultures read left-to-right and top-to-bottom, studies have repeatedly shown that the top-left portion of most any document or report is the most valuable in terms of real estate.

To illustrate this, you only need to open a web browser and navigate to your favorite website. Where did they put their logo? How about the website's navigation? In the same manner that web designers use layout for effectiveness, you can apply the same principles when delivering analytical reports.

A few guidelines to consider as you begin planning your layout are:

- The top-left portion of your report receives the most emphasis and should contain the most important data. This space should be reserved and should not contain non-data elements such as navigation links, report instructions, or items that are only decorative in nature.
- The further to the right and down on a report that a metric or measure appears, the less emphasis it will receive when compared to what appears above and to the left of it.
- Metrics or measures placed in the bottom-right of any report are often overlooked.

## Dashboard Headers and Footers

Headers and footers are typically used in reports when applying the corporate branding or identity. With dashboards and analytical reports, care needs to be taken, particularly when designing headers so they don't distract from data or the story the report is telling.

Beyond those considerations, both headers and footers are useful for the inclusion of report metadata such as a title, page number (if relevant), and a processing date or date the report was generated.

To add headers and footers to your dashboard, simply right-click the report design surface to launch the Context menu and then select Insert. On the Insert menu there are options to add a report header and footer.

Both the header and footer can contain a number of report items that you were introduced to in Chapter 3. The image component and text box are the two used most often for the inclusion of a company logo, title, and other metadata.

**WARNING** Some of the Reporting Services controls cannot be used in either headers or footers. Typically any control that is data bound cannot be used. These include the following controls:

- Matrix
- Table
- SubReport
- List
- Gauges
- Map
- Data Bars
- Sparklines
- Indicators

When considering what metadata to include on your dashboard, it is good practice to include a report date to easily identify when the dashboard was last refreshed or printed. Other metadata might include contextual comments or analyses, footnotes, parameter names and values, and page numbers, if applicable.

**NOTE** Once you have settled on a look and feel for your project, consider creating a template report to avoid having to continually re-create the headers and footers. Simply copy and paste from the template when adding a new report or copy to the following directories to add it to the New Item/Report dialog box:

- **For SQL Server 2008:** C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\IDE\PrivateAssemblies\ProjectItems\ReportProject
- **For SQL Server 2012:** C:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\IDE\PrivateAssemblies\ProjectItems\ReportProject

It might be tempting to throw the kitchen sink at your dashboard by including every metric available, but organizationally speaking, you should avoid this temptation as it creates clutter and hides the message you are conveying. Instead, focus on commanding the viewer's attention by limiting each section or page of your dashboard to a single decision. This introduces the concept of the multipage dashboard, which offers critical information in concise sets to limit the noise and drive the focus of your audience.

## Reports, Subreports, and Paging

After you have organized your dashboard into multiple pages, the most common implementation option is to divide each page into its own Reporting Services report. You then bring each page of your dashboard together into a single report using the subreport component. To make paging work in this environment, you must add the subreport components to a rectangle control to allow for the configuration of page breaks.

### Paging Step-by-Step

1. Drag a rectangle from the Toolbox to the report design surface.
2. Add a subreport component to the rectangle by dragging it from the Toolbox and dropping it within the boundaries of the rectangle.
3. Configure the subreport by right-clicking the component to show the Context menu and selecting Subreport Properties option. Select the name of the report you wish to embed at the Use this report as a subreport prompt. Be sure to map any required parameters. At this point, you might need to resize the rectangle to accommodate the embedded report.

4. Now configure the rectangle to force a page break. Right-click the rectangle and select Rectangle Properties. There are two options: one for adding a page break before and one for adding a page break after. After you have made your selection, click OK (see Figure 4-1) and preview your report.

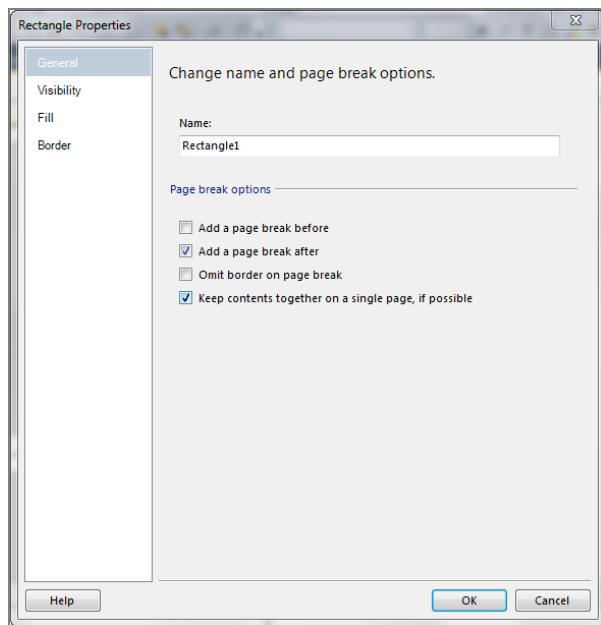


Figure 4-1

## Formatting Dashboard Elements

Reporting Services enables you to create information-rich dashboards and balanced scorecards that contain significant amounts of data but are still easy to understand. Making your report effective, however, requires more than just packing data into the dashboard. This section focuses on formatting Reporting Services elements to add value to the data through basic formatting. You will also learn about expressions and conditional formatting methods to call out important metrics or to alert your audience to a specific condition.

### Fonts and Text Boxes

Although the text box is the primary control used for labels and titles, the text used to display labels and data in your dashboard is found in nearly every control. Formatting and styling each text element becomes especially important to tell the story of the data and to drive your customer experience.

If you have worked with Microsoft products in the past then you are probably familiar with the common formatting options that are available for text in Reporting Services. The Report Formatting toolbar, which is enabled by default, allows for straightforward configuration of most text properties including font, size, color, style, effects, and alignment. In addition, you can control the text formatting options on the Font page of the Textbox Properties window in the Properties Windows (available by pressing F4) as well as from the Format menu on the toolbar (see Figure 4-2).



Figure 4-2

**NOTE** You are not limited to a single style within a text box. Rich text formatting is supported to enable the presentation of well-structured text. Limited HTML content is also supported within text boxes (see Figure 4-3).

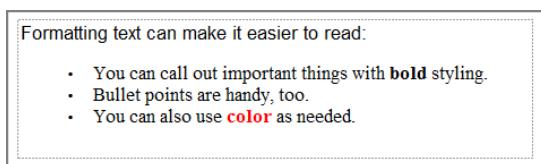


Figure 4-3

**NOTE** To rotate a textbox within Reporting Services, set the Writing Mode property in the Properties window. The supported modes are default, horizontal, vertical, and rotated 270 degrees.

## Borders, Lines, Color, and Images

Colors, borders, and lines are important when it comes to visually tying distinct elements together to make your dashboard more cohesive and easier to read. Reporting Services has standard border and line templates that make styling simple. Line style (solid, dotted, dashed, etc.), width, and color are all configurable on the Border property page of the Rectangle Properties dialog (see Figure 4-4).

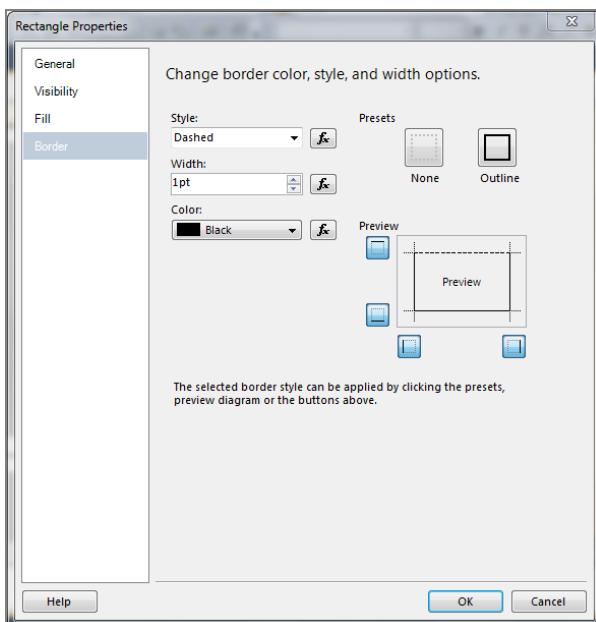


Figure 4-4

You can add graphics or images to reports in Reporting Services either using the Image Control or setting an image as the report background in the Report Body Properties. Both options for adding graphics support three methods for image inclusion:

- **Embedded:** The figure or graphic is encoded using a MIME format and stored directly in the report definition. This method ensures that the image is always available but increases the size of the report and restricts its usage, as it cannot be shared among reports.
- **External:** As the name implies, the image is external to the report and is hosted or served either from a remote web server or from the Reporting Services project, if it's included as a project item. When you include a graphic as a project item, it should be referenced using a relative URL. This method for including images is useful when an image needs to be shared among multiple reports, which is typical for images used in the report header. The risk associated with this method is that the resource might be unavailable when the report is rendered.
- **Data Bound:** Images included using the data bound method are stored in and subsequently retrieved from a database. This method enables images to be shared among multiple reports with a dependency on the database connection. Note that there are added complexities to hosting and managing images within a database instance.

## Text, Numbers, and Dates

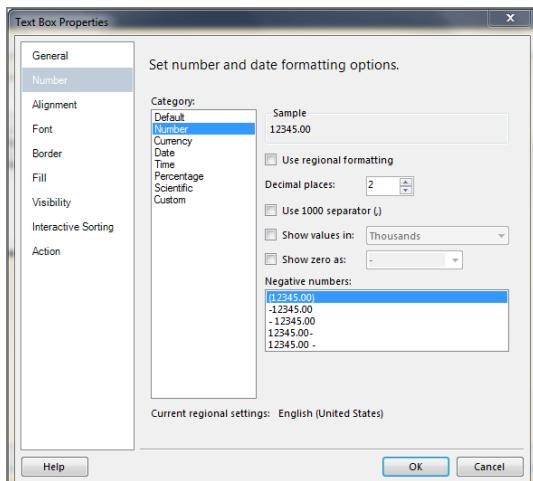
Formatting elements within your dashboard extends beyond styling fonts and borders. Formatting data elements makes them easier to read and interpret and will ultimately make or break your dashboard.

Typically, data is formatted based on its type, such as a currency notation for monetary-based metrics, scaled percentages, or a standardized YYYY/MM/DD for dates. Microsoft SQL Server Reporting Services has a built-in set of formats to simplify this process.

The following format sets are available out of the box:

- **Default:** No formatting; data is presented as-is
- **Number:** Supports thousands separator, decimal place rounding, and scaling to the thousands, millions, and billions
- **Currency:** Similar to the number format, with the addition of support for currency symbols and symbol placement
- **Date:** Supports standard date formats
- **Time:** Supports standard time formats
- **Percentage:** Supports specifying the number of significant digits
- **Scientific:** Supports scientific notation with a configurable decimal place
- **Custom:** A customizable format in which you use symbols to define the required pattern

Although the built-in formats are fairly inclusive (see Figure 4-5), it is not unusual to utilize the Custom format. Working with the Custom format can be intimidating at first glance, but as you can see in Table 4-1 the available symbols are limited, which makes working with Custom format very easy.



**Figure 4-5**

**Table 4-1:** Custom Format Symbols and Codes

DATA TYPE	SYMBOL/CODE	DISPLAY RESULT
Number	#	A numeric placeholder. #,### displays a whole number (rounded if necessary) with no trailing digit if the digit is absent.
Number	0	A numeric placeholder that preserves the trailing digit if it is absent.
Number	,	Thousands placeholder.
Number	.	Decimal point placeholder.
Number	_	The underscore (_) leaves a place in the format string. This is typically used to create a space between a currency symbol or positive/negative sign.
Number	;	Used to define different formats for positive and negative numbers; for example, 0.00;{0.00}.
Date/Time	m	Single-digit month number.
Date/Time	mm	Two-digit month number; zero padded when necessary.
Date/Time	mmm	Month abbreviation.
Date/Time	mmmm	Month full name.
Date/Time	d	Single-digit day of month.
Date/Time	dd	Two-digit day of month; zero padded when necessary.
Date/Time	ddd	Day abbreviation.
Date/Time	dddd	Calendar day name.
Date/Time	y	Single-digit year.
Date/Time	yy	Two-digit year.
Date/Time	yyyy	Four-digit year.
Date/Time	HH	12-hour clock hour.
Date/Time	hh	24-hour clock hour.
Date/Time	mm:ss	Minutes and seconds.

## Conditional Formatting

---

After you've applied base formatting to your dashboard, you might find it necessary to alter the formatting of elements within your report based on data. For example, you might want to clarify data by applying alternating colors to rows or draw the attention of your audience to some predefined deviation.

In any case, Reporting Services has rich support for conditional formatting through the use of expressions.

## Expressions

The expression language in Reporting Services is implemented using Visual Basic and is capable of controlling both content and report appearance. In addition to writing Visual Basic code, you have the option to use built-in functions, custom code in custom assemblies, as well as reports, groups, and user-defined variables. The basics are covered here, and Chapter 8, "Extending your Dashboard and Balanced Scorecard with Custom Code," offers a deep dive, covering both scripting and expressions in detail.

The expression most used in defining conditional formats uses the Visual Basic `IIF` statement. This statement is a shorthand notation of the `IF-THEN-ELSE` statement and takes the following form:

```
=IIF(<Condition>, <True>, <False>)
```

The condition must evaluate to a boolean result and can reference the fields in the dataset. The true and false placeholders are then used based on the result of the evaluation of the condition. Look at an example of an expression that could be used to change the color of the text for profit margin:

```
=IIF(Fields!ProfitMargin.Value < 0, "Red", "Black")
```

Using this expression, Figure 4-6 compares the value in the `SalesAmount` field to zero. If the `SalesAmount` is negative, the condition will evaluate to true and the font color is set to red. If the `SalesAmount` is greater than or equal to zero, the font becomes black.

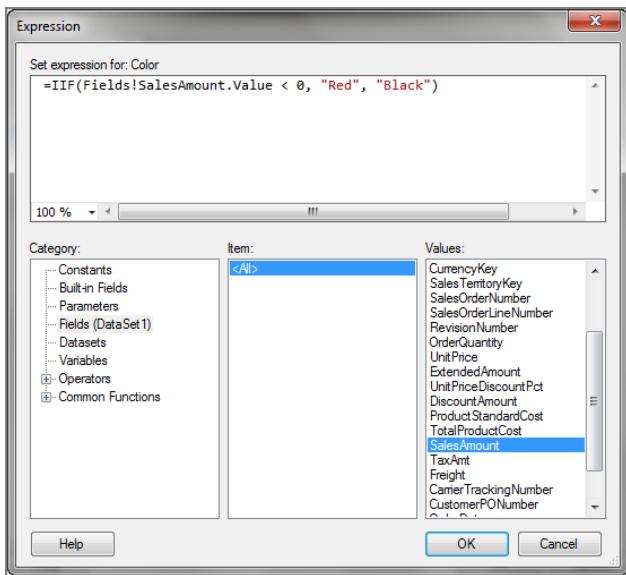


Figure 4-6

## Conditional Formatting Step-by-step

You can implement conditional formatting using the `IIF` statement in a few easy steps:

1. Right-click the grid cell or text box to which you want to apply the conditional formatting and select Textbox Properties.
2. On the Font page, shown in Figure 4-7, notice that each font property has an `fx` icon adjacent to it. Click the icon to launch the Expression dialog for the selected property.
3. In the Expression dialog, enter the `IIF (<Condition>, <true>, <false>)` syntax. Note that you can access the fields within your dataset(s) by clicking the fields in the category box. Clicking the Constants category displays the color palette. Double-clicking the selected field or value will add the proper syntax at the location of the cursor.
4. Click OK twice to complete and apply the conditional formatting.

Once the expression is applied to the property, you can notice that instead of the property value appearing in the Properties dialog, it has been replaced with `<><Expr>>` to indicate that an expression is in use for the property.

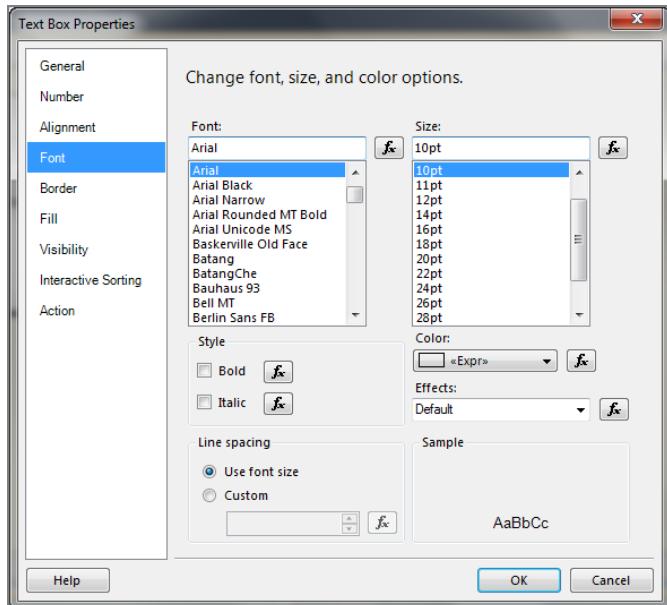


Figure 4-7

### ALTERNATING ROW COLORS IN TABLES

Implementing alternating row colors in a table (green-bar style) is an extension of conditional formatting. Instead of evaluating a data value, an expression uses the built-in `RowNumber()` function to capture the row number, and then determines whether the number's absolute value is odd or even. The expression is then applied to the `BackgroundColor` property. You can use the following expression to implement this functionality:

```
=IIF(RowNumber(Nothing) MOD 2, "PaleGreen", "White")
```

## Custom and Data-driven Labels and Titles

---

To this point, all the enhancements covered have been handled by setting properties either directly or through the use of expressions. This section discusses the skills necessary to add custom and data-driven labels and titles to your dashboard, which involves handling text, numbers, and dates in a more programmatic manner. Knowing how to combine and format these different data types enables you to build powerful, dynamic labels, and titles that are inevitably required for your scorecards, dashboards, and other analytical reports.

## Creating Custom Labels

Typically, labels and titles require some combination of text, numbers, and dates to be useful in your dashboards. As discussed previously, these labels and titles are usually implemented using the Textbox control. To make them custom, dynamic, and data-driven you must set expressions on the Textbox value.

These dynamic elements involve working with Visual Basic, so it's worth spending a few minutes to build a foundation for those who have little or no experience with Visual Basic.

When working with Textbox and Textbox values, you are ultimately dealing with strings. In Visual Basic you join strings by using the & (ampersand) sign. Take a look at the following example. Two string literals, identified by the double-quotes, are joined together to produce the new string:

```
= "Balanced Scorecards" & "-Operational Dashboards"
```

You can take this example a step further by adding in a numeric value, in this case 2013, the year this book was written. Since Visual Basic is a hybrid language, you can take advantage of the implicit type conversion and allow it to convert non-string data types to strings for you. The result of the following expression would be:

```
= "Balanced Scorecards" & "-Operational Dashboards" & 2013
```

Although the preceding two examples are the foundation for how custom labels and titles are built within Reporting Services, you can further control labels and titles by formatting numbers within the expression.

**NOTE** In addition to shortcuts to parameters, dataset fields and variables, the Reporting Services Expression designer has a number of built-in operators and functions that can be referenced when writing expressions.

The FormatNumber, FormatCurrency, and FormatPercentage functions are all built-in functions that duplicate the number formatting behavior seen when working at the property level. The following is an example that uses the FormatCurrency function:

```
= "The revenue for widgets is: " & FormatCurrency(1000, 0)
```

This expression produces the following result:

```
The revenue for widgets is $1000
```

**NOTE** All the preceding code examples use hard-coded values. In your dashboard or scorecard, these values will typically be data-driven and can be pulled from built-in fields, parameters, or even your dataset.

## Working with Dates and Times

Similarly, dates and times can also be joined to text and this is often done for titles and notations to give context to the dashboard's data. The most basic form of this expression is:

```
= "Report Generated on: " & NOW
```

This formula uses the built-in `NOW` function, which returns the current date and time. Since no formatting was applied, the result of the formula is:

```
Report Generated on: 2/11/2013 11:37:15 PM
```

The date/time could be formatted in a similar manner using the built-in `FormatDateTime()` function. Refining the previous example, the new expression uses the `ShortDateTime` format, which is indicated by the 2:

```
= "Report Generated on: " & FormatDateTime(NOW, 2)
```

**NOTE** You should always include a date/time stamp in a consistent location on every dashboard and scorecard to show the date and time of generation. This prevents the inevitable version confusion that occurs when your report is printed or screen-captured and circulated.

Beyond formatting dates within labels and titles, your dashboard might need to perform some basic date/time calculations. Although you might be familiar with some of the more common date functions found in Excel, T-SQL, and even Visual Basic, you might find that they are not very helpful when trying to solve some of the most basic challenges you will face in your dashboard. Luckily there are additional functions to help with these situations.

### *Calculating the First and Last Day of the Month*

The `DateSerial` function within Visual Basic enables easy calculation of the first and last day of the month:

```
= "Revenue Summary for: " & DateSerial(Year(NOW), Month(NOW) + 1, 1)
```

The DateSerial function takes the format of DateSerial(<YEAR>, <MONTH>, <DAY>). In this formula, notice the built-in NOW function is used to get a reference to the current date. The Year and Month functions are used to grab each date part and then a 1 is passed in for the DAY argument. Also note that the month is actually incremented by 1 since the DateSerial is offset by design. The result of this expression is:

```
Revenue Summary for 2/1/2013
```

To get the last day of the month, use the same formula, substituting a value of zero for the DAY argument:

```
= "Revenue Summary for: " & DateSerial(Year(NOW), Month(NOW) + 1, 0)
```

This expression results in the following:

```
Revenue Summary for 2/28/2013
```

### *Calculating Quarters*

Dashboards and other financial reports often report data in quarters. Summarizing data and creating custom titles and labels can be problematic, particularly when you need to find either the beginning or end of a quarter dynamically.

Determining the quarter for a given date is relatively straightforward using the following expression:

```
= "Qtr " & Int(MONTH(NOW) / 3)
```

Simple math is used to divide the month by 3, because each quarter of the year has 3 months. The Int function cleans up the messy division and rounds the result up to the next whole integer number.

Building on the previous examples, you can combine the expression used to find the first/last day of the month with the trick you just learned to find the quarter for the current month. The complete expression is shown here and in Figure 4-8:

```
= "Quarterly Revenue Summary for: " & DateSerial(Year(NOW),  
(Int((Month(NOW) - 1) / 3) * 3) + 1, 1)
```

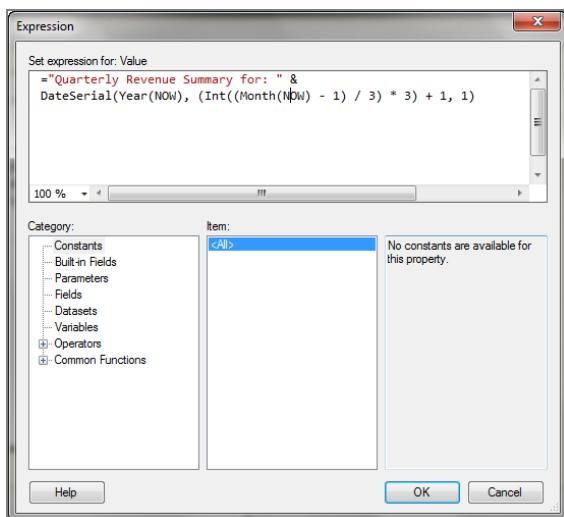


Figure 4-8

This expression results in:

Quarterly Revenue Summary for: 1/1/2013

As in the previous examples, a small tweak to the expression finds the last day of the quarter:

```
= "Quarterly Revenue Summary for: " & DateSerial(Year(NOW),  
(Int((Month(NOW) - 1) / 3) * 3) + 4, 0)
```

**NOTE** Custom, data-driven labels can result in labels and titles that are either too large or not large enough to contain the text. To prevent text boxes from taking up too much vertical space or not having enough space to display their contents, set the `CanGrow/CanShrink` properties to `true`.

## Summary

---

The focus of this chapter wasn't on metrics, analytics, or decision-making. Instead, this chapter introduced you to techniques used to enhance the organization and layout of your scorecards and dashboards. Conditional formatting is a critical part of any dashboard project, particularly when it reinforces the information and its readability. Custom labels and titles enable you to dynamically reflect the data using relevant, well-formatted descriptions and notations.

# Interactive Dashboards and Balanced Scorecards

When developing dashboards and balanced scorecards, a key component is bringing a report to life by creating an interactive experience for those that consume the report. Interactive reports make it so that what users input with the mouse is actually meaningful. This is different than the traditional view of reports, which is simply a static view of data. Report developers who keep interactivity in mind tend to use design space more effectively to help report visualization and provide a clear path for users to examine their data more closely.

Dashboards and balanced scorecards should enable users to examine data closer when they have deeper questions about the company's performance, for example. Some reporting tools are better equipped to design reports for data examination than others. Reporting Services excels at creating drillthrough reports, interactively sorting data, and filtering data with parameters to help users analyze results. These features are detailed in this chapter.

## Common Interactive Features

---

Reporting Services provides interactive reporting features that enable users to manipulate a report's appearance during run time. Many of these capabilities can provide big impact with little effort on the part of the report designer. This section describes some of the common methods for designing an interactive report using Reporting Services.

## Drillthrough

A *drillthrough* report is one that enables a user to click a link within one report that launches a second report. The second report usually has greater detail describing the information in the first report. Therefore, a report that uses Reporting Services' drillthrough capabilities in actuality is two separate reports: a summary report and a detail report, the latter of which is usually identified as the drillthrough report. The data contained in the drillthrough report is not retrieved until it is launched from the summary report.

The best way to visualize this is to imagine a dashboard like the one in Figure 5-1. This dashboard provides sales and sales quotas through a variety of visualizations like charts, indicators, and sparklines. When a user explores the sales quota scorecard, he can optionally launch a drillthrough report, which provides details on why the salesperson did or did not make his quota based on the orders processed.



**Figure 5-1**

The audience of this dashboard report is much wider, because it provides a high-level analysis of the company's performance for the executive team, while

simultaneously providing the manager who is focused on day-to-day operations the ability to dive into the details.

A drillthrough report often uses parameters to pass filter values between the summary and detail reports. A filtered detail report shows only a particular segment of data. Parameters and how they are used for interactive reporting are discussed later in this chapter, in the section titled, “Parameterization.”

## Expand / Collapse Groups

Another feature that you may find especially useful is the ability to expand and collapse groups that are part of your report. This can only be done within the context of a table or matrix and enables developers to create reports that don’t overwhelm users with too much data all at once. Expanding and collapsing groups are easily identified by the plus or minus sign next to the groups where the feature is turned on, which is visible in Figure 5-2. Adding this element can make a world of difference for the users’ experience because the initial view of the report is much more usable. For instance, imagine you’re working on this same report but without the ability to expand or collapse the groups. You can see in Figure 5-3 that this can be rather overwhelming for a user.

Subcategory	Product Name	Order Quantity
⊕ Bike Racks		328
⊕ Bike Stands		249
⊕ Bottles and Cages		7981
⊕ Caps		2190
⊕ Cleaners		908
⊕ Fenders		2121
⊕ Gloves		1430
⊕ Helmets		6440
⊕ Hydration Packs		733
⊕ Jerseys		3332
⊕ Mountain Bikes		4970
⊕ Road Bikes		8068
⊕ Shorts		1019
⊕ Socks		568
⊕ Tires and Tubes		17332
⊕ Touring Bikes		2167
⊕ Vests		562

Figure 5-2

Subcategory	Product Name	Order Quantity
Bike Racks	Hitch Rack - 4-Bike	328
Bike Stands	All-Purpose Bike Stand	249
Bottles and Cages	Mountain Bottle Cage	2025
	Road Bottle Cage	1712
	Water Bottle - 30 oz.	4244
Caps	AWC Logo Cap	2190
Cleaners	Bike Wash - Dissolver	908
Fenders	Fender Set - Mountain	2121
Gloves	Half-Finger Gloves, L	443
	Half-Finger Gloves, M	499
	Half-Finger Gloves, S	488
Helmets	Sport-100 Helmet, Black	2085
	Sport-100 Helmet, Blue	2125
	Sport-100 Helmet, Red	2230
Hydration Packs	Hydration Pack - 70 oz.	733
Jerseys	Long-Sleeve Logo Jersey, L	452
	Long-Sleeve Logo Jersey, M	442
	Long-Sleeve Logo Jersey, S	429

**Figure 5-3**

To utilize and modify expanding and collapsing groups for your own report, complete the following steps:

1. In the Row Groups pane of a report right-click the group that you want to collapse and select Group Properties.
2. Select the Visibility page and change the When the report is initially run: property to Hide.
3. Check the Display can be toggled by this report item, and select the group that you would like to collapse into; then click OK. Figure 5-4 shows an example of this step completed.

You can use these same techniques for any reports that use table or matrix reports with groups.

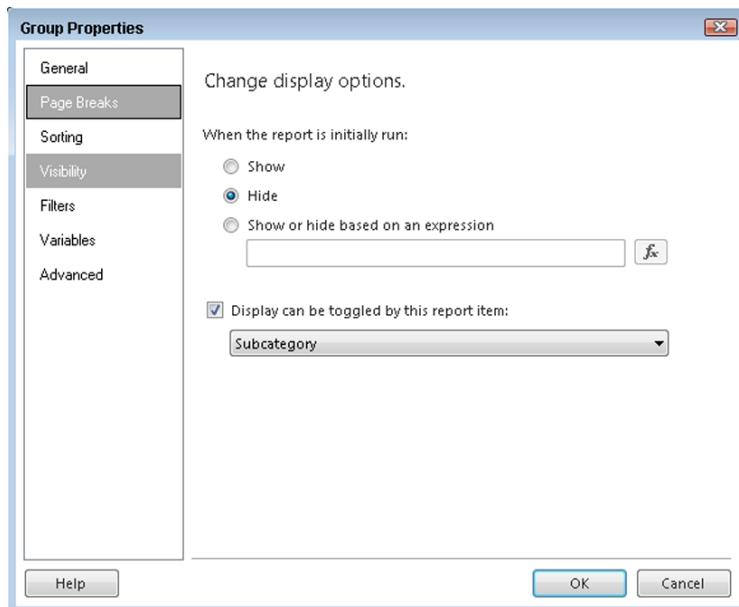


Figure 5-4

## Interactive Sorting

Another common interactive feature of Reporting Services is *interactive sorting*. This feature enables users to change the sort order of the data presented to them with a single click. This tool can be applied to one or multiple columns within a table or matrix in Reporting Services.

To the user, this appears as an up and down arrow next to a column header, which indicates that the column is capable of sorting interactively. Figure 5-5 shows an example next to the Sales column header.

Year	Sales
2005	\$506,570
2006	\$1,176,251
2007	\$1,227,895

Figure 5-5

To add interactive sorting to a column header in a table or matrix, follow these steps:

1. Right-click a column header in your table and select Text Box Properties.
2. Click the Interactive Sorting page.

3. Check the property Enable interactive sorting on this text box.
4. Choose what you want to sort. You can select to sort by the Detail rows or Groups. The Detail rows sorting is the lowest granularity of the dataset. Groups are defined by a developer and enable a table or matrix to aggregate data. It is similar to a GROUP BY in T-SQL.
5. Select which field you want to sort the data by changing the Sort by property (shown in Figure 5-6), then click OK.

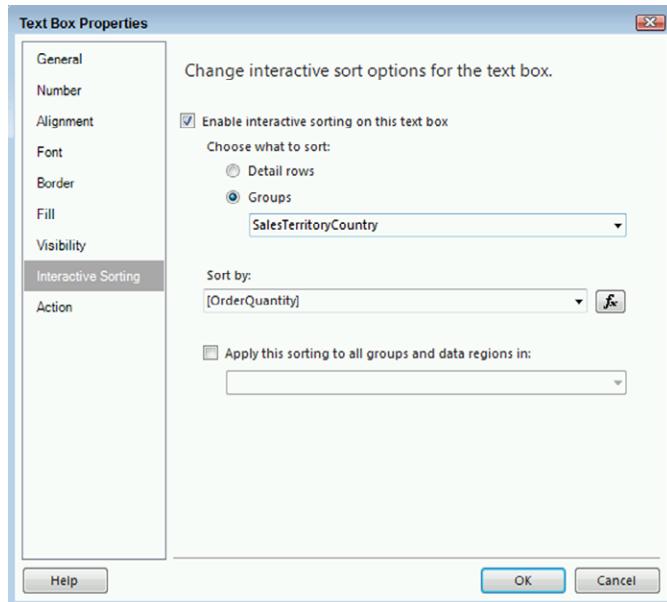


Figure 5-6

Next time the report is previewed, the up and down arrow indicating an interactive sort will be enabled. If the user clicks it one time, the data is sorted in ascending order. A second click causes the data to be in sorted descending order.

## Bookmarks

Making navigation simple and easy to follow in Reporting Services is an often overlooked part of the design process. *Bookmarks* are a great way for the report developer to guide users down a desired path.

To create a bookmark you must first find a location on your report on which you would like to focus the user's attention. After marking that location as a bookmark you can create an Action on any other spot on the report, linking the two positions together. The user simply clicks this link to "jump" to the bookmarked location in the report.

For example, the report in Figure 5-7 has several charts and graphs that tell a story about the business. Unfortunately, such a large amount of design space is taken up that the user has to scroll to find some of the important information at the bottom of the report, such as the bar chart.



Figure 5-7

In a situation like this, you can help the user by providing a bookmark that takes her directly to the part of the report she cares most about.

To add a bookmark location on a report, follow these steps:

1. In Design view, select the report object that you want to bookmark. For this example, choose the bar chart.
2. Press F4 to launch the Properties pane.
3. In the Properties pane, find the **Bookmark** property in the Other section and type in a name that describes your bookmark, for example, **CategoriesColumnChart**.
4. Now, decide where you want the bookmark link to be located on your report. This could be as simple as a text box at the top of the report directing users to click it. Right-click the text box and select Text Box Properties, then select the Action page.

5. Select Go to bookmark and type the same name you created in Step 3 when you initially created the bookmark. The text here is case sensitive so ensure it is exactly how you typed it in Step 3. Figure 5-8 shows this step completed. Click OK once complete.

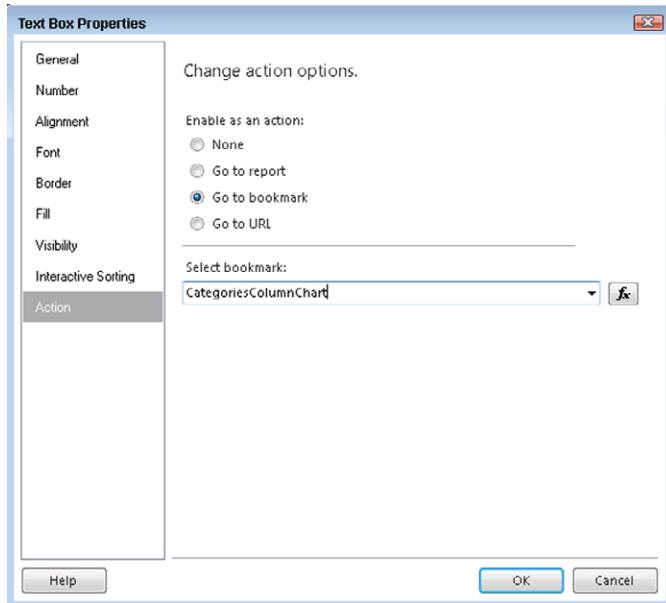


Figure 5-8

When the report is previewed, the bookmark link enables users to quickly navigate to all areas of the most important parts of the report with ease.

Bookmarks, interactive sorting, and drillthrough reports are great methods for making the appearance of a report interactive, but these changes are purely for aesthetics and have no impact on the data actually displayed. In the next section you learn about providing interactivity with the report data that is displayed through a method called parameterization.

## Parameterization

Regardless of the reporting tool, using *parameters* is one of the best ways to improve user interaction with your report. In fact, many reporting tools in the past several decades would *only* provide parameters or variables as a way to interact with reports.

Through parameters, users can give direct inputs into a report to affect how a report is rendered. This enables users to impact tasks like data retrieval, object visibility, connecting reports together, report formatting, and many other report skills.

This section focuses on how Reporting Services interacts with parameters and also details advanced development skills, like cascading parameters.

## Introduction to Reporting Services Parameters

Parameters are a vital part of developing reports with Reporting Services. Because parameters are such a commonly used component, they have been made fairly simple to create.

To add a basic parameter to a report, follow these steps:

1. In the Design view, open the Report Data pane.
2. Right-click the Parameters folder and select Add Parameter.
3. In the Name property, provide a name that will be clear for other developers to understand. For example, if your parameter was part of a date range you may call it `StartDate`. This name will not be seen by the users and cannot have spaces.
4. The Prompt property is what the users see when they are interacting with the report. Ensure this field has a value that makes it clear for users to know how to interact with it, as shown in Figure 5-9.

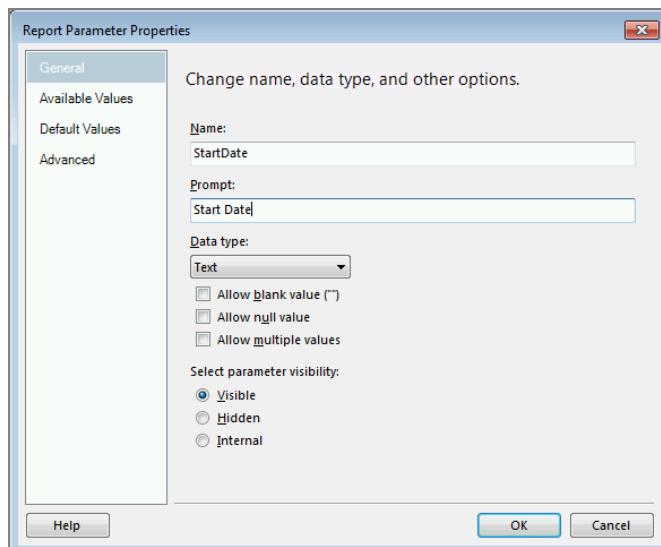


Figure 5-9

5. Other options on the General page include the Data type of the parameter, what kind of values the parameter can accept (blank, null, or multiple), and whether the parameter is visible to users. A developer may hide a

parameter from users so it can be used internally in a report. The other sections of Report Parameter Properties are discussed later in this section. Click OK to create the parameter.

There are several reasons why you may need to use parameters, and the following sections detail how, under different scenarios, parameters can be used with Reporting Services.

### ***Data Retrieval***

Manipulating data retrieval is the most common reason a report developer designs a report with parameters. Using parameters in this manner enables users to directly manipulate what data is returned during report execution. There are numerous business cases for requiring a report to use parameters in the data retrieval process. For example, imagine a source database with 25 years of shipping data. If a user ran a report on this data without limiting the date range in some way, the report would likely time out before returning results.

When using parameters for data retrieval, you identify parameters in the query of the dataset differently depending on the data source type. For example, if the query is against a SQL Server data source, then a parameter is identified with an @ sign. After the @ sign you must provide a name to the parameter. If you want to use parameters to filter a date range, then you can create two parameters: one called @StartDate and a second called @EndDate, as shown in Figure 5-10. Reporting Services automatically interprets these references in your query and turns them into parameters in the report. Other data source types have different ways of identifying a parameter in Reporting Services. For example, if the data source is Oracle, then you must use a colon in front of the parameter name instead.

In Figure 5-10 you see a query that is being modified directly in the report design. While this method is okay for developing quick reports, ideally all queries would be inside of stored procedures. *Stored procedures* are database queries that are saved as objects inside SQL Server databases. Storing queries that are used for reports in SQL Server stored procedures is commonly thought of as a best practice. Reporting Services can easily consume stored procedures through the Dataset Properties dialog box. After the stored procedure is created, you can simply change the Query type to Stored Procedure and select the name of the stored procedure from the dropdown box below, as shown in Figure 5-11. If the code inside the selected stored procedure has parameters, then they will automatically be created as Reporting Services parameters in the designer. One of the main reasons using stored procedures is considered a best practice is because if a developer needs to make a minor change to the source query, like altering the WHERE clause for example, then there is no need to open the report. He can simply open the stored procedure.

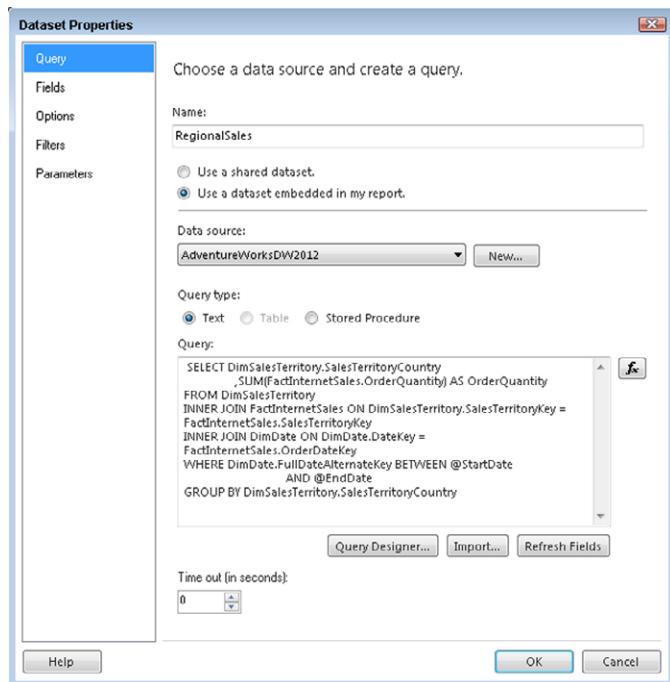


Figure 5-10

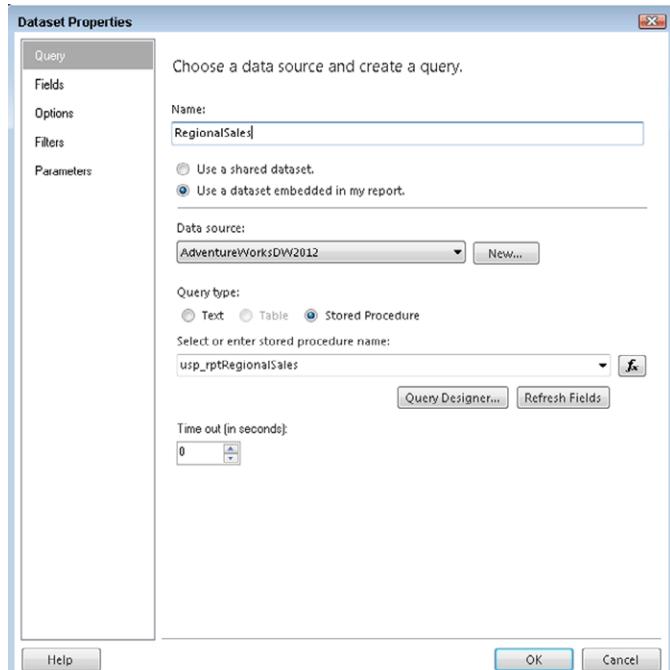


Figure 5-11

### *Object Visibility*

Occasionally, when designing a report, you might find it helpful if the user had the option to hide less significant elements of the report. Object visibility is often controlled by the report developer and can be data driven. This can be accomplished with expressions that control properties based on the data that is returned from the datasets.

To tie a parameter to the visibility of an object, follow these steps:

1. In the Design view, open the Report Data pane.
2. Right-click the Parameters folder and select Add Parameter.
3. Name the parameter so that other developers can clearly understand its purpose. In this example the parameter will be named `HideChart`.
4. Change the `Prompt` so users know what the parameter does. In this example, use `Hide chart`.
5. Change the `Data type` property to `Boolean`. This gives users the option to select `True` or `False` when they execute the report. Click OK to accept the parameter properties.
6. Right-click and select Properties on the object for which you want to control the visibility. In this example, right-click a chart and select Chart Properties.
7. Navigate to the Visibility page and change the `When the report is initially run` property to `Show` or `hide` based on an expression.
8. Click the `fx` button next to the text box and enter the expression `=IIF(Parameters!HideChart.Value=True, True, False)`, then click OK twice. This expression checks the value of the `HideChart` parameter, and if it is set to `True` then the object is hidden. If `False`, then it will remain visible.
9. Preview the report and you will see that the parameter provides users the option to turn on or off the visibility of an object, as shown in Figure 5-12.

Using parameters to control object visibility can be a powerful option, but you should use it on a limited basis because it can be taken too far. Some users might want you to design one report to solve all their problems using parameters to hide things they don't want during different executions of the report. While it may seem like a good idea, the long term management of such a report would be a nightmare when changes are inevitably required. You would not only have to make your new changes but also be aware of all the implemented hidden features.

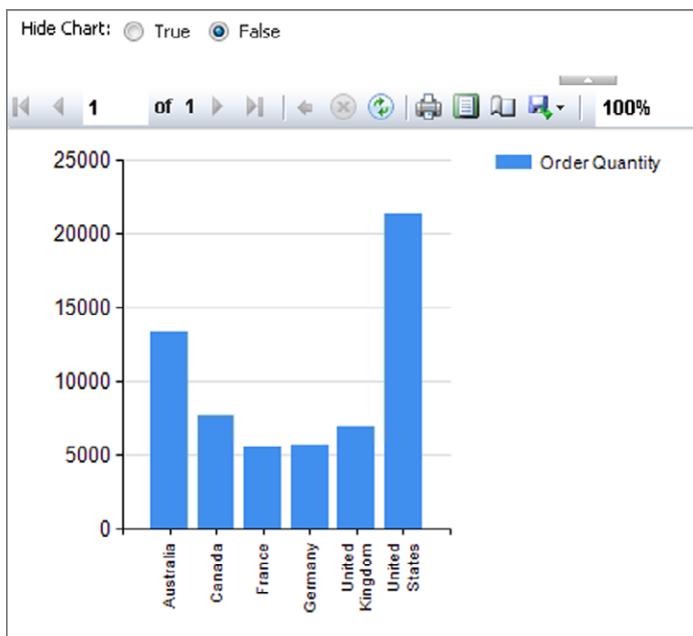


Figure 5-12

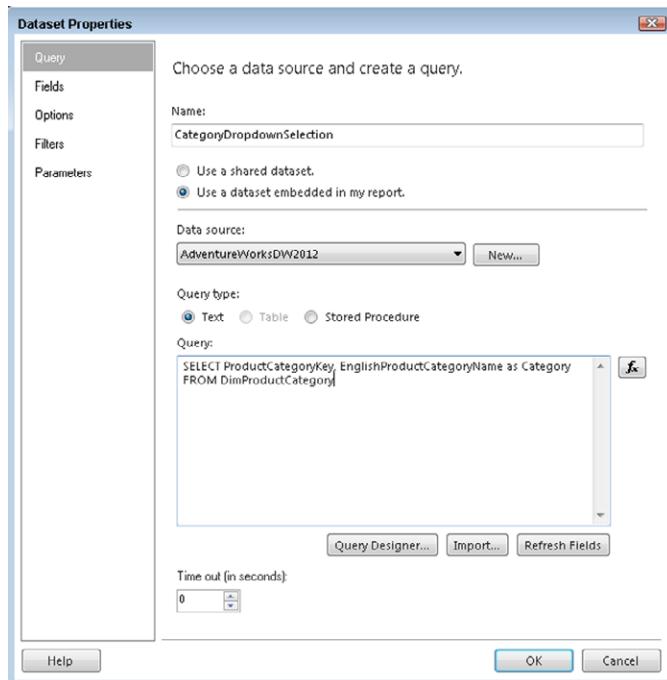
## Data-driven Parameters

So far you have learned how to create a basic parameter that gives users the ability to interact with reports by typing text. To take your parameters up a notch, you can turn those text-based parameters, where users must type their selection, into a dropdown selection of options for the user. This concept is known as a *data-driven parameter* and can be very powerful in guiding users through parameter selections.

To create the effect of a dropdown selection from a parameter, you must have a dataset whose job is to populate the dropdown list with appropriate values. When the dataset is created, you can then change the Report Parameter Properties to use the available values from this dataset. Within the Available Values section of Report Parameter Properties, you must select the appropriate dataset to populate the dropdown box and also map the Value and Label fields. These are important properties because you may want to show the users one field that is commonly known in the business (Label field), but the query is expecting something like a key value that makes for faster query performance (Value field).

To make a parameter data-driven, follow these steps:

1. In the Design view, open the Report Data pane.
2. Right-click Datasets, and select Add Dataset.
3. Name the dataset to identify it as a dropdown selection. For this example, use **CategoryDropdownSelection**.
4. Use an embedded dataset and select your data source.
5. Provide a query that will give you a distinct list of values to display in the dropdown, then click OK. Figure 5-13 shows this completed.



**Figure 5-13**

6. In the Report Data pane, right-click the parameter you want to change to a data-driven selection and select Parameter Properties.
7. Go to the Available Values page and change it to Get values from a query.
8. In the Dataset property, choose the dataset you completed in step 5.
9. For the Value field property, select the column that the query will use for filtering data.
10. In the Label field, select the column you want displayed to the user when interacting with the parameter, then click OK. Figure 5-14 shows this step.

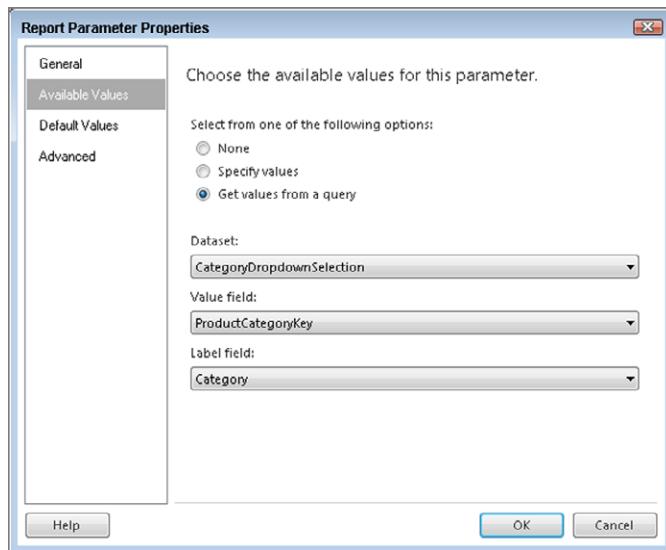


Figure 5-14

11. Now, preview the report and you see the parameter you created is now data-driven with a dropdown selection for the users to choose from. Figure 5-15 shows the user's interaction with the parameter.

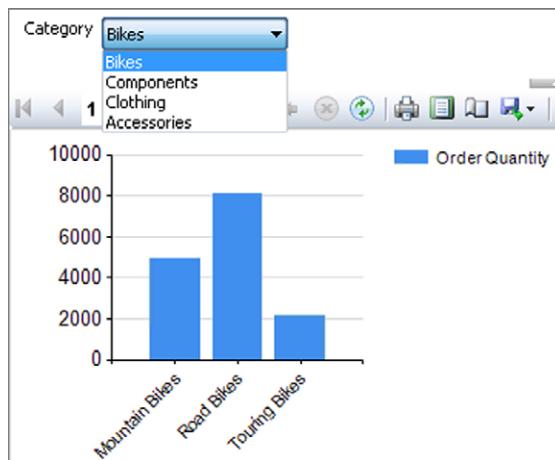


Figure 5-15

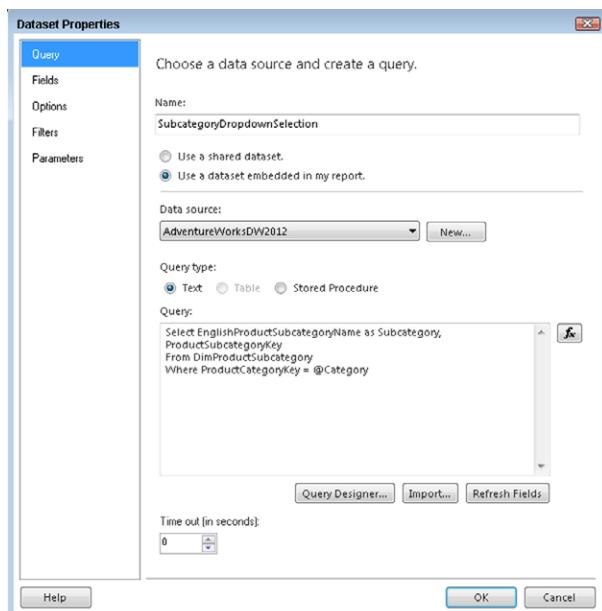
Using the data-driven parameter technique can save users a lot of the headaches that come with text parameters that require user interaction. Users no longer have to worry about things like simple spelling mistakes causing frustration to a report's design.

## Working with Cascading Parameters

Interacting with several parameters on a single report can be challenging. Think about it from a user's perspective: You launch a report with six parameters. How do you know which ones to select first, and how does your selection in one parameter affect the others?

This is where cascading parameters can be essential in guiding users and preventing them from making mistakes while inputting parameter values. *Cascading parameters* are multiple parameters that have a relationship and can work together to filter down values in each other. For example, if a report has parameters called `Country`, `State`, and `City`, you shouldn't expect, after selecting United States from the `Country` parameter, to see more than 50 values in the `State` parameter. If you start to see Canadian provinces then something has gone wrong during the report development. Cascading parameters control the process of stepping through multiple related parameters.

The design of a report using cascading parameters is similar to the data-driven parameter example. The only difference is there is more than one parameter, and one dropdown selection dataset feeds the next dropdown selection dataset. Suppose you are creating a report that uses product category, product subcategory, and product fields, and you want the subcategory parameter to only show values based on the selected category. To achieve the cascading effect, you simply change the subcategory dataset to accept the value from the category parameter, as shown in Figure 5-16. This prevents users from selecting one category and getting a list of subcategories from another category.



**Figure 5-16**

The user sees a seamless flow between each parameter selection and is only able to input values into these parameters in the correct order. The subcategory parameter is disabled until a value is selected for the category parameter. Figure 5-17 shows the user experience when using cascading parameters.

The screenshot shows a report interface with two cascading parameters at the top: 'Category' set to 'Bikes' and 'Subcategory' set to 'Mountain Bikes'. Below the parameters is a toolbar with navigation icons (back, forward, search, etc.) and a zoom control set to '100%'. The main content area displays a table titled 'Product' with columns 'Product' and 'Order Quantity'. The data rows are:

Product	Order Quantity
Mountain-100 Black, 38	49
Mountain-100 Black, 42	45
Mountain-100 Black, 44	60
Mountain-100 Black, 48	57

Figure 5-17

## Step-by-step Drillthrough Report

Now that you have an understanding of both drillthrough reports and parameters, you are ready to design an end-to-end drillthrough report solution. Dashboards and balanced scorecards that are designed in Reporting Services commonly use this combination of drillthrough reports with parameters.

Remember that drillthrough reports actually require the design of two reports: a summary report and a detail report. In this example, you first design the detail report and then connect the summary report that uses chart visualization. To design a drillthrough report, follow these steps:

1. Open SQL Server Data Tools and create a new Reporting Services project.
2. In the Solution Explorer, right-click Shared Data Sources and create a connection to the AdventureWorksDW2012 database. This database can be found at [www.codeplex.com](http://www.codeplex.com).
3. Right-click the Reports folder and select Add > New Item.
4. Select Report and then name the report `Drillthrough Detail Report` then click Add.
5. With the blank design surface now open, navigate to the Report Data pane. Right-click Data Sources, and select Add Data Source.
6. Select Use shared data source reference, name the data source `AdventureWorksDW2012` and then click OK.
7. Right-click the Datasets folder and select Add Dataset.

8. Name the dataset `ProductSales` and select the option Use a dataset embedded in my report.
9. Change the Data Source for the dataset to use `AdventureWorksDW2012` and click Query Designer.
10. Create a new dataset using the Drillthrough Detail Report Query found in the book's download page at [www.wiley.com/go/performancedashboards](http://www.wiley.com/go/performancedashboards). Once the query is complete, click OK to complete the dataset.
11. From the Toolbox pane, drag a Matrix onto the design surface of the report.
12. Place the field `EnglishProductName` in the Row Group, rename the column header `Products`, and then bold the text. Also, resize the column so it can fit a lengthier product name.
13. Make the field `CalendarYear` a column group and bold the text.
14. Place the `OrderQuantity` field in the Data section of the matrix so it will be automatically aggregated.
15. Optionally, you can add a Page Header by right-clicking outside the design surface of the report and selecting Add Page Header. Drag a text box from the toolbox into the header and type the text: **Product Sales by Year**.

This completes the drillthrough detail report. When you preview your design and type in the parameter value of **20** it should look similar to Figure 5-18. Now, you are ready to create the summary report.

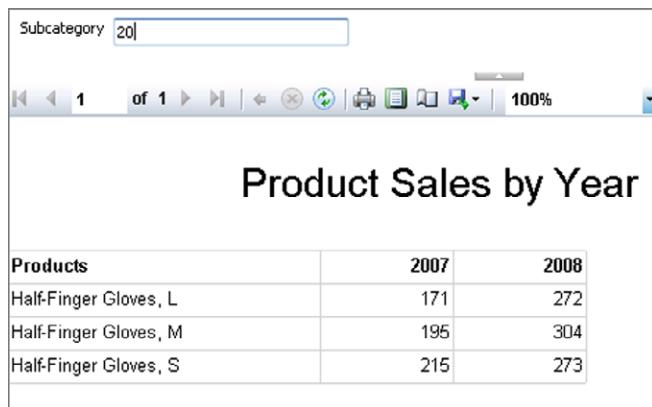


Figure 5-18

1. Right-click the Reports folder in the Solution Explorer and select Add > New Item.
2. Select Report, name the report `Drillthrough Summary Report`, and then click Add.

3. With the blank design surface now open, navigate to the Report Data pane, right-click Data Sources, and select Add Data Source.
4. Select Use shared data source reference, name the data source AdventureWorksDW2012, and then click OK.
5. Right-click the Datasets folder and select Add Dataset.
6. Name the dataset SubcategorySales and select the option Use a dataset embedded in my report.
7. Change the Data Source for the dataset to AdventureWorksDW2012 and click Query Designer.
8. Create a new dataset called SubcategorySales using the Drillthrough Summary Report Query found on the book's download page at [www.wiley.com/go/performedashboards](http://www.wiley.com/go/performedashboards). Once the query is complete, click OK to complete the dataset.
9. Create a second dataset for populating a dropdown selection on the Category parameter called CategoryDropdownSelection that returns back ProductCategoryKey and EnglishProductName from DimProductCategory.
10. In the Parameters folder, right-click the Category parameter and select Parameter Properties.
11. On the Available Values page, select Get values from a query and change the Dataset to CategoryDropdownSelection. Then change the Value field to ProductCategoryKey and the Label field to EnglishProductName, as shown in Figure 5-19.

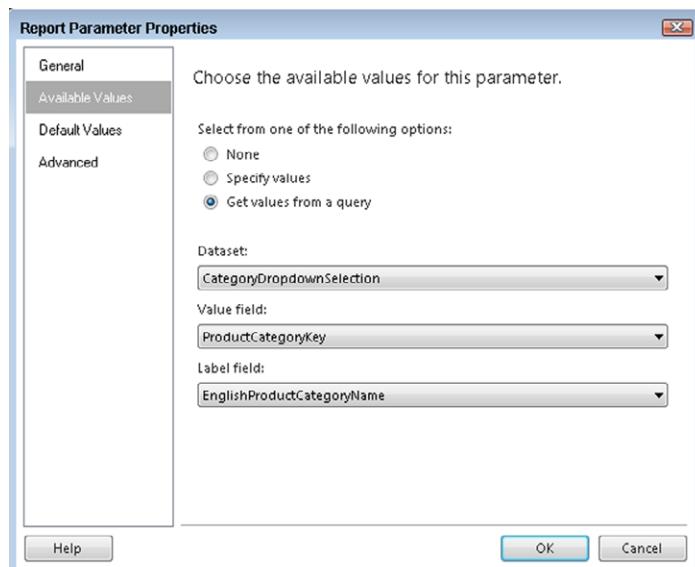


Figure 5-19

12. Go to the Default Values page and select Specify values. Click Add and then type 3 for the value.

Now you are ready to create your chart visualization. Use the following steps to add a chart and apply data to it:

1. From the toolbox, drag the chart to the design surface and select the Column chart when prompted.
2. Change the chart title to Subcategory Sales and then delete both Axis titles.
3. From the SubcategorySales dataset bring the EnglishProductSubcategoryName field to the Category Group of the chart. Then from the same dataset bring the OrderQuantity field to the Values section of the chart. Figure 5-20 shows this step.

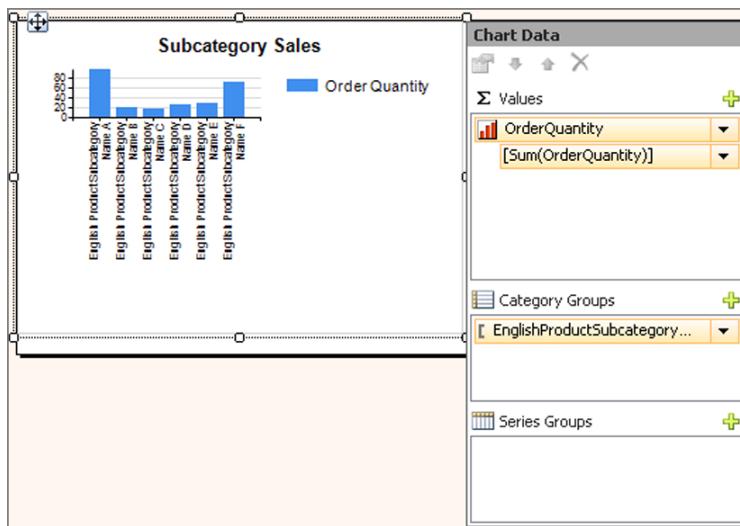


Figure 5-20

The last part of this report is to create the drillthrough connection between the summary and detail reports. To do so follow these steps:

1. Right-click one of the columns in the column chart and select Series Properties.
2. Go to the Action page and select Go to report. Select Drillthrough Detail Report under the Specify a report property.
3. Click Add to include parameters that need to be passed to the detail report. Select Subcategory from the Name dropdown box

and [ProductSubcategoryKey] for the Value. This takes the field ProductSubcategoryKey from the summary report and passes the value into the detail report parameter called Subcategory. Figure 5-21 shows this configuration completed.

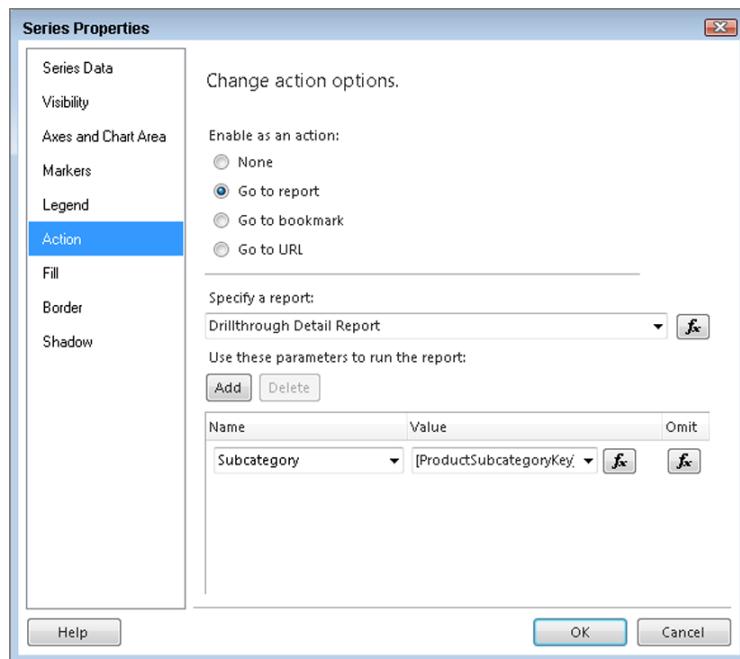


Figure 5-21

The report is now complete. You can preview the design and click any of the columns in the chart to see the drillthrough detail report launched and the parameter value passed. Figure 5-22 shows the completed summary report.

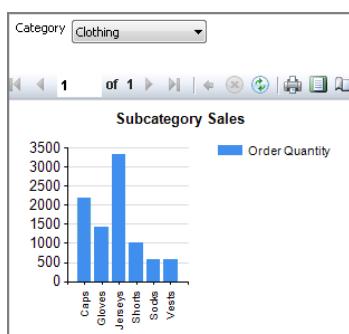


Figure 5-22

You've now designed a set of reports that utilizes many of the features discussed in this chapter. If you imagine this design on a much larger scale, you should get the idea of how to design a dashboard with several charts, gauges and indicators. Each of these visualizations can utilize the same technique described here for creating an interactive drillthrough report.

## Summary

There are several methods for creating interactive dashboards and balanced scorecards using Reporting Services. You can use both report visualizations and parameters to make a report capable of being interactive. You can also use interactive sorting and bookmarks as methods for interactive visualizations and data-driven and cascading parameters as methods for parameterizing your reports. Implementing these features brings your reports to life and changes them from boring static reports to interactive reports that provide a more enjoyable experience for your users.

# Visual Dashboards and Balanced Scorecards

The visual components of both dashboards and balanced scorecards are typically the first that come to mind when discussing analytic reports. From graphs and charts to gauges and status indicators, the visual components are the most important when it comes to easily convening information and helping your target audience understand and interpret the data presented.

This chapter introduces you to the various charts, graphs, gauges, and visual indicators available within Microsoft SQL Server Reporting Services. It focuses on choosing specific data visualizations, implementing the most common visualizations, and looking at advanced customizations for each.

## General Guidelines for Data Visualizations

---

The driving force behind the contemporary data visualization movement is a combination of the massive scale of data available and its increasing degree of interconnectedness. Although these issues are not always relevant when discussing dashboards and balanced scorecards, the lessons learned are certainly applicable when working with information rich analytical reports. It's worth noting that there are no hard and fast rules when it comes to data visualizations, only generalized guidelines that are often seen as part art and part science. This section focuses on defining a few of these high-level guidelines that are applicable when implementing analytical reports.

## Know your Audience

The types of visualizations used depend entirely on your audience. Typically executives are more interested in metrics or key performance indicators, which are best implemented using bullet charts, gauges, or status indicators. An operational manager, who is monitoring day-to-day activities, typically prefers charts or graphs. Knowing your audience helps you select the most appropriate visualization.

## Select the Right Medium

The data presented should determine your choice of medium, and the visualization you select should help rather than hinder your end-user interpretation of the data. For example, if you are presenting data for comparison, bar charts are the most appropriate medium. Likewise, if you are attempting to illustrate a trend, a line chart works best.

Beware of traps, however. Microsoft SQL Server Reporting Services is a robust enterprise reporting tool with a vast array of visual components available out of the box. Most of the visualizations, however, are typically not used in dashboards. There is often a tendency to introduce variety for the sake of variety, but this is counterproductive and will ultimately obscure the data you are trying to present. If the data dictates that all charts are bars, columns, or lines, then use the same visualization throughout.

## Kill Cute

You have inevitably seen them—over the top analytic reports that are as much about decoration as they are about data. In his book, *Information Dashboard Design* (O'Reilly Media, 2006), Stephen Few introduces the concept of reducing non-data pixels while enhancing data pixels. This is particularly applicable when working with the built-in visual tools in Reporting Services.

The easiest way to avoid getting carried away is to try and identify both the non-data and data pixels as you design your visualizations. Non-data pixels can be anything from graphics to 3D effects to borders, background colors, and gradients and even includes gridlines within charts.

For instance, 3D charts obscure the data being presented. Identifying trends in a 3D line chart is difficult, if not impossible. As you can see in Figure 6-1, a simple line chart is much more effective and easier for the user to understand and interpret.

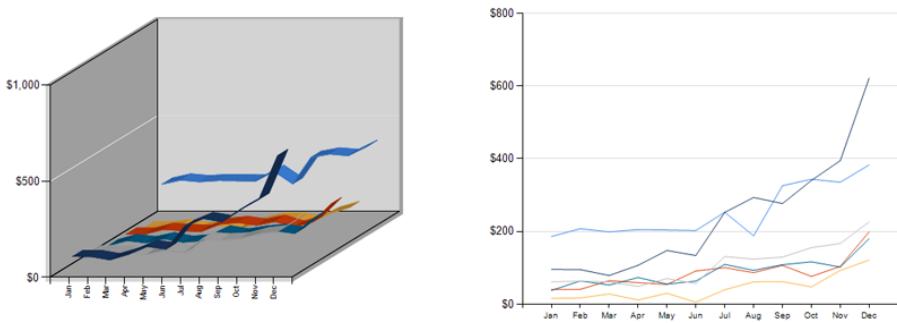


Figure 6-1

When it's not possible to remove non-data pixels, the focus should be on de-emphasizing them and making sure they are consistent across your report so they do not stand out above the data.

On the other hand, managing data pixels involves reducing unnecessary data, typically through summarization and by highlighting the remaining and most important data pixels.

## Use Colors Carefully

Colors are an important tool when designing data visualizations for analytical reports. The hue, saturation, and brightness of color can be used to encode data as well as differentiate it in the event of an anomaly or exception.

Care should be taken when selecting color palettes. Typically, you want to avoid palettes that consist primarily of intense or bright colors, because they can cause visual fatigue over time and make highlighting important data more difficult. Instead, choose colors that are less saturated and more muted, such as those typically found in nature, making sure that whatever scheme you choose allows for easy differentiation between hues of colors.

No discussion of color would be complete without addressing the impact of your color choices on those with color blindness. It is estimated that up to 10 percent of the male population and 1 percent of the female population are affected by color blindness. Some argue that communicating meaningful information only by the distinction of color should be absolutely avoided. Although shapes and patterns can be substituted for color, you can also vary the saturation of color rather than vary the hue.

## Working with Charts and Graphs

Charts and graphs are the most common visualization and are found across a wide variety of mediums, usually in the form of a bar, line, pie, or histogram chart. These charts are used to graphically represent both the data and the relationships that exist amongst the data.

SQL Server Reporting Services offers robust and native support for a wide variety of charts and graphs, some of which are more important in dashboards and balanced scorecards than others. Out of the box, there are eight categories of charts (shown in Table 6-1) that are available for you to use:

**Table 6-1:** SQL Server Reporting Services Chart Types

CHART TYPE	DESCRIPTION
Column	More commonly known as a “bar chart,” the column chart in Reporting Services is a vertically oriented bar chart implementation. Column charts excel at illustrating how data changes over time and comparing data items. Reporting Services offers several variations to the standard 2D bar chart in the forms of stacked, 3D, and cylinder charts.
Line	As expected, Reporting Services supports the most common form of chart, the line chart. Line charts are used to visualize trends. In Reporting Services, these charts range from a simple line chart to ones that are smoothed or stepped and can include markers to highlight each data point.
Shape	Shape charts are commonly used to illustrate proportional comparisons of data in terms of the dataset. The most well-known shape chart is the pie. Multiple options for shapes exist in Reporting Services including the pie, doughnut, funnel, and pyramid shapes. Each shape includes a 3D option.
Bar	The bar chart in SQL Server Reporting Services is the horizontal-orientated chart implementation. The usage and options for customization of these charts is the same as the column chart.
Area	The area chart is based on the line chart and is best used to graphically illustrate quantitative comparisons among sets of data. Reporting Services supports basic area charts along with options for charts that are smoothed, stacked, and 3D.
Range	Range charts vary from simple charts that have a shaded area between high and low values, to those that use columns or bars. This type also includes stock, candlestick, error, and boxplots, which are common in some industries. Range charts, as their name implies, excel at visualizing data ranges over time.

CHART TYPE	DESCRIPTION
Scatter	One of the more complex chart types, a scatter chart is a collection of points plotted on an x- and y-axis with each point representing a single variable. These charts are useful for graphically illustrating a correlation between variables. Out of the box, Reporting Services supports a simple scatter plot and a bubble plot, which uses the size of the bubble to illustrate the difference between two values.
Polar	Polar charts plot data points on a 360-degree circle, using the distance from a center point or pole to illustrate each data value. This chart type excels at quantitative comparisons and is typically used when a natural cycle (such as a daily or annual) exists. Reporting Services offers the polar chart, radar chart, and 3D radar chart.

## Getting Started with Charts

In Reporting Services, all eight of the chart types can be implemented using the Chart component found in the Report Item toolbox. When you drag the Chart component to the report design surface, the Select Chart Type dialog (shown in Figure 6-2) is presented.

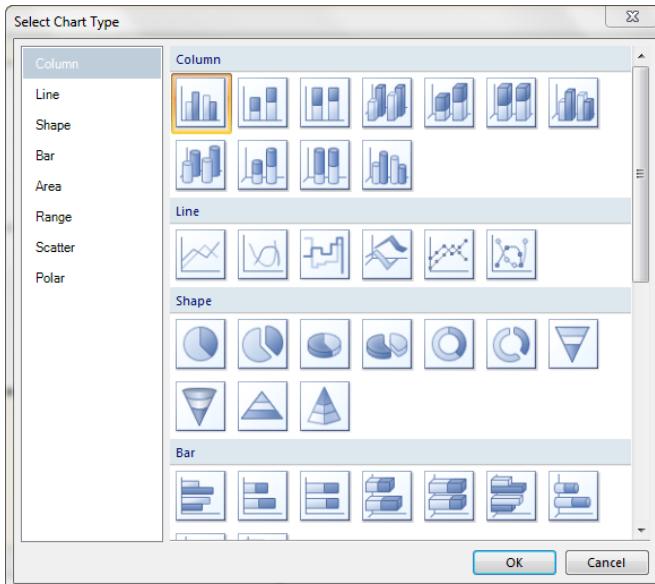


Figure 6-2

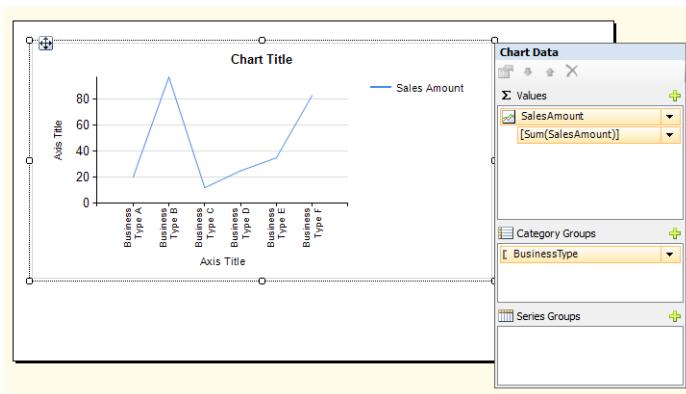
From this dialog you can select any of the built-in chart types. It's also worth noting that at any time you can switch between chart types. Simply right-click

the chart on your design surface and choose the Chart Type option to reinitiate the Select Chart Type dialog.

All of the chart types, except shape charts, share the same primary parts: the *chart data region*, the *chart area*, the *horizontal and vertical axes*, the *chart and axes titles*, and the *chart legend*.

### Chart Data Region

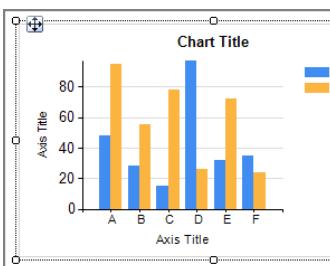
The chart data region is organized into three main parts: the *chart series*, the *category groups*, and the *series groups* (see Figure 6-3). The chart series are the numeric data or values encoded by the chart. Category groups are used for grouping data, and series groups are optional and can be used to add additional dimensions to the chart.



**Figure 6-3**

### Chart Area

The chart area is the portion of the chart or graph that the chart is actually rendered within. The background fill, gridlines, and 3D options (if they are available) are all found on the property pages for this part (see Figure 6-4).



**Figure 6-4**

### *Horizontal and Vertical Axes*

Both the horizontal and vertical axes of the chart are highly configurable. Axis options, which vary depending on the chart type, labels, label fonts, number format, major and minor tick marks, and the axis line itself, can all be tweaked to fit within the dashboard or other analytical report you are designing.

### *Chart and Axis Titles*

The title components of the chart come with the configuration options you would typically expect. Font, font size, font effects, and color are common across all titles. The chart title has additional options that enable the title to be rotated and docked, and use of borders and gradients.

### *Legend*

Aside from the options such as font, fill, and border you would expect, the legend can be rotated and docked, and layout of the items within the legend can be switched between variations of row, column, and table arrangements.

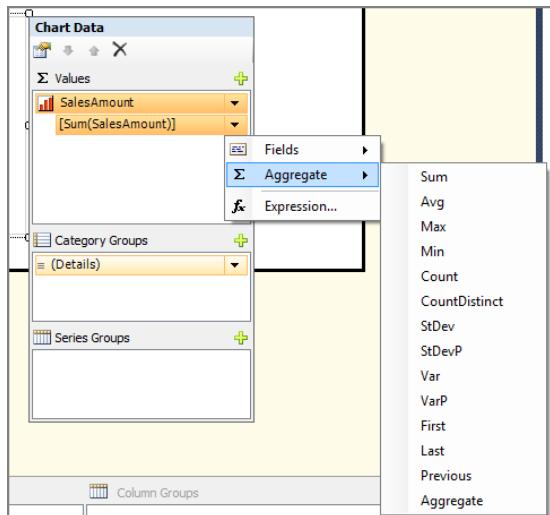
### *Implementing Graphs and Charts*

Take a look at an example for implementing a chart or graph in Reporting Services. For this example you implement a bar chart based on Reseller Sales for the Adventure Works data warehouse. Technically speaking, Reporting Services divides bar graphs into two categories: bar or column, depending on the orientation of the bar. For this discussion, although you will be working with a column chart, both the bar and column charts are referred to as bar graphs.

The source of data for the chart is as follows:

```
SELECT
    d.CalendarYear AS [Year],
    d.MonthNumberOfYear AS [Month],
    LEFT(d.EnglishMonthName, 3) AS [MonthName],
    r.BusinessType,
    SUM(s.SalesAmount) AS SalesAmount
FROM FactResellerSales s
JOIN DimReseller r ON (s.ResellerKey = r.ResellerKey)
JOIN DimDate d ON (s.OrderDateKey = d.DateKey)
GROUP BY
    d.CalendarYear,
    d.MonthNumberOfYear,
    d.EnglishMonthName,
    r.BusinessType
```

1. Using the skills you learned in Chapter 3, “Building Your First Dashboard and Balanced Scorecard,” set up your report, add a Chart component from the toolbox and select the Column chart type.
2. Resize the chart by dragging the boundaries so that you have ample space to display the visualization. Keep in mind that you can always resize and adjust the chart later if you realize you need more or less space.
3. To link the chart to the dataset containing the previous query, find the `DataSetName` property and select the dataset in the Properties window (press **F4** if the Properties window is not open).
4. With the Chart Data region visible, drag the `SalesAmount` column from the dataset to the Values, and the `MonthName` column to the Category Groups. Since the dataset includes multiple years, change the Aggregation type for the `SalesAmount` from Sum to Average (Avg). Figure 6-5 highlights the aggregate pop-up in the Chart Data window.



**Figure 6-5**

5. If you preview the chart, you will notice that the month names along the horizontal axis are out of order. To configure the sorting of the x-axis items: switch back to design view, click the down arrow next to `MonthName`

Category Group in the Chart Data and select Category Group Properties, as shown in Figure 6-6. The Sorting tab enables configuration of the sort order and, in this case, you can fix the order by choosing the Month column.

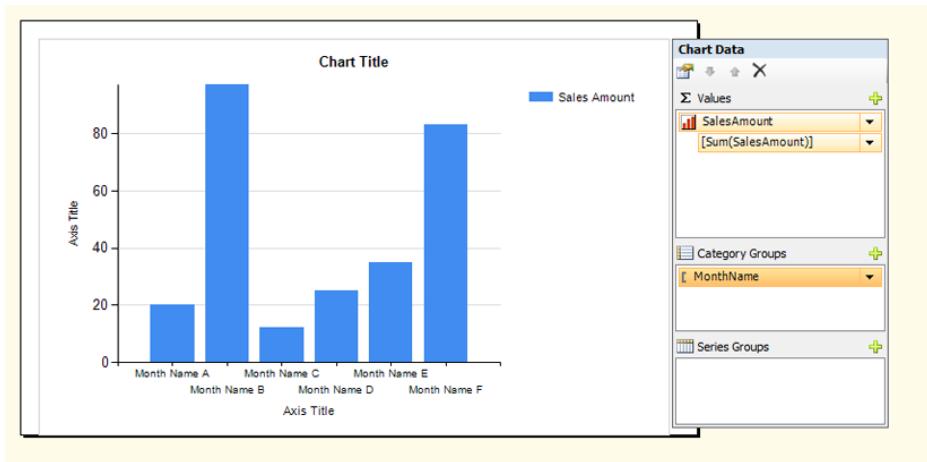


Figure 6-6

In its simplest form that's it! Add a chart title and you are ready for a quick preview of the chart. The defaults get you most of the way, and then all that is left is to clean up the chart and make it easier to read and use.

### ***Formatting the Horizontal Axis (X-Axis)***

The default interval used for the axes labels resulted in every other month being skipped. Changing this behavior is straightforward:

1. In Design view, right-click the horizontal axis and choose Horizontal Axis Properties.
2. On the Axis Options page, change the Interval value from Auto to 1 in the Axis range and interval section.
3. Since calendar months are self-explanatory, you do not need the additional noise from a title for the horizontal axis. Select the horizontal Axis Title as seen in Figure 6-7 and click Delete.
4. Click OK and then preview the chart.

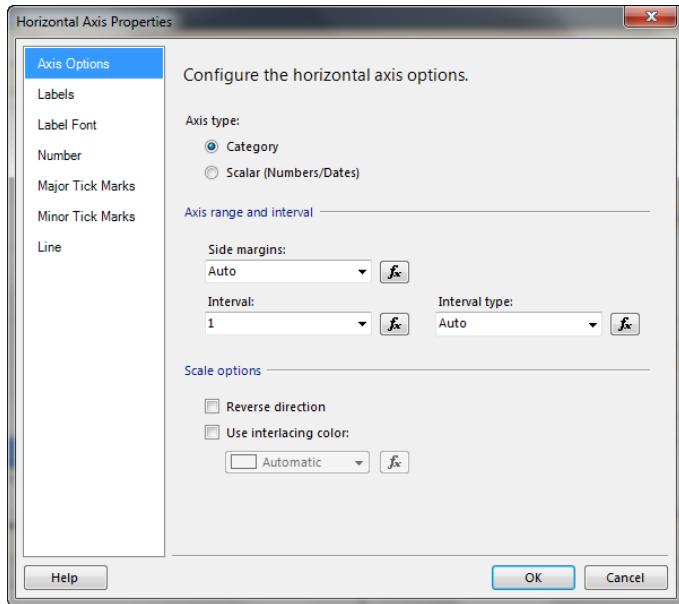


Figure 6-7

### Formatting the Vertical Axis (Y-Axis)

The reseller sales totals for Adventure Works range from just under \$4 million to almost \$10 million. The default chart display can be formatted to more clearly present the contained data.

1. Right-click the vertical axis and select the Vertical Axis Properties to begin configuring the vertical axis.
2. The first change you'll make is to the axis range. Instead of starting at zero, you want to scale the chart range. Rather than hard-coding a floor into the range, uncheck the Always include zero option in the Axis range and interval section. This option enables Reporting Services to auto-scale the range for you.
3. Now clean up the formatting of the axis labels to condense and clarify them. Select the Number page within the Properties dialog.
4. Change the Category to Currency and make the following modifications (see Figure 6-8):
  - Remove the decimal digits by entering 0 for Decimal places.
  - Check the Use 1000 separator (,) option.
  - Scale the label values by checking the Show values in option and then selecting Thousands.

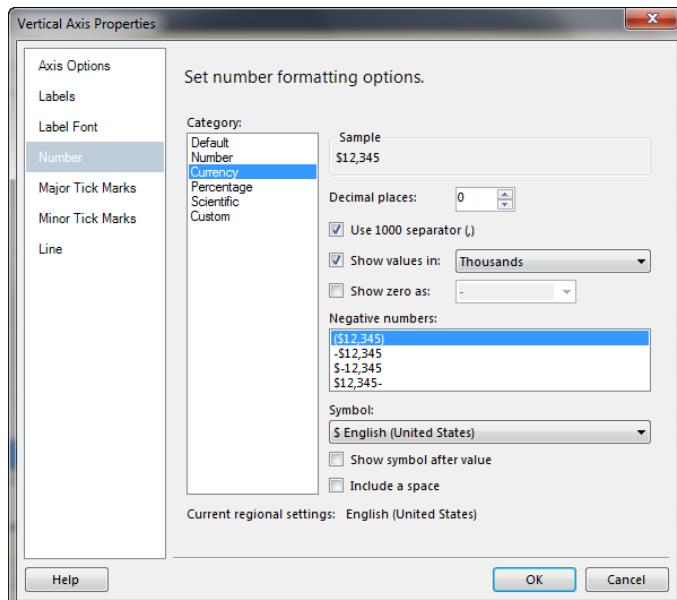


Figure 6-8

5. After making the changes to the vertical axis, repeat the previous steps to make the final change necessary so the user will know the scale used. Change the horizontal axis title to **In Thousands** to communicate this.
6. Save your changes and then preview the chart.

### *Adding a Chart Tooltip*

The goal with either dashboards or balanced scorecards is to provide charts that can be easily and quickly read and interpreted. When it is necessary to provide data point values, rather than cluttering the chart by embedding them directly into the visualization, you can make them available through pop-ups or tooltips. The Tooltip option is found on the Series Property dialog. You can right-click the series in the chart data region or directly on the bars in the graph to access the Series Properties dialog. The Tooltip field accepts simple expressions and predefined, case-sensitive keywords. Table 6-2 includes the keywords most commonly used and the equivalent expression.

**Table 6-2:** Chart Keywords

KEYWORD	DESCRIPTION	EXPRESSION EQUIVALENT
#VALY	Displays the raw value of the data point. The standard .NET formatting strings are accepted when appended to the keyword in curly brackets. For example, #VALY{C0} displays the value formatted as currency with no decimal digits.	=Fields!SalesAmount.Value
#LABEL	Displays the data point label.	Not applicable
#AXISLABEL	Is used for shape charts to display the axis data point label.	=Fields!SalesAmount.Value
#PERCENT	Displays the percent of the total data values in the series.	=FormatPercent(Fields!SalesAmount.Value / Sum(Fields!SalesAmount.Value, "MyDataSet"), 2)
#TOTAL	Displays the total or summation of the data point values in the series.	=Sum(Fields!SalesAmount.Value)
#AVG	Displays the average of the data point values in the series.	=Avg(Fields!SalesAmount.Value)
#MIN	Displays the minimum data point value in the series.	=Min(Fields!SalesAmount.Value)
#MAX	Displays the maximum data point values in the series.	=Max(Fields!SalesAmount.Value)

The complete list of keywords can be found at: <http://msdn.microsoft.com/en-us/library/dd207017.aspx>.

You can try out displaying the formatted data point value. Simply enter **#VALY{C0}** in the Tooltip field and then click OK. You can preview the report and hover your mouse pointer over a bar within the chart to see your changes, as shown in Figure 6-9.

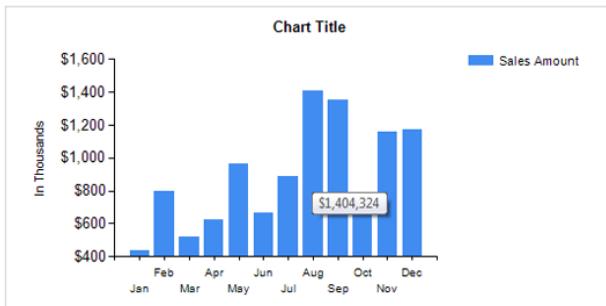


Figure 6-9

### Modifying the Chart Color Palette

The default color palette in Reporting Services leaves a lot to be desired. Luckily, like everything else you have seen, it is customizable.

1. Make sure the Chart component is selected and the Properties window is visible (press F4 to launch the Properties window).
2. When you locate the Palette property, notice that there are a number of built-in palettes to choose from. Choose one, or if none of the built-in color palettes work for your organization, you can customize your own by selecting the Custom option.
3. If you select the Custom option, you must then create your own palette of colors using the `CustomPaletteColors` property. Click the ellipsis on the `CustomPaletteColors` property to launch the `ChartColor Collection Editor`.
4. The editor functions by adding colors to the collection. Either select built-in system colors (shown in Figure 6-10) or enter a color Hex code.

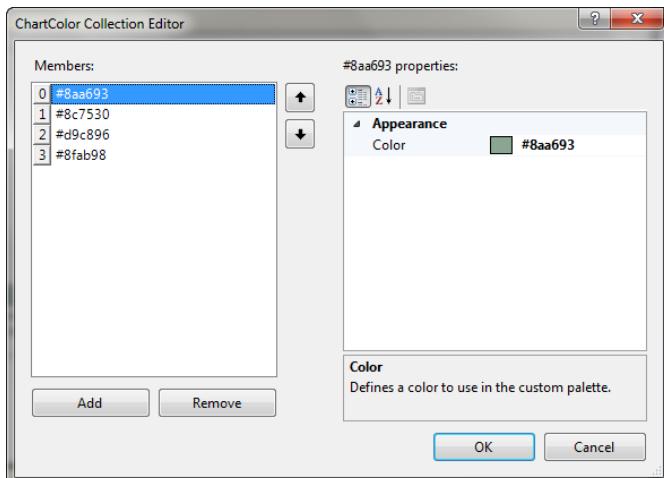


Figure 6-10

**NOTE** Creating a color palette can be tough, particularly when there are organization colors in play. A great tool to facilitate the design of your color palette is Kuler by Adobe. Kuler allows you to build a palette as well as view palettes created by other users. Kuler is a free tool available at: <https://kuler.adobe.com/>.

## Extending the Reporting Services Chart

The Reporting Services chart example you looked at in the previous section provides a solid foundation for building most of the basic charts that you will encounter. But you have only scratched the surface of what's possible in Microsoft SQL Server Reporting Services.

In this section you take a look at a couple more advanced scenarios that are possible within a Reporting Services chart, including defining trendlines, using a secondary axis, and using expressions to dynamically control color within the chart to draw attention to an exception or other important data event.

### *Defining Trendlines*

*Trendlines* are lines that are added to a chart (usually a bar chart) to illustrate trends and facilitate the analysis of the data within the chart. Adding a trendline to a Reporting Services chart involves adding a new series to the chart.

There are two types of series: The first is the *standard series* you used previously to chart the data from the prior chart example. The second type of series is called a *calculated series*. A calculated series is one that is calculated from another series with a predefined formula. The predefined formulas that are available consist of multiple types of moving average, Bollinger bands, mean, median, relative strength index, MACD, standard deviation, and a few more. See Table 6-3 for a more thorough description of each of these formulas. The most common use of the calculated series is the addition of a moving average trendline to a chart.

**Table 6-3:** Calculated Series Formulas

FORMULA	DESCRIPTION
Moving Average	This is sometimes referred to as the sliding or rolling average because it's calculated on a subset of the data using some interval. Common intervals can be three-, six- and twelve-month moving averages.
Bollinger Bands	Bollinger Bands visualize the volatility of data and consist of three lines: a moving average, an upper band, and a lower band, which are both one standard deviation above and below the point.

FORMULA	DESCRIPTION
Mean	This typically refers to the arithmetic mean or the average.
Median	The median is the statistical midpoint for a dataset where half the points are higher and half the data points are lower.
Relative Strength Index	This is a special type of indicator used in technical financial analysis that measures the velocity and magnitude of data point changes.
MACD	MACD is the acronym for Moving Average Convergence Divergence and is a technical financial analysis measure that measures the strength, direction, momentum and duration of a change in data point or stock price.
Standard Deviation	Standard Deviation is a statistic principal that measures the variation of a data point from the average or expected value.

Continue on the previous example on modifying the chart color palette by adding average and moving average trendlines using both a standard and calculated series.

1. With the report and chart from the prior example open, select the chart to display the chart data region.
2. Click the green plus-icon in the Values section and select the `SalesAmount`. This adds a second series instance of the `[Sum([SalesAmount])]` as seen in Figure 6-11.

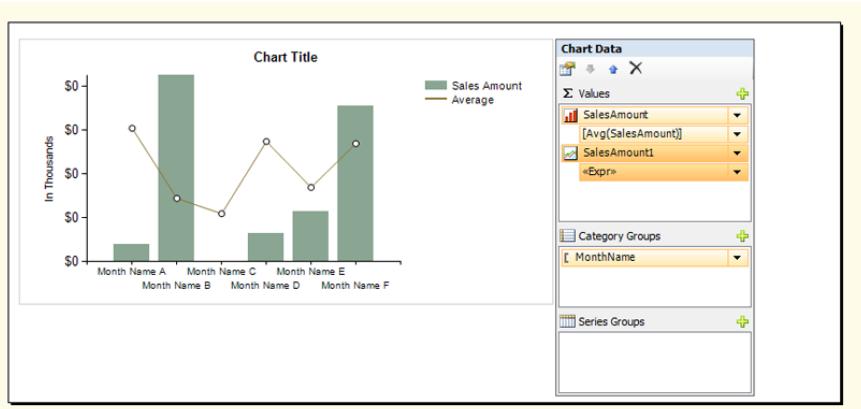
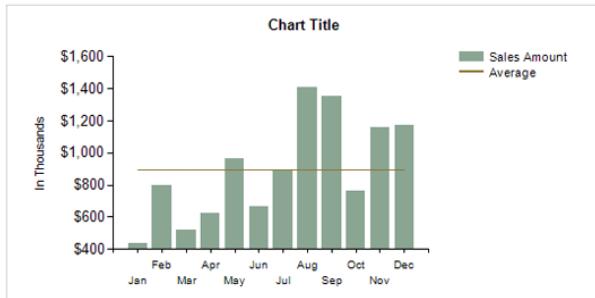


Figure 6-11

3. To calculate the average of the data, open the Series Properties dialog and use the following formula as the Value expression. Note that it may be necessary to either rename your dataset or update the formula to replace the "MyDataSet" with the appropriate name.

```
=Sum(Fields!SalesAmount.Value, "MyDataSet") /  
Count(Fields!SalesAmount.Value, "MyDataSet")
```

4. To show the average value as a tooltip, use the same formula in the "Adding a Chart Tooltip" section for the Tooltip: #VALY{C0}.
5. Before closing the Series Properties, click the Legend page and set the Custom legend text to Average so that the line will be described appropriately. Click OK to close the Properties dialog.
6. Now the chart type for the series needs to be changed so that the trendline shows as a line instead of bar. Right-click the series, choose Change Chart Type and then select the Line chart. This is a good illustration of Reporting Services' ability to combine chart types to produce powerful visualizations.
7. Save your changes and then preview the results, shown in Figure 6-12.



**Figure 6-12**

8. You can also add a second trendline to include the moving average. The moving average trendline uses the calculated series. Right-click the original bar graph series in the chart data region and select Add Calculated Series.
9. In the Calculated Series Properties, select the formula to apply. The default is Moving average, which works for this example. Notice that there are formula parameters that can be configured to set the number of periods as well as whether the formula starts at the first point in the series. Set Period to 6 for a 6-month moving average and check Start from first point (see Figure 6-13).

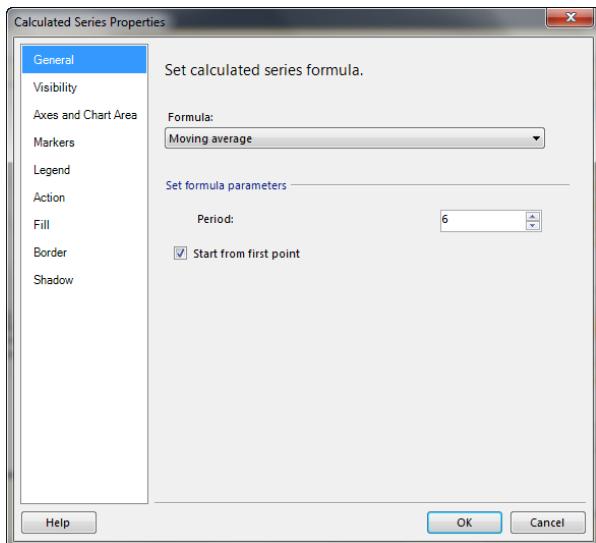


Figure 6-13

10. Before closing the properties window, again select the Legend page and change the name in the legend to **Moving Average**, then click OK.
11. Save and then preview the results. They should look like Figure 6-14.

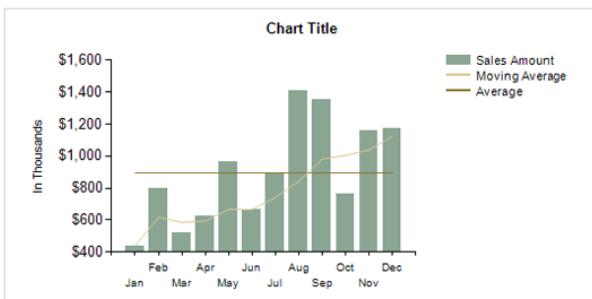


Figure 6-14

### Configuring a Secondary Axis

In some combined series scenarios, the primary axis has either an incompatible range or a range of values that obscures the visualization. In these situations, a secondary axis can be added to compensate. You can add a secondary axis for the Moving Average series by completing the following steps:

1. Click the calculated series in the chart data region to open the Calculated Series Properties.

2. Select the Axes and Chart Area page, change the Vertical axis from Primary to Secondary, and then click OK.
3. Format the secondary axis and change the Axis Title in the same manner that you formatted the primary axis earlier in the “Formatting the Horizontal Axis (X-Axis)” section of this chapter.
4. After your formatting is complete, save and preview the chart, which should appear as shown in Figure 6-15.

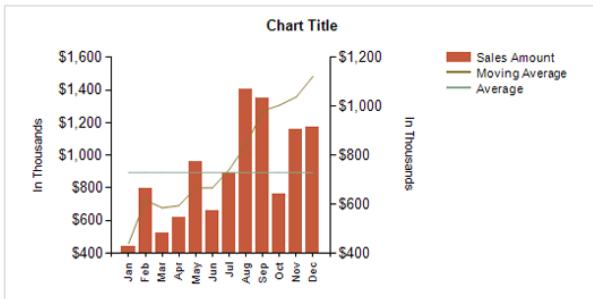


Figure 6-15

### Dynamically Controlling Color within a Chart

Analytical reports and dashboards in particular are data dense, and there are times when it's necessary to alert the user to an important event or exception situation. One way of accomplishing this is by dynamically highlighting a portion of the chart with a different color so that it stands out. For this example, you highlight months in which sales are below the \$600,000 mark.

1. Ensure the bar chart series in the chart area is selected and the properties window is open.
2. Find the Color property. The default value is Automatic, which depends on the chart color palette. Click the drop-down arrow and select the Expression option.
3. The expression needed compares the SalesAmount to the \$600,000 mark and sets the color based on whether the value is above or below the defined threshold. In the Expression dialog, enter the following expression and then click OK:

```
=iif(Avg(Fields!SalesAmount.Value) > 600000, "#8AA693", "#C45A3B")
```

- Save and preview the chart, which should look like Figure 6-16.

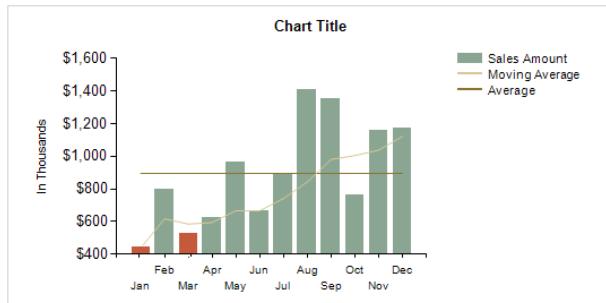


Figure 6-16

**NOTE** Instead of hard-coding threshold values into expressions, consider using report variables (discussed later in Chapter 8, “Extending Dashboards and Balanced Scorecards with Custom Code”) to either calculate the threshold or load the threshold from a query.

## Working with Sparklines and Data Bars

Up to this point, the focus of this chapter has been purely on the most common and standard chart types. It’s time to shift gears and introduce some of the other visualizations that are available and handy when designing dashboards and other analytical reports.

*Sparklines* and *data bars* are small or miniature charts that are intended to condense a large amount of information into a small space. These charts are often used in conjunction with either text or tables.

### Sparklines

Yale professor Edward Tufte originally conceived and described the theory behind sparklines in his book *Beautiful Evidence* (Graphics Pr, 2006). He proposed simple, word-sized, high-resolution graphics that function as data words.

In practice, sparklines represent a series or multiple data points over time and are tied to a single category or group within a matrix or table. This type of visualization excels at illustrating trends, and although typically represented as either a line or area graph, Reporting Services supports multiple types of sparklines.

The examples for sparklines and data bars are based on a report of Adventure Works Internet sales by country for 2007. The following select statement can be used to build your dataset:

```

SELECT
    d.CalendarYear AS [Year],
    d.MonthNumberOfYear AS [Month],
    g.EnglishCountryRegionName AS Country,
    SUM(s.SalesAmount) AS SalesAmount
FROM FactInternetSales s
JOIN DimCustomer c ON (s.CustomerKey = c.CustomerKey)
JOIN DimGeography g ON (c.GeographyKey = g.GeographyKey)
JOIN DimDate d ON (s.OrderDateKey = d.DateKey)
WHERE d.CalendarYear = 2007
GROUP BY
    d.CalendarYear,
    d.MonthNumberOfYear,
    g.EnglishCountryRegionName

```

Using the Reporting Services fundamentals from Chapter 3 as a base, you can implement a sparkline chart using the built in component, starting with creating a new report. Because both sparklines and data bars are usually displayed as part of a table or matrix, add a matrix to the report and add the Country column to the first row of the matrix. This report will be the basis for both the following sparkline example and the upcoming data bar example.

1. Drag the Sparkline component from the toolbox to the second column of the matrix and select the Area chart as the Chart Type. By adding the Sparkline component directly to the matrix, the Category group has been configured to the grouping of the matrix, which for this example is the Country.
2. Configure the definition of the series by selecting the SalesAmount column in the Values portion of the Chart Data region.
3. Resize the column and the chart will automatically resize and fill the column. Stretch the column to an appropriate size.
4. Save and preview the report; it should look like Figure 6-17.

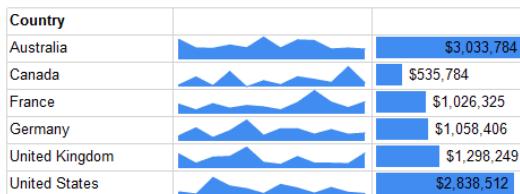


**Figure 6-17**

## Data Bars

Where sparklines represent multiple data points, data bars are used to visualize a single data point for the purpose of comparison or to evaluate performance. Use the following steps to add a data bars visualization to your report.

1. With the same report from the sparklines example, add a third column inside the group and to the right of the sparkline.
2. Drag the Data Bar component from the toolbox and drop it into the new column. Select the Bar chart as the chart type for the data bar and then click OK.
3. In the Chart Data region, set up the series using the SalesAmount column.
4. To display the data value labels within the visualization, click the series drop-down arrow and select the Show Data Labels option.
5. Before previewing your chart, format the data label by right-clicking the label that is now visible within the chart area and open the Series Label Properties.
6. Format the data labels for currency, with no decimal digits, using the thousands separator.
7. Save and preview the report, and it should appear as shown in Figure 6-18.



**Figure 6-18**

**WARNING** The sparkline and data bar visualization have a peculiar behavior when placed inside a matrix. Both charts resize to the height and width of their containing cell, which can lead to oddly shaped charts when text lines wrap in other columns. You can correct the visual issue by ensuring that your matrix cells are correctly sized to the content they contain.

**NOTE** Although this example involves only a single data point, data bars are capable of supporting multiple data points in much the same way as a bar or column chart functions. In these situations the data consists of multiple series that have no category or series groupings.

## Working with Gauges and Bullet Graphs

When most people, particularly business users, think of dashboards, the first thing that comes to mind is a collection of gauges, dials, and thermometers. Reporting Services supports all three (and several variants) through the linear and radial gauge types as well as an alternate form known as the bullet graph. In the next few sections, you will be introduced both to the out-of-the box gauge forms and the rationale and steps required to implement a bullet graph.

### Gauges

A *gauge* represents a single data point within a dataset and can be used to display data, such as key performance indicators (KPIs). Gauges can be displayed individually, embedded in tables or matrices, or grouped into a panel to compare data between fields (see Figure 6-19).



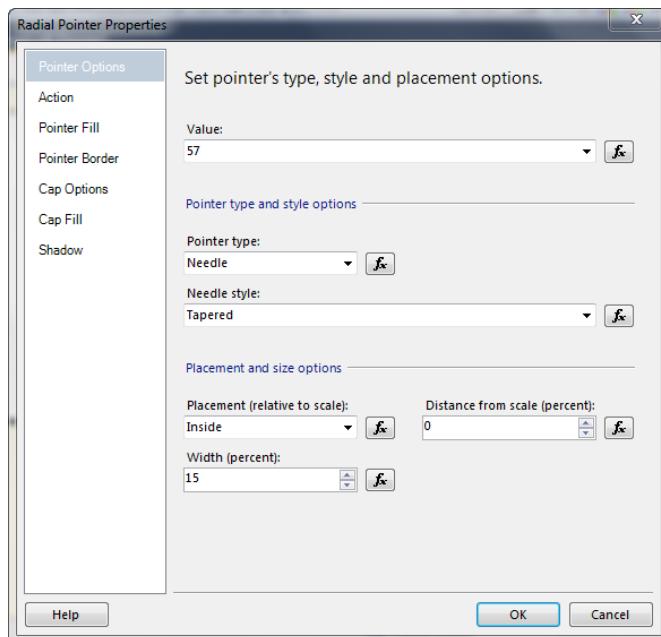
Figure 6-19

*Radial gauges* are circular or are displayed as a fractional circle ( $90^\circ$  and  $180^\circ$  slices) and resemble dials, meters, or speedometers. These gauges tend to express data as a velocity and use a needle or pointer to mark data values. *Linear gauges* on the other hand, resemble thermometers or rules and express data along a scale. Instead of needles, linear gauges use thermometers or bars as markers.

Both radial and linear gauges contain configurable scales (with major and minor tick marks), gauge labels, and have an option to include a range. Linear and radial gauges are also interchangeable like other chart types such as the line and bar chart covered earlier in this chapter.

Configuring a gauge can be done in just a few steps after adding the Gauge from the toolbox to the report design surface and selecting the desired gauge type. The following example walks you through this entire process:

1. In a new report, add a Gauge control to the design surface. On the Gauge Type dialog, select the Radial Gauge.
2. Bind the dataset containing the measure or KPI values by setting the DataSetName property in the Gauge Properties window.
3. In the Chart Data region, click the RadialPointer1 (LinearPointer1 if working with a linear pointer) and open the Pointer Properties window.
4. Select the Dataset Field column containing the measure value for the Value property. To demonstrate the control, enter a static value such as 57, as seen in Figure 6-20.



**Figure 6-20**

With that, you have a fully functioning gauge. You can change the gauge type to experiment with the different linear and radial gauges to get a feel for all the design options that are available. Before moving on, look at some additional configuration options that are available:

- The pointer can be styled as a needle, marker, or bar. The default is Needle and is the most common implementation. The Marker and Bar options are useful when adding a second pointer to a gauge to mark a comparative

value such as a target or goal. You can add a second pointer to function as a target marker on the gauge by completing a few easy steps:

1. Right-click the gauge and then select the Add Pointer option.
  2. Open the Pointer Properties for the new pointer and define a static value of 80. Like the first pointer, this value could be bound to the gauge dataset.
  3. Before closing the Properties window, change the Pointer type from Needle to Marker. Click OK to close the Properties window.
- Ranges are a useful tool for highlighting an important span of values and can help the report user quickly identify whether a measure is outside an acceptable range or is approaching a potential area of concern. The Range Properties window is available by right-clicking the Range in the Chart Area. Set the Start and End range to 50 and 100 respectively for this example and then click OK.
  - The configuration options for the gauge scale are similar to those you have worked with previously. The Scale Properties pages allow you to configure the minimum and maximum scale values, scale interval, as well as the tick marks and labels. Much like the charts discussed earlier, adjusting the gauge scale is a useful tool in making your gauges easier to read. Change the minimum value from 0 to 25 and save and preview the completed gauge, as seen in Figure 6-21.



Figure 6-21

## Bullet Graphs

If you aren't familiar with a *bullet graph*, there is no immediate cause for concern as they are relatively new. Stephen Few, an authority figure in the world dashboard design, created bullet graphs as an alternative to the gauges and thermometers that have become synonymous with the dashboard.

His goal was to create a new, easy-to-understand visualization that would present a single metric along with a comparative measure (for example, a target)

and qualitative range (such as good, satisfactory, or bad). The implementation of this visualization in Reporting Services (shown in Figure 6-22) is actually embedded into the gauges component but it is discussed separately because of its usefulness and importance, particularly in dashboards.



**Figure 6-22**

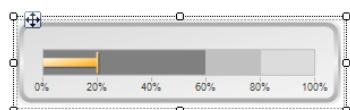
Few's vision for the bullet chart is a no-frills chart that consists of five parts that can be used either as a standalone visualization or embedded into a table in the same manner as sparklines or data bars. The five parts of the bullet chart include the following:

- Text label
- Quantitative scale
- Bar to encode the measure value
- Symbol or marker that encodes the comparative value
- Background fill colors that encode the qualitative range (in this case good, satisfactory, or bad)

The Reporting Services implementation includes the five original parts but also comes with a "shiny wrapper" frame that Stephen Few would likely frown upon because it does nothing but distract from the data contained within. The example you will build for this section modifies the built-in control to better align it with his original vision.

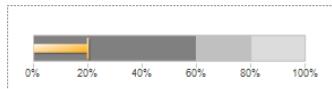
1. In a new report, create a new dataset with the query below and add a Gauge component from the toolbox to the design surface. The Bullet Graph option is found under the Linear Gauge types. Figure 6-23 shows the default bullet graph when added to the design surface from the control toolbox.

```
SELECT 20 AS ActualValue, 100 AS TargetValue
```



**Figure 6-23**

2. Start by removing the non-data pixels, which primarily consist of the decorative frame surrounding the graph. Right-click the chart and open the Linear Gauge Properties dialog.
3. On the Frame page, change the Style from Edged to None to hide the frame, as demonstrated in Figure 6-24.



**Figure 6-24**

4. If you intend to use the bullet graph in a standalone manner, you should find the quantitative scale, which provides the range of values much like an axis on a more traditional chart, to be a useful tool. Otherwise, if the chart is going to be embedded with multiple bullet graphs, you can hide the scale to reduce the amount of noise.

Your work is now complete except for mapping your dataset to the control. To map the dataset to your bullet graph, use the chart data window as shown in each previous example.

5. Alternatively, if you plan to embed the chart into a table, there's more work to do, starting with hiding the quantitative scale. Right-click the chart, select Gauge Panel, and then Scale Properties. On the Labels page, check the Hide Scale Labels option and then the Hide Major Check Marks option on the Major Tick Marks page.

If you embed the bullet graph in a report cell as is, you will immediately notice a problem. The graph resizes to the cell size making it practically illegible. You need to adjust the sizes of each part of the graph so it scales better.

6. The `LinearPointer1` represents the bar that encodes the measure value. In the Pointer Properties, link the Value to the `ActualValue` column in your dataset.
7. To fix the scale issue, change the Width in Placement and size options from 10 percent to 30 percent.
8. `LinearPointer2` represents the symbol that encodes the comparative or the target value. Link it to the `TargetValue` column in your dataset and set the Width to 8 percent and Length to 80 percent.
9. For each of the three default ranges, Figure 6-25 configures the properties such that the Start and End widths are 60 percent.

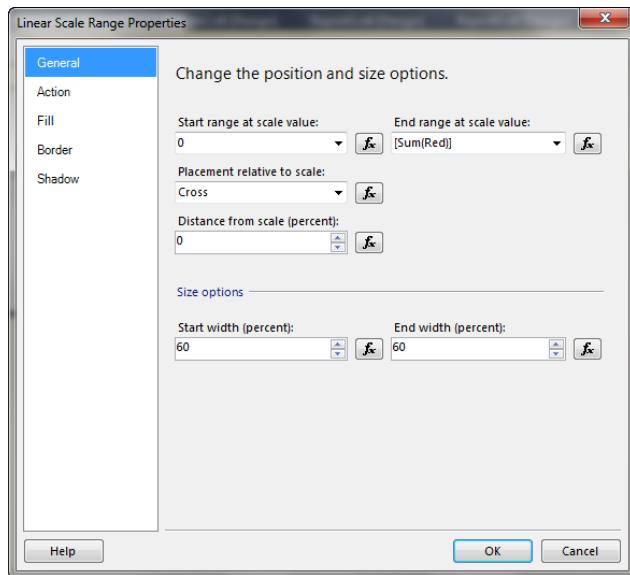


Figure 6-25

### BULLET CHART RANGES

By default, three ranges (Good, Satisfactory, and Bad) are included. Ranges can be removed or added if fewer or additional ranges are needed. The range boundaries can also be loaded from the dataset by mapping the start and end range value properties to columns in the dataset.

The chart, as seen in Figure 6-26, is now ready for use within a table or matrix.



Figure 6-26

## Working with Indicators

*Indicators* are like gauges in that they represent a single measure or data value. Instead of encoding the actual value of the data point however, indicators effectively visualize the state, condition, or trend of the measure in a glance.

Gauges can stand alone and require a fair amount of real estate, but indicators are most commonly found in tables and matrices and can be very small. In fact they remain equally effective even when scaled to a very small size, making them ideal candidates for inclusion in data rich reports such as dashboards and balanced scorecards.

Despite their differences in appearance, underneath the covers Reporting Services indicators are nothing more than small gauges that have neither a frame nor pointer. The Reporting Services implementation of indicators includes four types of indicators (see Figure 6-27):

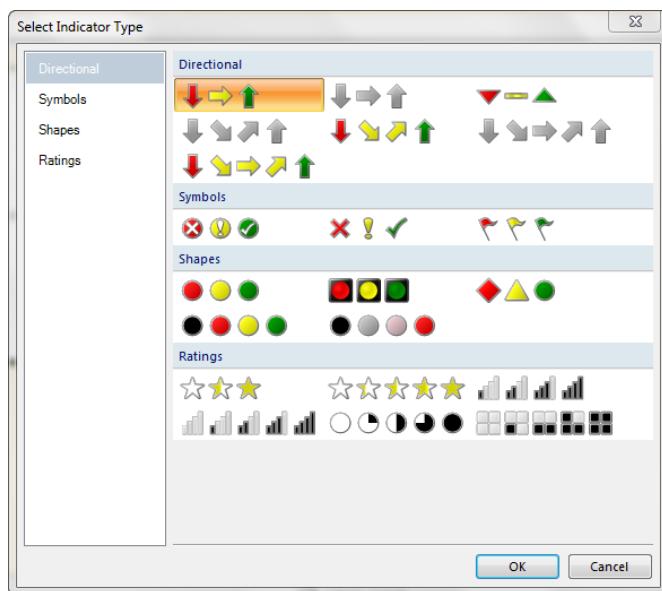
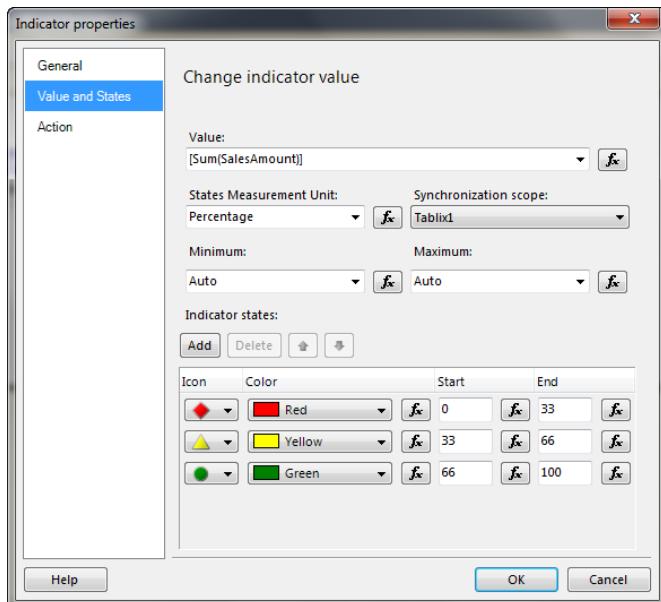


Figure 6-27

- **Directional:** Consists of arrows, which are ideal for showing trends (up, down, and flat).
- **Symbols:** Contains checkmarks, exclamation points, and flags, which are used to show state.
- **Shapes:** Comprised of traffic lights or signs, which are used to show condition (red, yellow, and green).
- **Ratings:** Includes stars, bars, and squares, which are used to show progress or a scale.

The Indicator component is most commonly added to a report by way of a column in a table or matrix. The indicator value and states are all configured as seen in Figure 6-28, through the Indicator Properties window.



**Figure 6-28**

By default, the indicator starts with three states (red, yellow, and green), but you can easily add or remove states to accommodate the requirements of your report. Figure 6-29 demonstrates a complete chart with indicators embedded.

Group	Country	Region	
Europe	France	France	◆
	Germany	Germany	◆
	United Kingdom	United Kingdom	◆
North America	Canada	Canada	●
	United States	Central	▲
		Northeast	◆
		Northwest	▲
		Southeast	▲
Pacific		Southwest	●
	Australia	Australia	◆

**Figure 6-29**

Another feature of indicators enables them to be embedded into gauges, since both indicators and gauges share a common base (the Gauge Panel). Indicators can be docked into a gauge by dragging and dropping them onto the gauge design surface. The techniques for configuring the indicators are identical in this scenario to those previously discussed in the section and they further enhance the visualization by including the state of the measure or KPI.

## Working with Maps

---

Although *maps* are not one of the more common visualizations used in dashboards and balanced scorecards, other reports, such as process maps and performance dashboards, make them a critical visualization to know. The map name itself sometimes leads to confusion. When most people hear the name “map,” it is instantly associated with the geographic variety, like a world or road map. There is, however, a range of other maps that can be relevant, including office or building maps and even maps of important business processes.

Luckily, Reporting Services has robust support for mapping charts, including a built-in map gallery and support for both ERSI shape files and SQL Server Spatial queries. Maps are composed in Reporting Services in layers, which is the most atomic concept of a map. A map’s composition is created by adding one or more layers. There are four types of layers to be aware of:

- **Polygon:** The polygon layer consists of data and points required to build an outline, such as a country, region, or state. The Map Gallery has the United States at the county level available out of the box.
- **Point:** The point layer is a data layer that consists of specific points on the map. For example, mapping store locations or sales on the map.
- **Tile:** A tile layer functions as the background for the map. Out of the box, Reporting Services supports Bing Maps integration to provide road, aerial, and hybrid map tiles.
- **Line:** The line layer is another data layer that consists of lines or paths between points.

When the layers are composed together, rich and interactive maps are possible. The following example will better illustrate what’s available to you with this visualization.

The basis of this example is a map that encodes Adventure Works’ employees who are located in the U.S. and their prior year’s sales. You will use the encoded

spatial location, which is contained in a geography data type from the Address table to map each employee's point in the point layer.

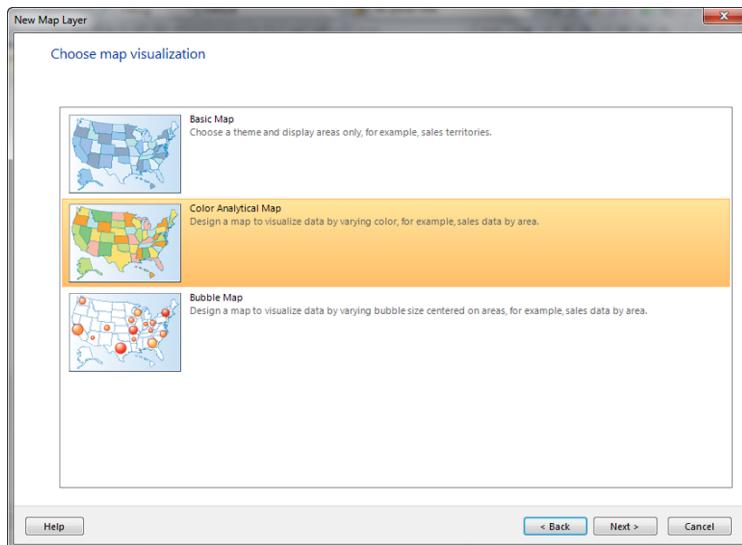
Use the following query against the AdventureWorks database as the data source to create the dataset for your map report:

```
SELECT
    p.FirstName + ' ' + p.LastName AS Employee,
    p.City,
    RTRIM(sp.StateProvinceCode) AS [State],
    p.SalesLastYear,
    a.SpatialLocation
FROM Sales.vSalesPerson p
INNER JOIN Person.BusinessEntityAddress ea
    ON p.BusinessEntityID = ea.BusinessEntityID
INNER JOIN Person.Address a ON ea.AddressID = a.AddressID
INNER JOIN Person.StateProvince sp ON a.StateProvinceID = sp.StateProvinceID
WHERE
    CountryRegionName = 'United States' AND SalesLastYear != 0
```

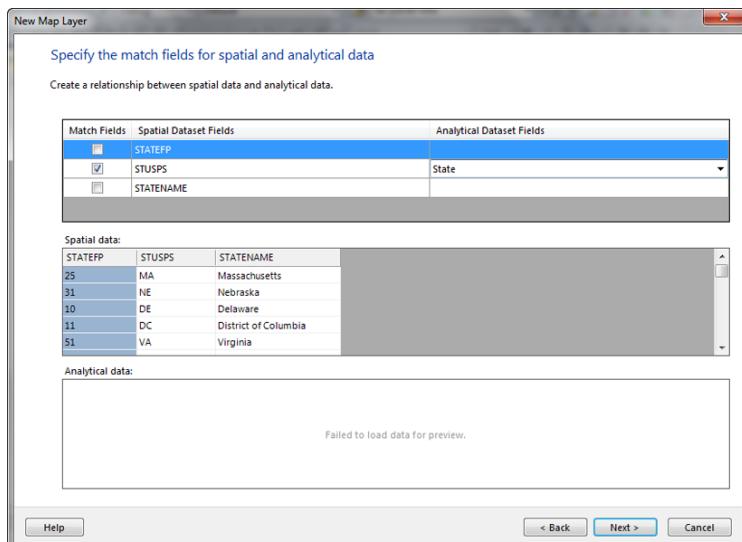
Once you have created the report and dataset, it's time to get started.

### Creating the Polygon Layer

1. Drag the Map component to the design surface to launch the Map Layer Wizard. Rather than using the wizard, click the Cancel button so that you can add each layer to the map manually.
2. Click the map on the surface to show the Map Layers region. The first layer you add is the polygon layer. Click the New Layer Wizard button.
3. There are three sources of spatial data available for the polygon layer: the Map Gallery, which is a set of built-in maps, ESRI shape files, which are standardized map encodings available from multiple sources on the Internet, and SQL Server spatial queries. For this example, use the built-in Map Gallery. Select the USA by State option and click Next.
4. Click Next again to skip the spatial data and map options for now.
5. Map visualizations range from a Basic Map, which does not encode any data in the polygon layer, to a Bubble Map, which encodes data values by changing the size of a marker fixed at specific points on the map. For this example use the middle group, the Color Analytical Map, which encodes data by color variation within the polygon boundaries (see Figure 6-30).

**Figure 6-30**

6. Next you must link the dataset to the polygon layer and create the required relationship between the built-in map data and the dataset so that your Color Analytic Map functions properly. Choose the dataset you created and click Next.
7. Map the STUSPS field in the map data to the State column in the Adventure Works dataset as demonstrated in Figure 6-31 and then click Next.

**Figure 6-31**

8. To configure the visualization portion of the polygon layer, select `SUM([SalesLastYear])` as the Field to Visualize and click Finish.
9. The polygon layer is now complete. Before previewing the map for the first time, delete Alaska and Hawaii so that the map scales better. Also, hide the Meridians and Parallels by right-clicking the map design surface and deselecting each option. Now you are ready to preview the polygon layer.

### Adding a Point Layer

1. The point layer of this map will encode the sales representatives as points and labels onto the map. In the layers region of the Map control, begin by clicking the Add Layer icon and then choosing the point layer.
2. Right-click your new Point Layer and select Layer Data. In the Map Point Options dialog, use the spatial field in the dataset option to link the layer to the spatial data within the dataset.
3. To make the dataset available to the points, click the Analytical data page.
4. Select your dataset and add map the data columns from the spatial dataset to the analytical dataset as seen in Figure 6-32. Click OK to close the Properties dialog.

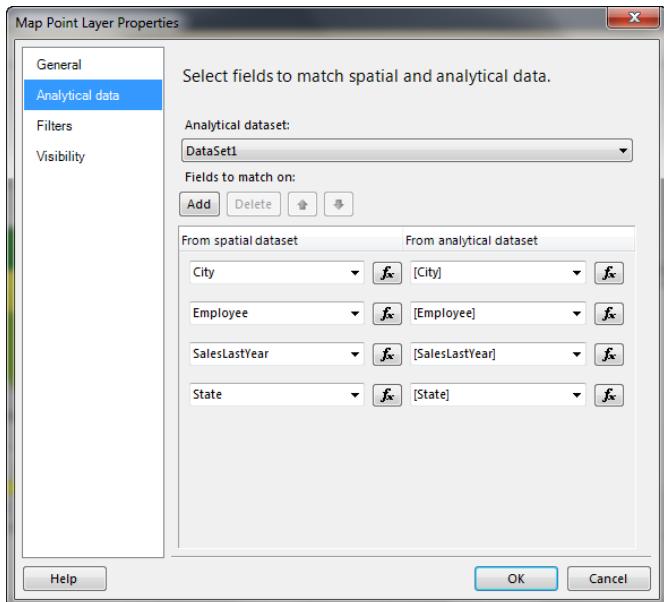


Figure 6-32

5. Add labels to the point layer by setting the Label in the Map Point Layer Properties dialog. The Tooltip option is also available as in previous examples. Use the following expressions to label each point with the city and state name and provide a tooltip pop-up with each sales representative's name and sales data:

**Label:**

```
=Fields!City.Value + ", " + Fields!StateCode.Value
```

**Tooltip:**

```
=Fields!Employee.Value + " (" + Format(Fields!SalesLastYear.Value, "C0") + ")"
```

6. Save the report and preview the map with both the polygon layer and point layers.

### Adding Tiles

1. Back in the map layers region, again add a new layer, only this time select a Tile Layer.
2. In the Tile Properties, you can select which of the Bing Map types you want to tile your map with. For this example, use the Road tile type and then click OK to close the dialog (see Figure 6-33).

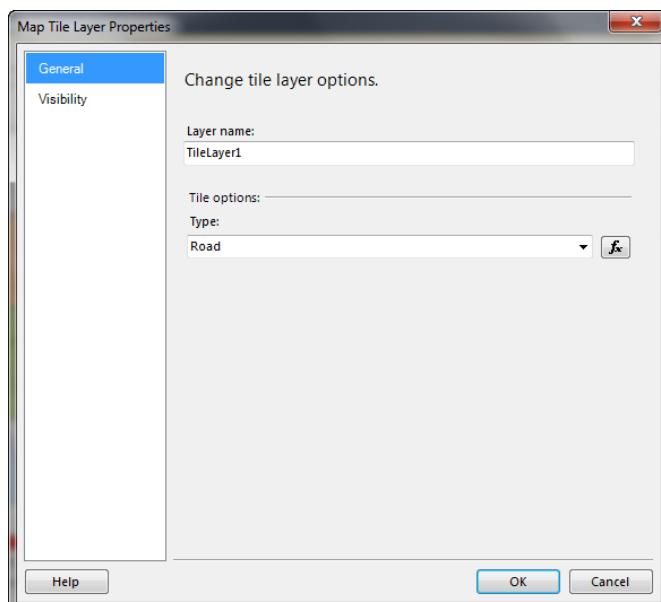


Figure 6-33

3. If you were to preview the map right now, you would notice that there is a problem. The polygon layer's background fill obscures the map tiles. To fix this problem, edit the polygon layer properties, changing the fill for the layer from white to the no-color option.
4. Since the polygon data is also encoding analytical data, it is necessary to change the transparency of the shading. This setting is found on the Visibility page of the Polygon Data Layer Properties. Use a value of 50% for this example.
5. Figure 6-34 shows the completed map after you have saved and previewed your map.

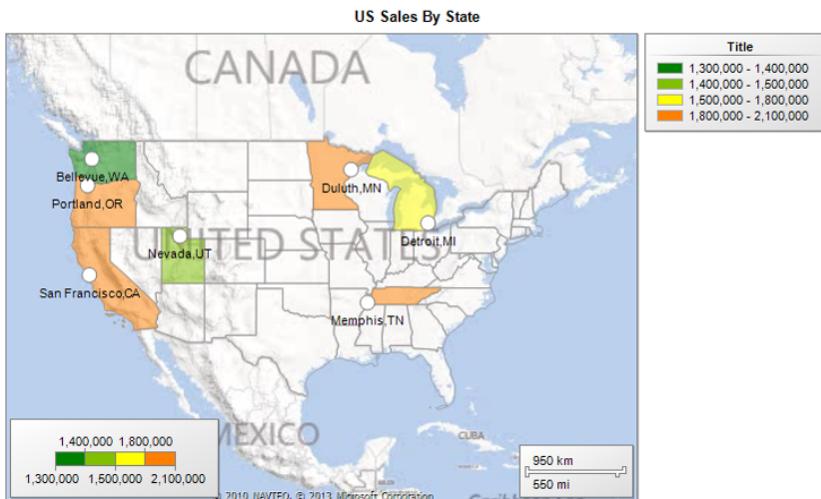


Figure 6-34

In this section, you used the built-in Reporting Services map control to add an analytic map for displaying geospatial data. This control is capable of encoding a large amount of data and presenting it in a format that enables your audience to quickly decipher and decode it, simplifying the decision-making process.

## Summary

Visualizations come in numerous forms from the typical bar and line charts, to gauges, indicators, and even maps. Properly used data visualizations can contribute as critical pieces in your dashboard and balanced scorecard projects. The series of general guidelines for working with charts, graphs, sparklines, data bars, gauges, bullet graphs, indicators, and maps that you learned in this chapter should serve as a framework as you move forward in designing your visualizations.



# Mobile Dashboards and Balanced Scorecards

If you look around during your next business presentation or meeting, the one thing you will inevitably find is a plethora of smartphones, tablets, and other mobile devices. And, if your business isn't already clamoring for business intelligence data on these devices, it certainly will be, as growth in the mobile market is exponential.

Although the mobile business intelligence market is still in its relative infancy, it's quickly becoming more reality than hype. This transition requires that traditional thinking in the context of planning and building reports or dashboards that are delivered on paper or computer monitor be tossed out the window and reimaged. In this chapter, you will be introduced to the tools, techniques, and considerations for implementing mobile dashboards and balanced scorecards.

## Planning for the Right Audience

---

As you begin planning your mobile business intelligence (BI) implementation, start by evaluating the potential or primary audiences that will consume or use the dashboards, balanced scorecards, and other analytical reports you will be building. The audience typically determines both the type and depth of reports required.

First, consider that your primary audience is either executives or high-level managers or directors. This audience generally favors dashboards, key performance

indicators (KPIs), and high-level reports that are highly summarized or aggregated. This audience is generally interested in seeing how broad strategic metrics, such as revenue, are tracking versus defined objectives or goals. Managers, sales associates, or field technicians, on the other hand, will require reports that offer much greater detail, in addition to a contextual dashboard. This audience often needs the “nitty-gritty” level such as sales order line item detail.

Your audience and their requirements will help drive subsequent decisions, from selecting a delivery mechanism and technology to design and performance considerations.

## Browser or Native?

After identifying the audience, the next decision you will inevitably be faced with is whether you will use a browser-based or native solution to deliver your dashboard or balanced scorecard. Since this is a book about Microsoft Reporting Services, the focus of this chapter is on the Reporting Services browser-based implementation and the loosely related Microsoft PerformancePoint for SharePoint. Note that it's important not to discount native mobile business intelligence solutions, as there are pros and cons (see Table 7-1) to be considered.

**NOTE** There are multiple vendors who provide both browser-based and native business intelligence development platforms. These options are intentionally not discussed but should be considered and evaluated as part of an enterprise mobile business intelligence initiative.

**Table 7-1:** Browser-based and Native Mobile BI Comparison

	BROWSER-BASED	NATIVE
PROS	<ul style="list-style-type: none"> <li>■ Works across multiple platforms and devices (Android/iOS/Windows/Blackberry).</li> <li>■ It is quick and relatively cheap to develop/implement.</li> <li>■ Data is not downloaded, which limits security exposure.</li> <li>■ Authentication support exists through host browser.</li> </ul>	<ul style="list-style-type: none"> <li>■ Data is downloaded or cached and can be analyzed in offline mode.</li> <li>■ Advanced navigational support exists including techniques such as pop-up menus.</li> <li>■ Offers more control over screen real estate.</li> <li>■ Support exists for multi-gesture controls.</li> <li>■ It can be integrated with other applications.</li> </ul>

	BROWSER-BASED	NATIVE
CONS	<ul style="list-style-type: none"> <li>■ Navigation options are limited.</li> <li>■ Screen real estate is limited.</li> <li>■ Data is not downloaded; requires online connection.</li> <li>■ It is dependent on host browser CSS implementation.</li> </ul>	<ul style="list-style-type: none"> <li>■ It is slower and more expensive to develop/implement.</li> <li>■ It is device and vendor specific.</li> <li>■ Operating system and hardware updates can break application.</li> <li>■ Downloaded/Cached data poses a security risk.</li> </ul>

The remainder of this chapter focuses on the considerations required to implement a solution in either Reporting Services or PerformancePoint and offers a brief introduction to PerformancePoint for those readers who are unfamiliar with this technology.

## Choosing a Mobile Platform

Although often overlooked for their ability to act as mobile business intelligence platforms, both Microsoft Reporting Services and PerformancePoint for SharePoint are robust and powerful tools capable of delivering high-value, browser-based mobile dashboards and balanced scorecards. Each platform has its own strengths and weaknesses.

### The Pros and Cons of Reporting Services

Throughout this book, you have seen many of the features and capabilities of Reporting Services. From advanced formatting techniques discussed in Chapter 3, “Building your First Dashboard and Balanced Scorecard” to drillthrough reports covered in Chapter 5, “Interactive Dashboards and Balanced Scorecards,” and data visualizations covered in Chapter 6, “Visual Dashboards and Balanced Scorecards,” these topics are all relevant to the mobile discussion. As a platform for your mobile dashboard, Reporting Services excels in three areas:

- Dashboards and scorecards that require drillthrough or Report Action actions that display more detailed information (see Chapter 5).
- Dashboards and scorecards that require advanced data visualizations, such as bullet graphs, sparklines, data bars, and maps (see Chapter 6).
- Dashboards and scorecards that require a high degree of control over the layout of report elements, commonly referred to as *pixel-perfect layouts*.

The relative strengths of Reporting Services are counter balanced by what can be considered weaknesses in the context of a mobile dashboard platform:

- Because reports are rendered via a web browser, they are subject to the whims of the browser's interpretation of the Reporting Services cascading style sheet (CSS).

**NOTE** *Cascading style sheets (CSS)* is a technology from the web domain that provides formatting directions to the web browser on how it should interpret styles that are embedded in the web page HTML code.

- There is no method to enforce report consistency within Reporting Services. The degree of control that Reporting Services provides over report formatting often leads to inconsistent dashboards and balanced scorecards, because developers typically have different styles.
- All dashboard and scorecard actions within Microsoft Reporting Services result in a postback of the data to the server. The postback of the data causes the entire page to refresh, which generates page flicker that is typically disconcerting to mobile users.

**NOTE** *Postbacks and callbacks* are web behaviors typically associated with a web page. When a postback is invoked, data is sent to the web server in the form of an HTTP POST and the entire webpage is refreshed. This causes the screen to flicker.

The alternative is a callback which happens when JavaScript is used to send data asynchronously back to the web server in a manner that enables portions of the web page to be refreshed. This is the technique used in PerformancePoint when data is refreshed.

All things considered, Reporting Services remains a solid platform for building and delivering mobile dashboards and balanced scorecards in any organization. This is particularly true beyond what's outlined in this section if your organization already supports Reporting Services for either reporting, dashboards, or balanced scorecards.

## The Pros and Cons of PerformancePoint

PerformancePoint is an application that falls under the SharePoint umbrella of products. It is installed as part of the overall business intelligence capabilities of the SharePoint platform. Figure 7-1 displays a full PerformancePoint dashboard as it appears when published.

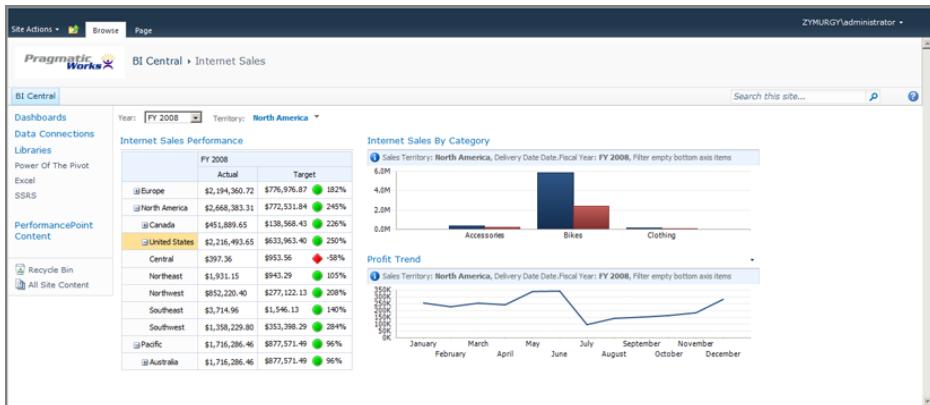


Figure 7-1

Figure 7-2 shows the same dashboard in the Dashboard Designer, which is the tool used to compose a PerformancePoint dashboard.

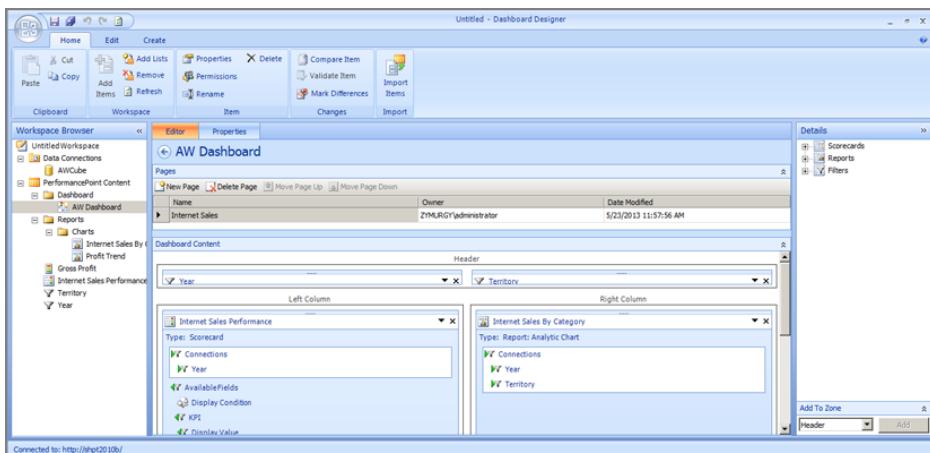


Figure 7-2

**NOTE** The intent of this section is to introduce you conceptually to PerformancePoint, and not to walk you through a full implementation. For further information on implementing a PerformancePoint dashboard, check out the MSDN Hands-On lab at: [http://msdn.microsoft.com/en-us/sql10r2byfbi-trainingcourse\\_sql10r2byfbi08-hol-01.aspx](http://msdn.microsoft.com/en-us/sql10r2byfbi-trainingcourse_sql10r2byfbi08-hol-01.aspx)

The dashboard shown in both Figure 7-1 and 7-2 consists of six distinct parts, including the following:

- **Data Source:** The data source (see Figure 7-3), as its name implies, is the primary connection to the data that is used to render the dashboard. The most common and also most powerful source for data is an OLAP or Analysis Services cube.

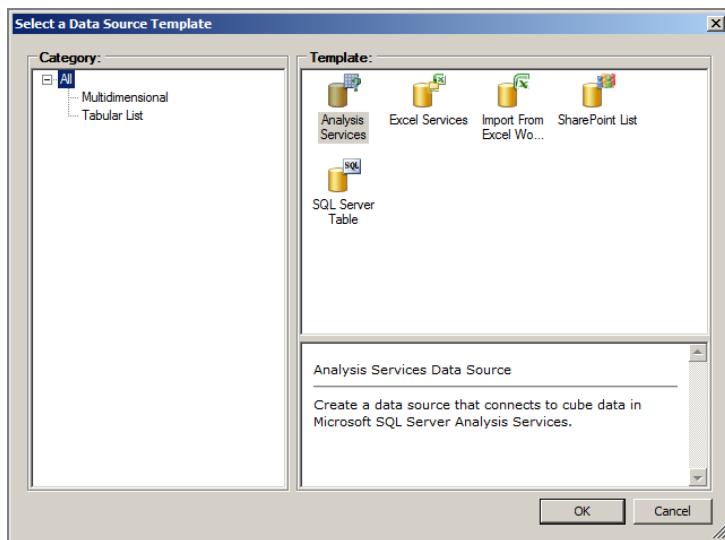


Figure 7-3

**NOTE** If you do not currently have a cube and are not planning to build one out, PerformancePoint also supports PowerPivot, Excel, SharePoint Lists, and SQL Server Tables as sources of data, although their functionality is far more constrained.

- **Filters:** Filters enable you to define the context within the dashboard by selecting a subset of the data. They exist external to all other objects, enabling a single filter to affect multiple objects through connections that are configured, as shown in Figure 7-4.
- **Indicators:** Indicators (see Figure 7-5) are visual representations of key performance indicators (KPIs). They are very similar to the indicators found within Reporting Services, discussed previously in Chapter 6.
- **Scorecards:** Scorecards (see Figure 7-6) are simply containers or collections of indicators.

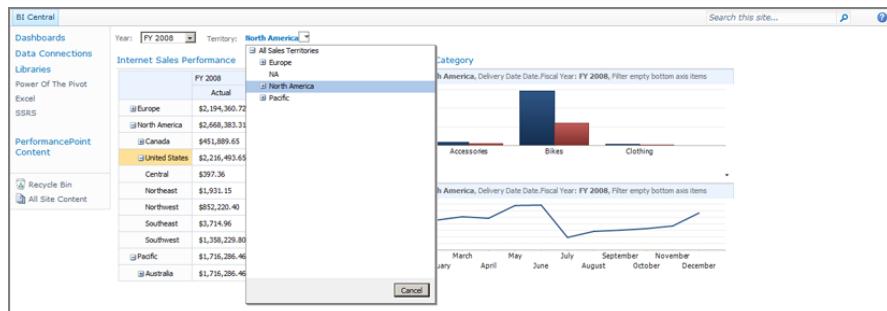


Figure 7-4

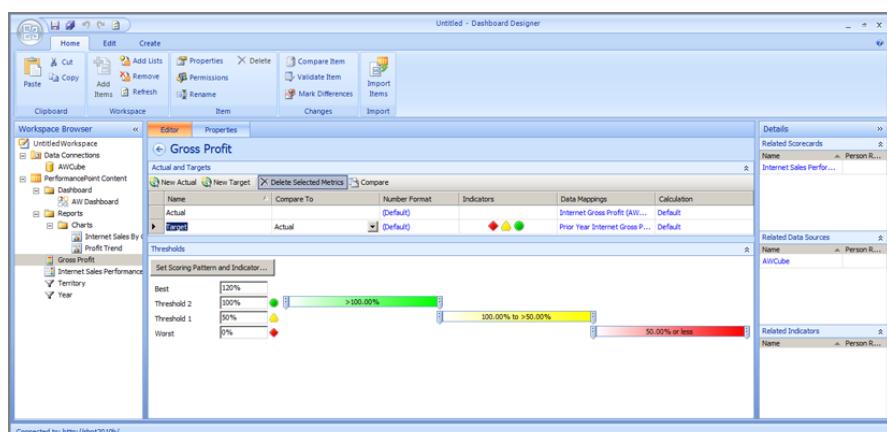


Figure 7-5

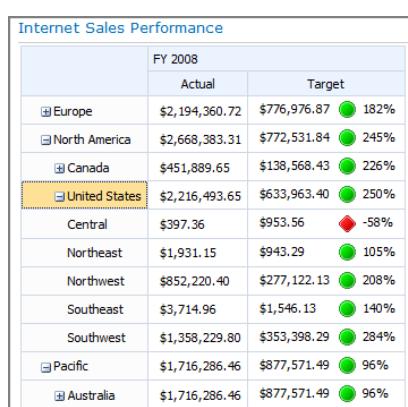
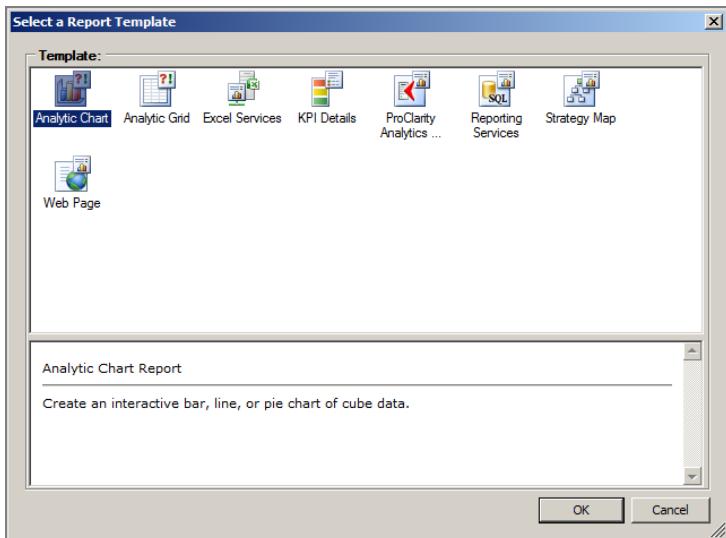


Figure 7-6

- **Reports:** Reports within PerformancePoint is a broad grouping that encompasses many different types of supported reports, as shown in Figure 7-7. The common reports you would expect to find, such as Reporting Services reports, Excel (via Excel Services in SharePoint), and external web pages, are all there. In addition, an Analytic Grid, Analytic Chart, and Strategy Map, which are built on top of a diagram created within Visio, are also available.



**Figure 7-7**

- The Analytic Grids and Charts (see Figure 7-8) are typically the work-horses of a PerformancePoint dashboard because of the built-in interactions that are available. Users can easily explore the reports by drilling into or slicing across the data.



**Figure 7-8**

**NOTE** The *decomposition tree diagram*, which enables users to dig into the data and see how a specific data point is composed, is another useful feature of a PerformancePoint dashboard. To view the decomposition tree, right-click either a cell or chart within a PerformancePoint dashboard and select Decomposition Tree (see Figure 7-9). Figure 7-10 shows the resulting decomposition tree window.



Figure 7-9

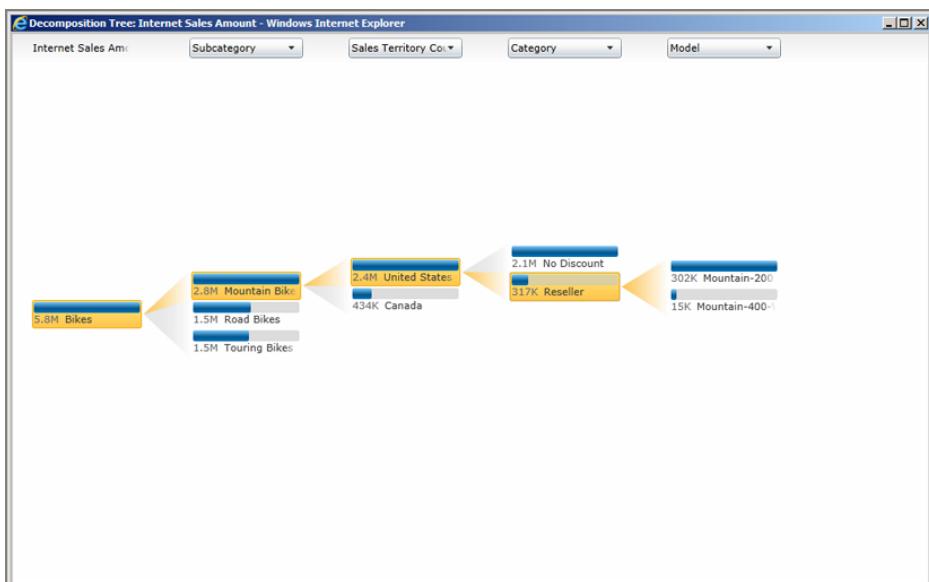


Figure 7-10

- **Dashboards:** The term “dashboard” in PerformancePoint can be a little misleading. In this context, it acts as the container that brings data sources, filters, reports, and scorecards together, as you saw in Figure 7-1. The dashboard is responsible for the layout and look and feel, and is capable of supporting multiple pages.

Now that you have a better understanding of PerformancePoint, a discussion in Table 7-2 of the pros and cons of using it as a platform to deliver your mobile dashboards will make more sense.

**Table 7-2:** PerformancePoint Mobile Platform Evaluation

PROS	CONS
<ul style="list-style-type: none"> <li>■ Robust out-of-the-box functionality enables rapid development.</li> <li>■ It enforces a consistent look and feel.</li> <li>■ Support exists for ad-hoc visualizations.</li> <li>■ It uses callbacks instead of postbacks to minimize page flicker.</li> </ul>	<ul style="list-style-type: none"> <li>■ It requires SharePoint and all the costs associated with SharePoint licensing.</li> <li>■ It requires an OLAP cube for many of the out-of-the-box features to work.</li> <li>■ Customizations to the look and feel require changes to a MasterPage and CSS in SharePoint.</li> </ul>

From the strengths and weaknesses described in Table 7-2, it should be evident that the biggest difference between Reporting Services and PerformancePoint is a trade-off between out-of-the-box functionality and customization, flexibility, and control. When all is said and done, the decision made for your mobile business intelligence platform should be driven by the requirements for the solution you are designing. Practically, however, corporate policies may guide the decision if you find yourself in an organization that either does not use SharePoint or conversely does not use Reporting Services.

## General Considerations for Mobile Dashboards

Performance and security are two areas that require more careful consideration for mobile dashboards than for those delivered on traditional media. Although most people are happy (or at least more tolerant) to watch a progress indicator spin while they sit comfortably at their desktop, they are typically far less understanding when waiting for a report to render on a mobile device as they stand in front of a client. Likewise, major emphasis is generally placed on report security, as they typically contain business sensitive data of this highest order. The next sections discuss mobile dashboard performance and security in general, as well as specific considerations for Reporting Services and PerformancePoint.

## Performance

Maximizing performance for either a mobile dashboard or more traditional analytic report implementation requires that you take a bottom-up approach. This means that your report design, from the data source to the mobile medium used to deliver your report, will have performance implications.

At the lowest level, the data source, or how your data is modeled and stored, will have a significant impact on performance. In some cases, your data will come directly from a line-of-business (LOB) application, ERP, or even CRM system, in which case you will have to take into account the resources and complexities involved with querying a highly-normalized online transactional processing (OLTP) system and the degradation in performance that follows.

More commonly, the data for analytic reports is pulled from a data mart/data warehouse or an online analytical processing (OLAP) source, such as Microsoft Analysis Services cube. In the Microsoft world, these sources of data are typically modeled in a star-schema that is optimized for reporting workloads and query performance.

The *star schema* is a form of dimensional modeling introduced by Ralph Kimball as a means to simplify and optimize the summarization or aggregation of data during reporting. The star schema concept is organized around fact tables (event or transactional data such as sales, orders, or phone calls) and dimensions (such as customer, product, and time) that describe the fact.

The star schema name is derived from the appearance of the logical model that results from this methodology, as shown in Figure 7-11.

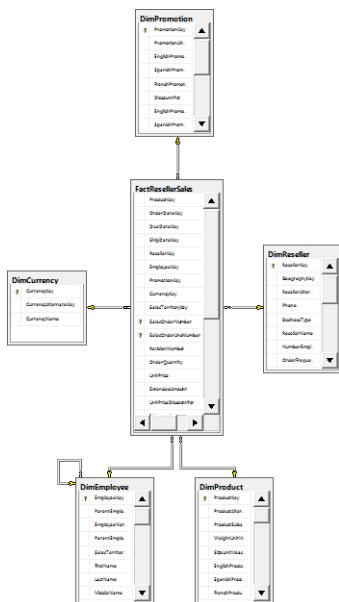


Figure 7-11

Beyond the source and modeling of the data required for your dashboard or balanced scorecard, consider these additional factors that can enhance performance:

- Database indexing strategy
- Compression within SQL Server
- Data storage on the physical disk

## Security

Planning and handling security for your mobile dashboard is one of the more complex topics to be addressed. Think for a second of your more traditional reporting environment. Whether you are using Reporting Services or another platform, this environment typically resides securely and comfortably behind a myriad of firewalls within your internal corporate network.

Public access to the network is generally tightly controlled, limiting authorized access through a security mechanism, such as a virtual private network (VPN). Users access your reports from either within the boundaries of the corporate network or by connecting in a secure manner.

With this in mind, imagine your mobile business intelligence solution as an extension of your traditional reporting environment. The key difference between the two is that the primary method by which your users access the dashboards and balanced scorecards is over mobile or cellular networks.

Mobile networks introduce a fair amount of complexity. Resolving access issues while maintaining a high-level of security requires you to adopt one of the following approaches:

- If you have a corporate VPN, many mobile devices have VPN clients that enable the devices to function as though you are working remotely from a home computer or laptop.
- Alternatively, you can publish the reporting data externally using forms-based authentication with Secure Socket Layers (SSL) for encryption between the web server and the mobile device.

The details and implementations of these security options are well beyond the scope of this chapter, and the intent is simply to make you aware of the considerations for planning. It is worth noting that in many medium- to large-sized organizations, there are teams of either network or server administrators that stand at the ready for this kind of thing. There may even be policies already in place to guide you along.

## Design Considerations

While the core principals and ideology behind delivering dashboards and balanced scorecards on a mobile device are the same, successfully delivery requires a rethink of the process and a totally new approach. It is important to understand that many of the techniques that are possible or effective in more traditional mediums don't translate well. Table 7-3 brings together just some of the most important dashboard and scorecard design considerations for successfully delivering a mobile business intelligence solution to get you started.

**Table 7-3:** Mobile Design Considerations

USABILITY	ERGONOMICS
<ul style="list-style-type: none"><li>■ Use filters, drill downs, and tooltips to add user appropriate context.</li><li>■ Avoid combining chart and graphs types, as they become more difficult to read.</li><li>■ Avoid dashboards or scorecards that require the user to zoom.</li><li>■ Consider the mobile network when planning to use bandwidth-intensive visualizations, such as maps.</li><li>■ Prefer lists of items over tables.</li><li>■ For backgrounds, use color gradients instead of solid colors.</li><li>■ Avoid unnecessary screen flicker in Reporting Services by limiting unnecessary actions.</li></ul>	<ul style="list-style-type: none"><li>■ The general rule-of-thumb for mobile devices is that buttons and links be a minimum of 44 pixels in height. Smaller buttons make it considerably more difficult for users to click.</li><li>■ Consider the range of mobile devices on which your reports will be available. If smartphones are going to be your primary target, plan on reports being viewed in portrait mode. For tablets, portrait and landscape views are needed.</li><li>■ If you are planning a dashboard that requires landscape-mode viewing, don't forget about considering the on-screen keyboard.</li><li>■ Avoid requiring your users to scroll either horizontally or vertically. Use navigation and paging instead.</li><li>■ Multi-touch gestures are great for power users but can be difficult and are typically forgotten by a normal user.</li></ul>

## Summary

With the explosive growth in the availability of powerful smartphones and tablets and the widespread adoption of these devices by not only consumers but also by business decision makers, the trend towards mobile business intelligence

solutions is growing. Both Microsoft Reporting Services and PerformancePoint for SharePoint are robust, powerful platforms for delivering browser-based mobile solutions. As you have seen, planning for this eventuality requires that you consider not only the physical dashboard or balance scorecard design implications but also the performance and security implications.

# Extending Dashboards and Balanced Scorecards with Custom Code

SQL Server Reporting Services is capable of creating eye-popping visualizations that can be consumed by a variety of business users. The tool has built-in functionality for developers to produce reports that users will find not only functional, but also visually appealing. A report developer is tasked with designing these reports using many of the native components such as charts, gauges, and sparklines, to name a few.

Along with these report tools, developers must learn Reporting Services expressions. Expressions are used frequently in reports to control content and report appearance. This expression language is powerful, but it does have some gaps in functionality that can leave a developer searching for ways to solve complex problems.

This chapter focuses on using custom code to overcome the gaps left by both native components and expressions. Learning how to use custom code with balanced scorecards and dashboards is especially helpful because requirements for these types of reports can be complex and difficult to complete with the native components and expressions. Leveraging custom code can help you overcome these difficult requirements and complete reports with ease. You'll also learn many other advantages of using custom code, such as creating consistency across all reports developed.

Before diving in, it is beneficial to have a foundational understanding of Visual Basic, because the chapter's examples all use it. The intent of this chapter is not to teach you Visual Basic but rather to show how it can be used in Reporting

Services through some real-world examples. To gain that Visual Basic foundation there are many other book resources you should start with; a good starting point is Bryan Newsome's, *Beginning Visual Basic 2012* (Wrox, 2012).

## Choosing to Embed Custom Code

Reporting Services can often produce several challenges that are difficult to overcome with the native tools provided. A report developer could easily spend weeks racking his brain trying to figure out how to manipulate the native tools and expressions in Reporting Services to fit an end-user requirement.

Using custom code can make complex user requirements a lot simpler to solve and, in the end, may even require less code but yield the same result. Embedding custom code is as simple as inputting your code and previewing the report to see how it changed the design. You also have the ability to add multiple, custom functions at once, which can't be done with native Reporting Services tools.

Another often-overlooked benefit of custom code is that it gives you one consolidated place to edit code. When using Reporting Services expressions you often apply the same expression value to multiple cells. If you needed to change an expression value, you have to open each text box property individually. With custom code you alter the embedded code in one spot, and it applies to all the necessary cells.

There are some cons that come with embedding custom code, however. Custom code must be written in Visual Basic (not C#), so you must have some previous knowledge of the language. Another disadvantage is that the embedded code window lacks a lot of the functionality of Visual Studio, which is the program where developers typically write .NET. For example, IntelliSense is not available and even basic code errors are not shown until the report is in preview mode.

**NOTE** If you don't have previous knowledge of Visual Basic, it is typically available to find what you're looking for with a few simple web searches.

Table 8-1 provides a summary of many of the pros and cons of using custom, embedded code.

**Table 8-1:** Pros and Cons of Embedded Code

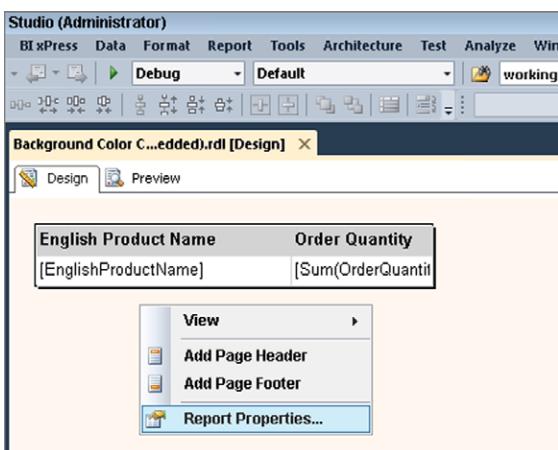
PROS	CONS
You can embed more than one function.	You must use Visual Basic.
You can immediately see the results of code without creating a Visual Studio project.	IntelliSense is not available in the code window.
You need to make changes to code in only one place.	Code errors are not visible until report preview.

## Direct Inputting Code

Once you have determined that the best way to solve your business reporting problem is with custom code, you can begin writing.

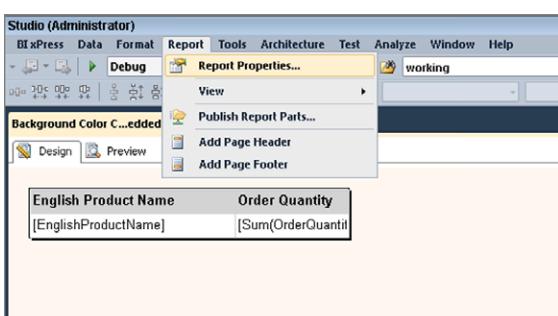
You must input the embedded custom code on each report for which you want to use the code. For example, if you have written code that helps make your report formatting consistent, then you must add the code to every report that you wish to share that formatting.

When you are ready to input your code, go to the Report Properties window. You can find the Report Properties window two different ways. The first method is to right-click outside the design surface of a report and select Report Properties, as shown in Figure 8-1.



**Figure 8-1**

The second method for navigating to the Report Properties window is found within the toolbar, as shown in Figure 8-2. From the toolbar, select Report > Report Properties.



**Figure 8-2**

**WARNING** Sometimes the Report menu is hidden when other windows are active. For example if you are looking at the properties of a text box in your report, then the Report menu in the toolbar is hidden. If you do not see the Report menu in the toolbar, then first select somewhere in the background of your report and it should appear.

Either way you choose to get to the Report Properties window, you will get the same result. After the Report Properties window opens, select the Code page where you can enter your custom code, as shown in Figure 8-3. The following section walks you through an example of embedding custom code.

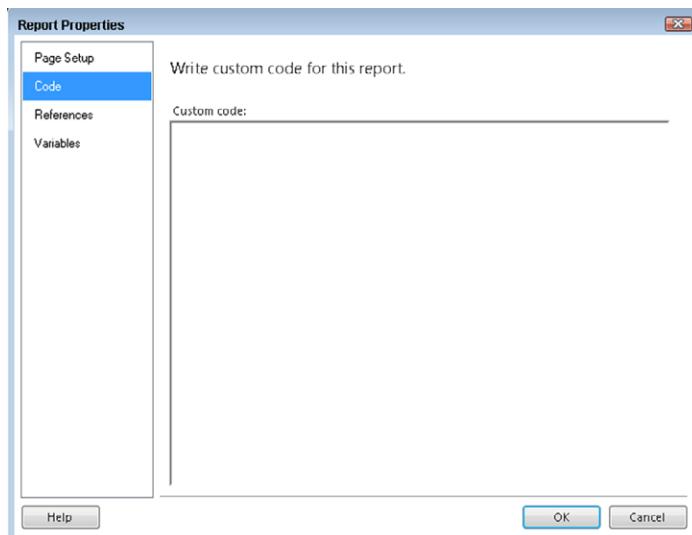


Figure 8-3

**WARNING** This is where you'll need to use your knowledge of Visual Basic (VB). As mentioned previously, VB is currently the only acceptable coding language for embedding code in a report. This is not likely to change in future versions of SQL Server either, so you need to get comfortable with VB before proceeding.

## Writing a Function

In the first example, you will learn how to embed custom code to do something that is possible with native SSRS expressions, but would require complex nested logic that can be much more confusing. You begin with a very simple report that lists all products and the sales for each, as shown in Figure 8-4. This base report and the completed report, Extending SSRS, can be found on the download page for this book at [www.wiley.com/go/performancedashboards](http://www.wiley.com/go/performancedashboards).

English Product Name	Order Quantity
All-Purpose Bike Stand	249
AWC Logo Cap	2,190
Bike Wash - Dissolver	908
Classic Vest, L	195
Classic Vest, M	199
Classic Vest, S	168
Fender Set - Mountain	2,121
Half-Finger Gloves, L	443
Half-Finger Gloves, M	499
Half-Finger Gloves, S	488
Hitch Rack - 4-Bike	328
HL Mountain Tire	1,396
HL Road Tire	858
Hydration Pack - 70 oz.	733
LL Mountain Tire	862
LL Road Tire	1,044
Long-Sleeve Logo Jersey, L	452
Long-Sleeve Logo Jersey, M	442

**Figure 8-4**

The requirement for this example is to change the background color of the Order Quantity column based on the data values. The users would like the following:

- Order quantity less than 500 to show maroon
- Order quantity less than 1,000 to show yellow
- Order quantity less than 2,000 to show orange
- Order quantity greater than or equal to 2,000 to show green

You could accomplish this using the Reporting Services expression language with several nested IIF statements, but to make any changes to the thresholds you would have to alter every cell that has had the formatting applied to it. Using a function, you can simply go back to the code window to adjust the custom code and see the immediate results next time you preview the report. To input the function, navigate to the Report Properties window and open the Code page. Use the following code to satisfy the requirement:

```
Public Shared Function SetColor(ByVal Value as Integer) As String
    SetColor = "Green"
    If Value < 500 Then
        SetColor = "Maroon"
    ElseIf Value < 1000 Then
        SetColor = "Yellow"
    ElseIF Value < 2000 Then
        SetColor = "Orange"
    End If
End Function
```

This code accepts a parameter value and compares that value to return the appropriate color in the report. The thresholds for each color are set per the user requirements and a default value of Green is set to return for all others. The Code page should look like Figure 8-5 before you click OK and return to the report.

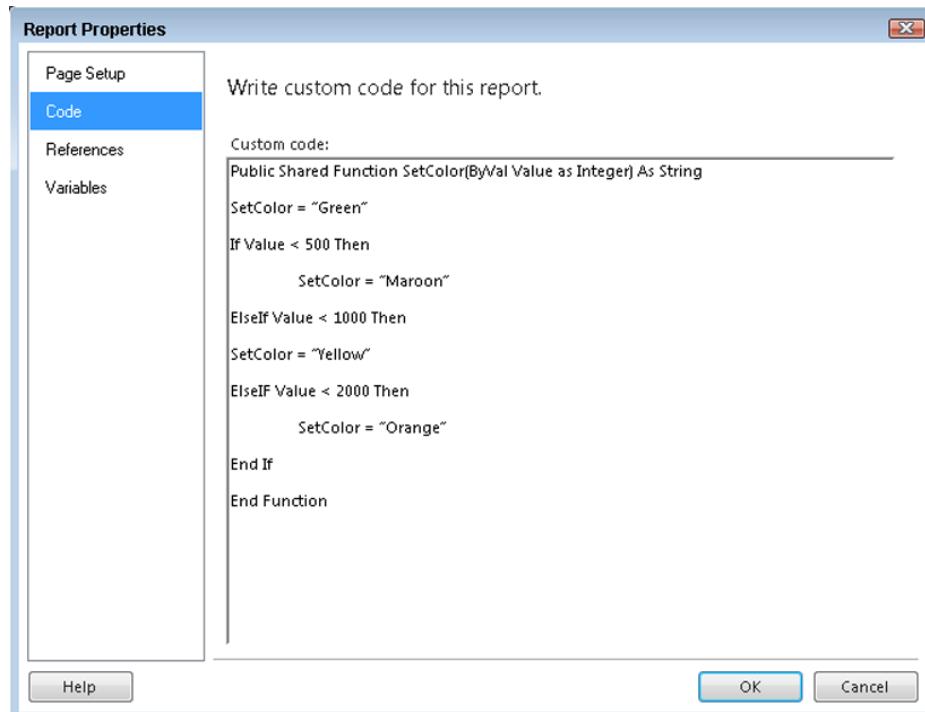
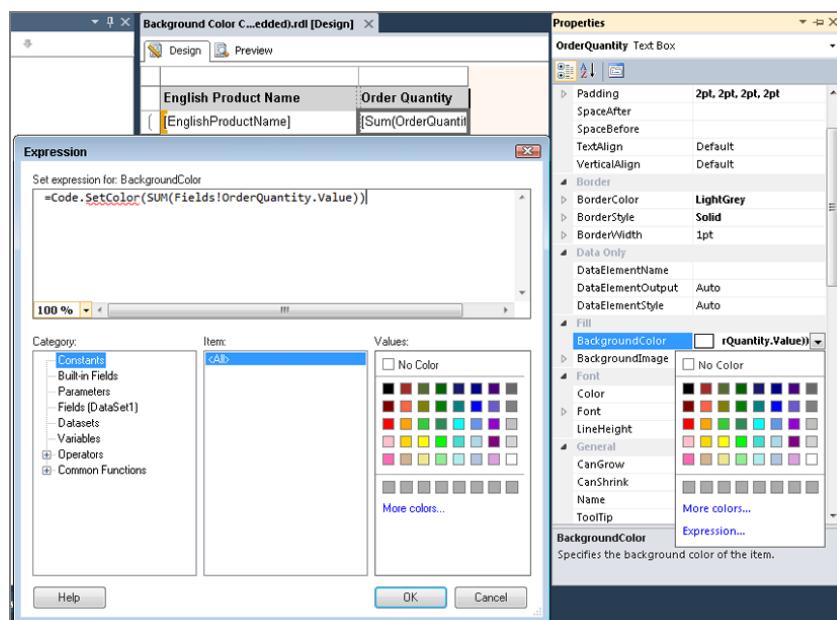


Figure 8-5

With the custom code set, it is now time to actually call the code in the background property of the cell that you would like to format.

1. Select the cell to which you'd like to apply the formatting and then open the Properties menu (press F4).
2. Find the `BackgroundColor` property and then select Expression to make the property value dynamic.
3. In the Expression window, use the following code to call the custom code and pass in the value from the `OrderQuantity` field, as shown in Figure 8-6:  
`=Code.SetColor(SUM(Fields!OrderQuantity.Value))`

4. Once these changes are complete, click OK and preview the report. Your completed report should look similar to Figure 8-7.



**Figure 8-6**

English Product Name	Order Quantity
All-Purpose Bike Stand	249
AWC Logo Cap	2,196
Bike Wash - Dissolver	908
Classic Vest, L	195
Classic Vest, M	193
Classic Vest, S	168
Fender Set - Mountain	2,121
Half-Finger Gloves, L	441
Half-Finger Gloves, M	499
Half-Finger Gloves, S	483
Hitch Rack - 4-Bike	326
HL Mountain Tire	1,396
HL Road Tire	858
Hydration Pack - 70 oz.	733
LL Mountain Tire	862
LL Road Tire	1,044
Long-Sleeve Logo Jersey, L	454
Long-Sleeve Logo Jersey, M	441

**Figure 8-7**

## Creating Assemblies for Reusability

---

If you are looking for consistency in how your developers write reports, custom code is a great way of accomplishing it. The best long-term solution for achieving consistent standards in report development is through external assemblies. External assemblies can be shared with multiple report developers to standardize how reports are developed. After creating and deploying an assembly, Reporting Services will reference the code externally, enabling the developer to manage the code separately from the reporting tool. Any time a change needs to be made to the code, you simply redeploy the assembly and all the reports using it will automatically pick up the changes. Any custom code is compiled in Visual Studio without the possibility of someone that is less familiar with .NET making changes with embedded code.

**NOTE** You may have noticed that last statement was less specific about the coding language that can be used. As stated previously, with embedded code you can only use VB. With external assemblies, however, you are no longer limited to VB; you are free to develop your custom code using either VB or C#, because the code you develop with external assemblies is done in Visual Studio and not Reporting Services.

**WARNING** The Visual Studio shell that is installed with SQL Server does not have all the components needed to develop assemblies. To create assemblies you should install the full Visual Studio package separately.

External assemblies are also a great way to manage custom code. You aren't forced to open the report to make changes to the code, because the report simply creates a reference to where the assembly is stored. This is great, because you can make a change to the assembly, redeploy it, and affect hundreds of reports that use the code reference.

Of course, there are disadvantages to using external assemblies, too. For example, it can be very tedious to deploy an assembly. The steps for deploying are described in the section "Deploying an Assembly," but for now understand that it is much more time-consuming than using the embedded custom code method described previously. External assemblies also have restricted access to system resources. Therefore, if you anticipate the assembly using the filesystem or accessing outside data, there are some configuration settings that must be changed.

Table 8-2 provides a summary of the pros and cons of using external assemblies.

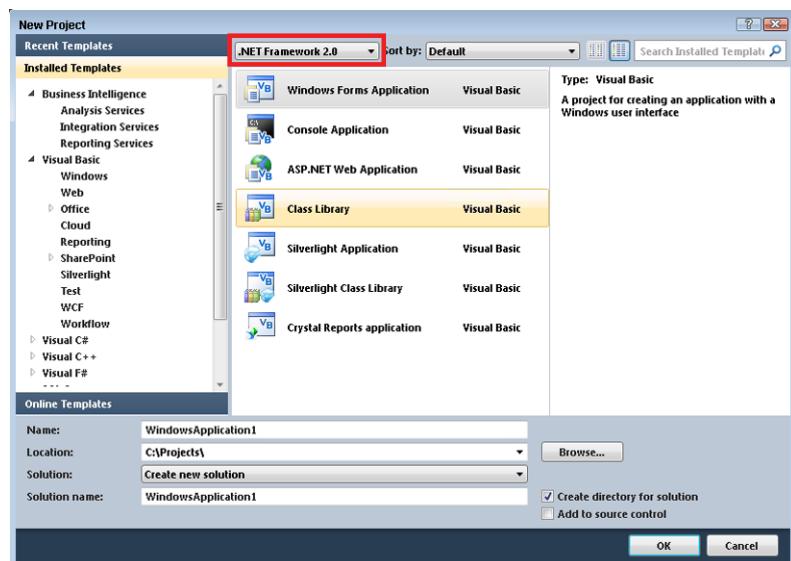
**Table 8-2:** Pros and Cons of External Assemblies

PROS	CONS
Can be written in any .NET language.	Deployment is tedious.
Updates to code can be managed from outside SSRS.	Assemblies will have restricted access to system resources.
Best way to standardize custom code by sharing .dll files with other report developers.	

## Building a .NET Assembly

After you have weighed the pros and cons of using embedded custom code and building an external assembly and have decided to go with building an assembly, your next step is to walk through creating an assembly. This section uses the same example scenario used for embedded code in which you formatted the background color based on the value of the Order Quantity field.

1. Open Visual Studio and create a new `Class Library` project named `ColorFormat`, as shown in Figure 8-8. Before you press OK, ensure that the project is created using .NET Framework 2.0, otherwise Reporting Services will not accept the assembly.

**Figure 8-8**

- Once the project is created, rename the `Class1.vb` file in the Solution Explorer to `ValueFormat.vb`. Open the `ValueFormat.vb` file if it is not already open. Use the same code from the previous example and place it between the existing code, as shown in Figure 8-9:

```
Public Class ValueFormat
    Public Shared Function SetColor(ByVal Value As Integer) As String
        SetColor = "Green"

        If Value < 500 Then
            SetColor = "Maroon"

        ElseIf Value < 1000 Then
            SetColor = "Yellow"

        ElseIf Value < 2000 Then
            SetColor = "Orange"
        End If

    End Function
End Class
```

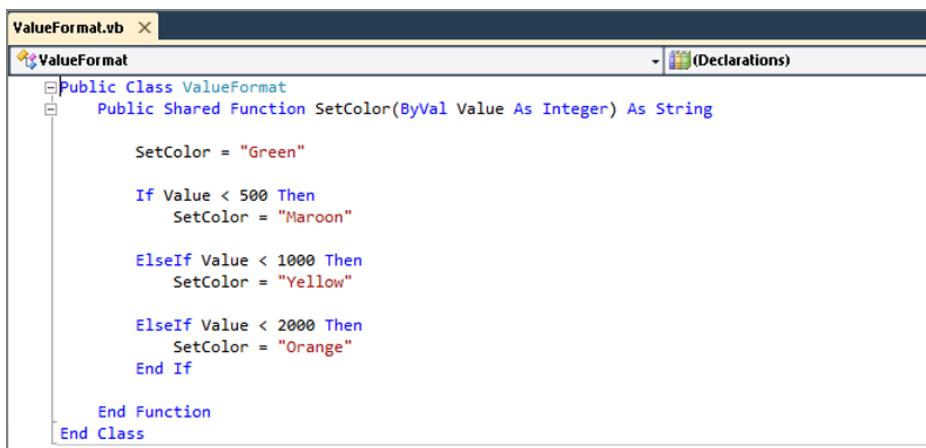


Figure 8-9

- When you are ready to register the `.dll` to the Global Assembly Cache (GAC), set the `<Assembly: AllowPartiallyTrustedCallers()>` property. Without this declaration, applications like Reporting Services are unable to use the assembly. To make this change you must navigate to the Solution Explorer and with the project selected, click the button to Show All Files, as shown in Figure 8-10. Once all the project files are visible, expand `My Project` and open the `AssemblyInfo.vb` file.

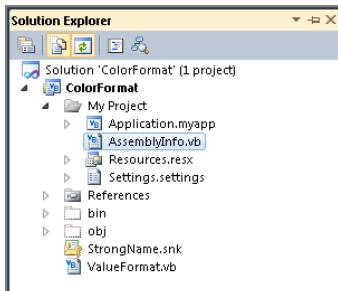


Figure 8-10

- With the AssemblyInfo.vb file open, add the Imports System.Security namespace and the <Assembly: AllowPartiallyTrustedCallers()> declaration, as shown in Figure 8-11.

```

Imports System
Imports System.Reflection
Imports System.Runtime.InteropServices
Imports System.Security

' General Information about an assembly is controlled through the following
' set of attributes. Change these attribute values to modify the information
' associated with an assembly.

' Review the values of the assembly attributes

<Assembly: AssemblyTitle("ColorFormat")>
<Assembly: AssemblyDescription("")>
<Assembly: AssemblyCompany("")>
<Assembly: AssemblyProduct("ColorFormat")>
<Assembly: AssemblyCopyright("Copyright © 2012")>
<Assembly: AssemblyTrademark("")>
<Assembly: AllowPartiallyTrustedCallers()>
<Assembly: ComVisible(False)>

```

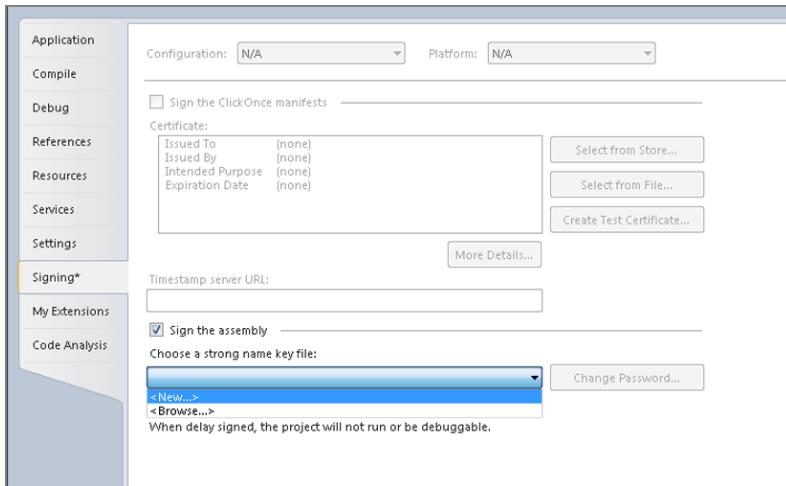
Figure 8-11

- Once these changes are complete, save and close the AssemblyInfo.vb file.

The last change you need to make before this assembly is complete and ready for deployment is to sign it with a *strong name*. A strong name is basically a way of naming or versioning the dll. If you do not give the dll a strong name, it cannot be registered in the GAC.

- To give the dll a strong name, right-click the project name in the Solution Explorer and select Properties.
- Go to the Signing page and check Sign the assembly.

- Create a new strong name key file by selecting <New...> from the dropdown box, as shown in Figure 8-12.



**Figure 8-12**

- Name the file `StrongName` and uncheck `Protect my file with a password`, then click `OK`.
- Once this is complete, Save All and perform a build of the project. You can build the project by selecting the Build menu from the toolbar and then Build `ColorFormat`.

With the assembly created, you can now learn how to properly deploy it. From here on, you are done with Visual Studio.

## Deploying an Assembly

Complete the following steps to deploy an assembly for Reporting Services' use.

- Copy the `.dll` you created to the following two locations:
  - `C:\Windows\assembly`
  - `C:\Program Files\Microsoft SQL Server\MSRS11.MSSQLServer\Reporting Services\ReportServer\bin`
- Navigate to the `bin\Debug` folder in the path where the Class Library project was created.
- In the `Debug` folder you should see the `ColorFormat.dll` file. Copy and paste this file to the `C:\Program Files\Microsoft SQL Server\MSRS11.MSSQLServer\Reporting Services\ReportServer\bin` folder.

4. Place the `dll` in the GAC. This cannot be done by copying and pasting the file. It must be dragged and dropped into the `C:\Windows\assembly` folder.

Once this is complete you will see the `dll` appear in the GAC as shown in Figure 8-13.

Assembly Name	Version	Cul...	Public Key Token	Process...
Accessibility	2.0.0.0		b03f5f7f11d50a3a	MSIL
ActiproSoftware.Sha...	1.0.104.0		36ff2196ab5654b9	MSIL
ActiproSoftware.Syn...	4.0.286.0		21a821480e210563	MSIL
ActiproSoftware.Syn...	4.0.286.0		21a821480e210563	MSIL
ActiproSoftware.Syn...	4.0.286.0		21a821480e210563	MSIL
ActiproSoftware.Wi...	1.0.104.0		1eba893a2bc55de5	MSIL
ADODDB	7.0.330...		b03f5f7f11d50a3a	
AgeAssembly	1.0.0.0		19652d9cd300b788	MSIL
AuditPolicyGPMana...	6.1.0.0		31bf3856ad364e35	x86
AuditPolicyGPMana...	6.1.0.0		31bf3856ad364e35	AMD64
BDATunePIA	6.1.0.0		31bf3856ad364e35	x86
BDATunePIA	6.1.0.0		31bf3856ad364e35	AMD64
ChilkatDotNet2	9.0.4.0		eb5fc1fc52ef09bd	x86
ChilkatDotNet2	9.0.4.0		eb5fc1fc52ef09bd	AMD64
ColorFormat	1.0.0.0		2659107ae1bcd167	MSIL
ColorFormatSeptem...	1.0.0.0		672ef0303c4f4081	MSIL
ColorFormatting	1.0.0.0		2feb5d1add29065b	MSIL
ComSvcConfig	3.0.0.0		b03f5f7f11d50a3a	MSIL
Crypto	1.6.1.0		47acf905d0337c39	MSIL
Icscompmgd	8.0.0.0		b03f5f7f11d50a3a	MSIL
CustomMarshalers	2.0.0.0		b03f5f7f11d50a3a	x86
CustomMarshalers	2.0.0.0		b03f5f7f11d50a3a	AMD64
DeployLX.Licensing...	4.1.200...		798276055709c98a	MSIL

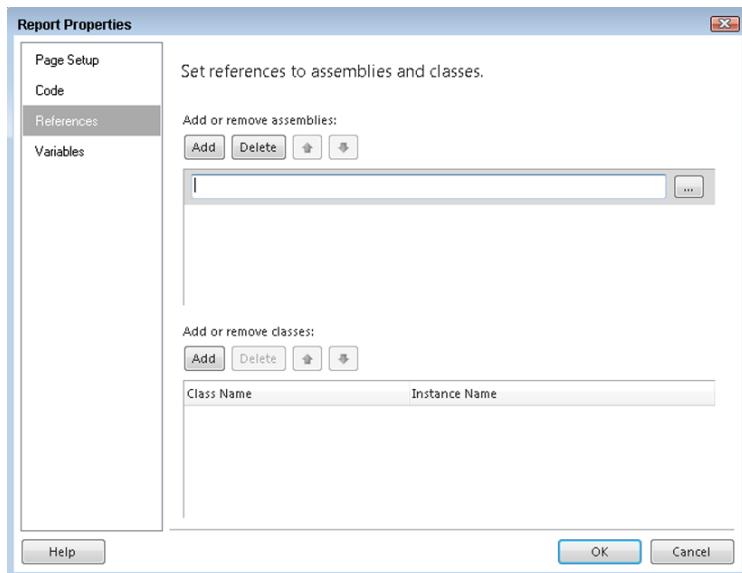
Figure 8-13

## Using an Assembly in a Report

With the `dll` creation and deployment completed, the assembly should now be ready to use. To start using it, open the designer for Reporting Services with the base report with which you started in the first example and complete the following steps. Remember, all of these examples can be found on the downloads page of [www.wiley.com/go/performancedashboards](http://www.wiley.com/go/performancedashboards) with the beginning and completed examples.

1. Find the Report Properties menu that you learned about in the first section of this chapter. With the Report Properties menu open, select the References page. This is different from the embedded code example where you just started typing the code on the Code page. The Reference page is where you will point the report to any assemblies that you previously created.

2. Click Add, under Add or remove assemblies and then select the ellipsis next to the new entry, as shown in Figure 8-14.



**Figure 8-14**

3. On the Browse tab of the Add Reference window, select the `ColorFormat.dll` then click OK. Although you have selected the `.dll` from the `C:\Program Files\Microsoft SQL Server\MSRS11.MSSQLServer\Reporting Services\ReportServer\bin` folder, it will actually load the GAC version when the report runs. Also, notice now that the reference appears on the Reference page of Report Properties. Click OK again to return to the report designer.
4. As you did in the embedded code example, select the cell to which you want to apply the formatting and then open the Properties menu (F4).
5. Find the `BackgroundColor` property then select Expression to make the property value dynamic.
6. In the Expression window, use the following code to call the assembly and pass in the value from the `OrderQuantity` field as a parameter (shown in Figure 8-15):
  - **Template:**  
`=AssemblyName.ClassName.FunctionName(Parameter)`
  - **Code for this example:**  
`=ColorFormat.ValueFormat.SetColor(SUM(Fields!OrderQuantity.Value))`

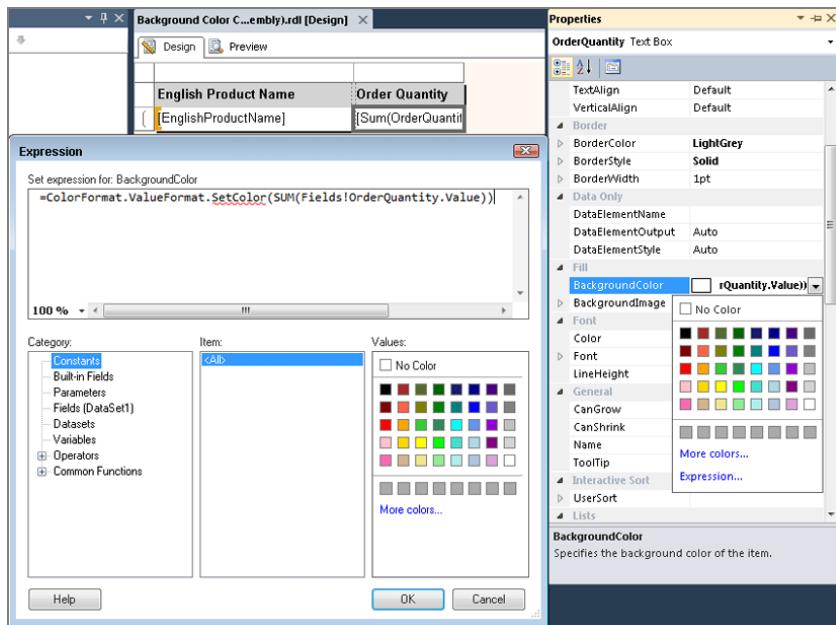


Figure 8-15

- Once the expression is entered, click OK and preview the report.

The report should look exactly the same as the embedded code example, but now the effect is achieved using an external assembly. If you ever need to modify the color thresholds, you would modify the Class Library project and redeploy the d11. This is helpful because if the same d11 is used in several reports, then you can change the report appearance without having to open a single rdl report file.

## Summary

---

This chapter covered how to use custom code to extend the functionality of Reporting Services. Using custom code enables you to standardize coding practices among developers that may try and accomplish the same goal using multiple methods. While the direct embedding of custom code is quick and simple to do, it does not scale nearly as well as an external assembly when multiple reports use the same functions. An external assembly is a little more cumbersome to create and maintain, but it is the ideal solution because you are not limited by .NET language, and it can be managed from outside of Reporting Services.



# Publishing Dashboards and Balanced Scorecards

The hard work is done. You have tirelessly designed a dashboard and a balanced scorecard solution that fits all the necessary requirements and you are ready to share the results with others. Now that you have completed the reports, you must publish them to a central repository to which others have access. You can do this by deploying your report files (.rdl) to either a native Report Server or to SharePoint. This chapter discusses publishing reports to both the native Report Server and to SharePoint and also details what to do with reports once they are there. This includes managing security to grant others access to the reports, scaling out your Report Server to assist with report processing, and planning an organization strategy for your Report Server.

## Choosing a Deployment Location

---

With two Reporting Services report deployment location options, you must consider which is best for your organization. There is no clear-cut choice, so it may help to get a better understanding of each.

SharePoint is Microsoft's collaborative software that often serves multiple purposes for companies. Some use it as their internal intranet site, some use it for document sharing, and others utilize it for its workflow capabilities. SharePoint

also contains many components that interact with Business Intelligence, such as Reporting Services Integration. While having a Report Server integrated with SharePoint creates a one-stop shop for all of a company's needs, it can come at a hefty price. Consider that SharePoint uses a Server/CAL (Client Access License) model, which is an additional cost on top of the licensing required for SQL Server. This compares to the native Report Server front end, which requires no additional licensing cost other than what you already paid for SQL Server.

Another major consideration is features that are unsupported when running in SharePoint integrated mode. Key features like Linked Reports, My Reports, and My Subscriptions are not operational when running in integrated mode. For some companies, losing these features can be a deal breaker when considering migrating from a native Report Server to SharePoint integrated configuration. If My Reports is a critical feature that users utilize to develop their own reports and store them in a personal repository, then you can quickly eliminate SharePoint as an option. Although some features are unsupported, you will find that additional tools, such as Data Alerts and Power View, are available when using a SharePoint integrated topology.

The choice of whether you deploy to a native Report Server or SharePoint is rarely the decision of the report developer, but understanding the differences is key. Although you may not have the purchasing power to make the decision to go with the more expensive option of SharePoint, you certainly are an influencer who can provide key details on the benefits received by utilizing the Reporting Services integrated approach with SharePoint.

## **Configuring and Publishing to a Native Report Server**

---

If you decide that a native report server is the best deployment location for you, you need to properly configure a Report Server prior to deploying your dashboard and balanced scorecard reports. This section explains how to do this, and also walks you through techniques for implementing a scaled-out architecture, which will help balance the workload required to process reports. Finally, you will learn the steps for deploying your reports to a native Report Server.

### **Setting Up and Configuring Reporting Services**

Reporting Services is unique in the SQL Server stack in that it has its own devoted configuration tool to set up each component of the service. The Reporting Services Configuration Manager, shown in Figure 9-1, assists with configuring the following settings:

- Service Account
- Web Service URL

- Database
- Report Manager URL
- E-mail Settings
- Execution Account
- Encryption Keys
- Scale-out Deployment

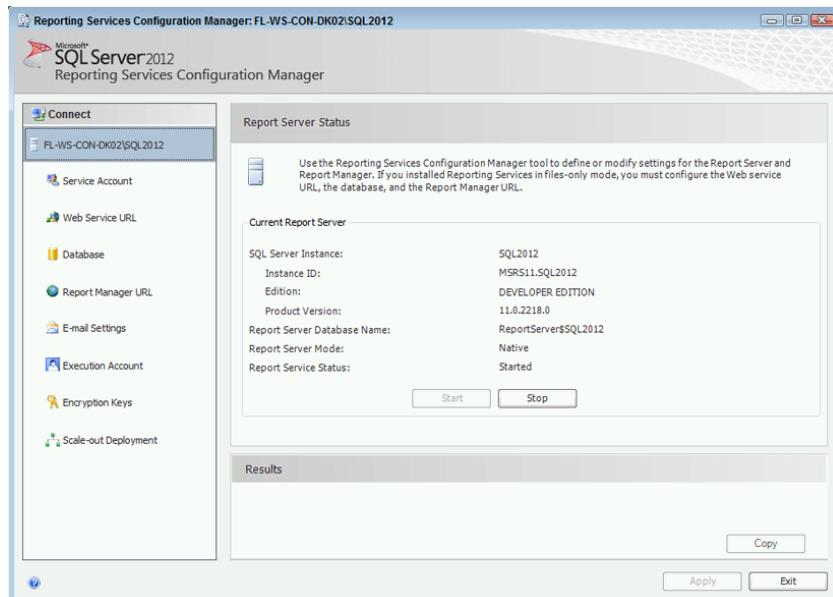


Figure 9-1

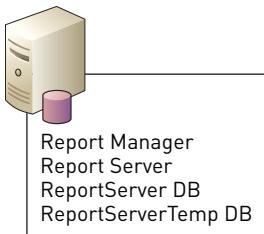
Several of these features are optional and they often do not need to be modified after the initial install, but others are more critical and will require your attention. The following list discusses each of these options and places emphasis on those that require you to take action:

- The Reporting Services service account is automatically configured during install, but if you need to make a change this section describes how to do it.
- The Web Service URL section is the location from which you deploy all your reports. If you installed Reporting Services without configuring it with a URL, you would have to assign one here. This is also where you can associate any SSL certificates on the server with the Report Server. *Secure Socket Layer (SSL) certificates* are what companies use to secure access to their websites. The URL for this type of site starts with `https` instead `http`.

- On the Database page you find the location of the database repository for all Reporting Services activity and metadata. This is also one of the pieces of information that you must change to go from a native Reporting Services instance to SharePoint integrated. If you installed Reporting Services without letting the installer configure it, you would now identify a database location.
- The Report Manager URL is the web front end and management portal for Reporting Services. If you installed Reporting Services without configuring it, then you must assign a URL. The same SSL certificate properties that were available in the Web Service URL are also available here by clicking the Advanced button.
- If you plan on using Reporting Services subscriptions to e-mail reports to your users, then you must configure the E-mail Settings page. The configuration is basic. You simply provide your SMTP server name and an e-mail address that you wish to be the sender.
- The Execution Account is an optional setting that is rarely needed. If you enable report data sources that do not require credentials or you need to connect to remote files, then you might need to set up this account.
- One of the most important setting pages in the entire Reporting Services Configuration Manager is the Encryption Keys page. This page is devoted to preparing and recovering from a disaster scenario. It is needed because inside the `ReportServer` database all the connection string information is encrypted, preventing someone from searching the database for login information used in connections. If you ever have to restore the `ReportServer` database, you will need the encryption key to restore all those encrypted connection strings. Immediately following the install of Reporting Services you should open the Configuration Manager, run a backup of the encryption key, and store it in a safe place.
- Last is the Scale-out Deployment feature, which is discussed in detail in the next section of the chapter.

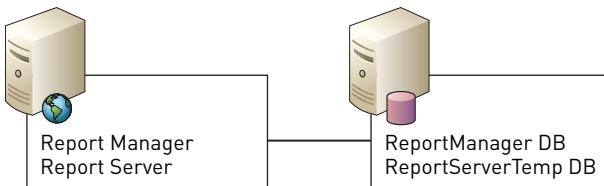
## Planning for Growth and Scalability

Many small companies that are just beginning with Reporting Services will often start with a single server for managing all their needs. An example of this is shown in Figure 9-2, which illustrates a single server that acts as both web server and database server. For small companies with very few users this may be acceptable, but when you begin to feel the performance degradation of an increased user base, you might need to take a large company strategy with your architecture.



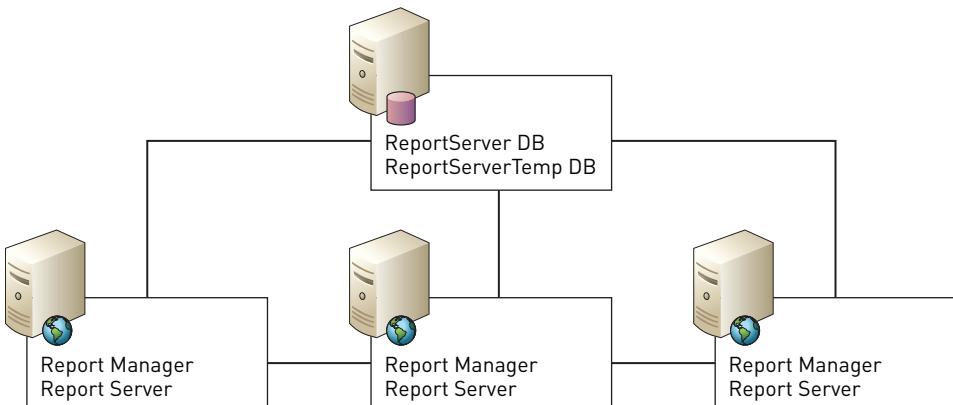
**Figure 9-2**

The first step you can take to relieve resource pressure is to split the database and Report Servers onto their own machines, as shown in Figure 9-3.



**Figure 9-3**

Most companies that anticipate a large user base start scaling out their environment by creating a web farm of Report Servers. The goal of using a scaled-out Reporting Services architecture is to split the workload required for rendering reports to your users. Figure 9-4 shows an example architecture that splits the report processing work across three web servers. In a typical web farm architecture, these three web servers would share a single database, which is used as a repository for report metadata.



**Figure 9-4**

This type of architecture plan is often implemented in combination with a *network load balancer* that is tasked with sending users to the most underutilized server. Figure 9-5 shows the final architecture including the user making a request that the network load balancer distributes to the appropriate server.

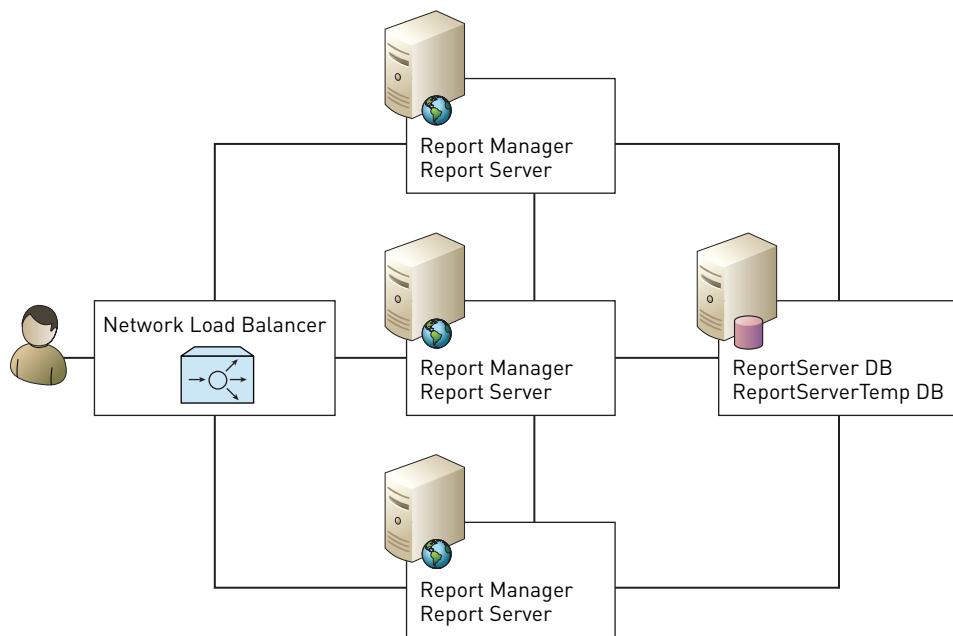
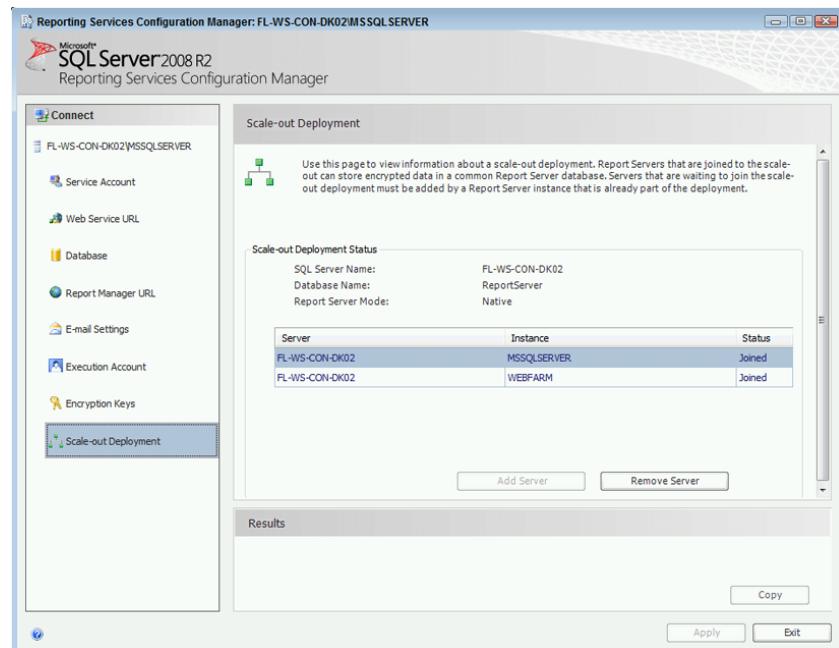


Figure 9-5

To create a simple Reporting Services web farm like this, follow these steps:

1. Install the SQL Server database engine on one server.
2. Install Reporting Services on two servers and do not configure them.
3. Configure one of the Report Servers using the Reporting Services Configuration Manager. Create the ReportServer databases on the database engine you created in step one.
4. Configure the second Report Server to use the same ReportServer database that was created during the configuration of the first server. Assign Report Server and Report Manager URLs to the new server.
5. Go back to the first Report Server you configured and in the Reporting Services Configuration Manager, click Add Server under Scale-out Deployment to join the instances, as shown in Figure 9-6.



**Figure 9-6**

## Deploying to SQL Server Reporting Services

Whether you are deploying to a native Report Server or a SharePoint integrated instance of Reporting Services, your method for publishing reports will be very similar. After completing the development of your reports in SQL Server Data Tools, there is a simple process for setting up the environment for deploying.

To configure SQL Server Data Tools for deploying to your Report Server, follow these steps:

1. With SQL Server Data Tools open to the Report Project you wish to deploy, right-click the project file, and select Properties in the Solution Explorer.
2. On the General page, type in the appropriate TargetServerURL and TargetReportFolder, as shown in Figure 9-7. The TargetServerURL is typically a path similar to the following: `http://<servername>/ReportServer`. Click OK.
3. Right-click the Report Project file again and select Deploy to send the files in your project to the server location you specified in the previous step.

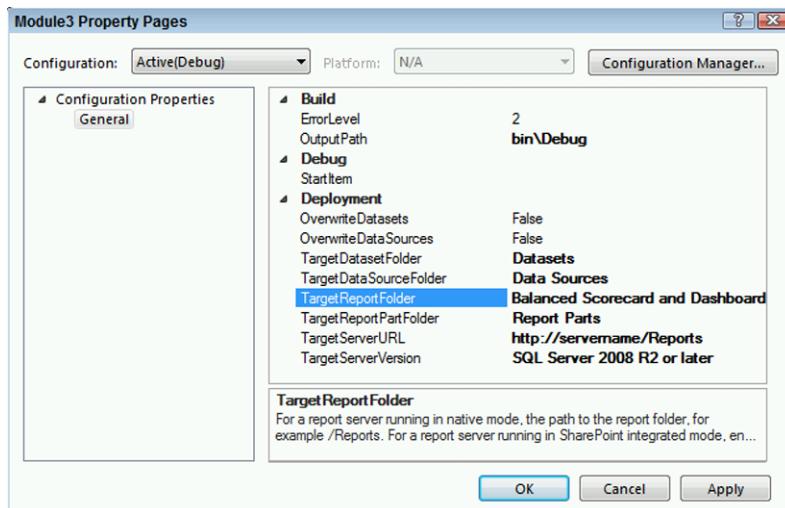


Figure 9-7

## Managing Security

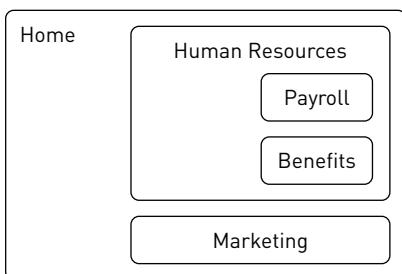
Once your reports have been published to the Report Server, you need to ensure users can access them. Reporting Services uses a role-based security model to enable users to perform tasks on the server. There are two types of roles: system-level and item-level. System-level roles enable you to grant site-wide actions. Item-level roles apply security to individual objects on the report server, such as reports that have been deployed. Item-level roles need to be changed most frequently so it's best to focus your attention there first.

When an item-level role is assigned to a user, the user is permitted to perform the tasks that are grouped under the selected role. Item-level roles can be assigned to individual reports or to the folder level. When security is assigned at the folder level all the objects inside of it inherit the folder's permissions by default. Reporting Services' security is cumulative, so if a user is assigned permission to multiple folders or multiple roles, he would have the rights to all of these assignments.

When planning the security for your Report Server, consider these recommendations to limit the maintenance needed:

- Avoid applying security to individual reports. Instead, opt for managing all security at the folder level. Once you start granting permission at the report level, it breaks the inheritance of folder-level security, and it is difficult to go back to managing the higher level folders.

- Organize your folders by how you intend to secure them. Figure 9-8 shows a Report Server that uses folders in a manner that would likely follow the company's security needs too.
- Avoid assigning individual users access. Instead, opt for using groups. Using Figure 9-8 as an example, you could create Active Directory groups that store all the user accounts you desire to have access to Benefits in one group and a second group of all the users you want to have access to Payroll. The benefit of this is you would not have to ever modify your Reporting Services security after the initial create, because you are simply adding people to existing Active Directory groups, which can be done externally from Reporting Services.



**Figure 9-8**

Reporting Services provides several default item level roles to use. Table 9-1 is a list of the default roles and a description of what they do.

**Table 9-1:** Reporting Services Default Item Roles

PREDEFINED ROLE	DESCRIPTION
Content Manager Role	Includes all item-level tasks. Users who are assigned to this role have full permission to manage Report Server content, including the ability to grant permissions to other users, and to define the folder structure for storing reports and other items.
Publisher Role	Users who are assigned to this role can add items to a Report Server, including the ability to create and manage folders that contain those items.
Browser Role	Users who are assigned to this role can run reports, subscribe to reports, and navigate through the folder structure.

*Continued*

**Table 9-1** (continued)

PREDEFINED ROLE	DESCRIPTION
Report Builder Role	Users who are assigned to this role can create and edit reports in Report Builder.
My Reports Role	Users who are assigned to this role can manage a personal workspace for storing and using reports and other items.

**CREATING CUSTOM ROLES**

The roles in Table 9-1 should solve the majority of your security problems, but in the rare cases they don't, it is possible to create custom roles. For example, if a default or combination of default roles cannot do what you find necessary to secure the report server item, you could create a custom role to do so. Custom roles can be created by logging on to your Report Server through SQL Server Management Studio, expanding the Security node, and right-clicking on Roles to select New Role.

System-level roles differ from item-level roles in that they only help manage security at a site-wide level. The only times you will find it necessary to add someone to a system-level role is if they are either an administrator of your Report Server or the user needs to use the Report Builder development tool. System-level security is applied under Site Settings in the Report Manager interface. System-level security also has default roles available, which are listed in Table 9-2.

**Table 9-2:** Reporting Services Default System Roles

PREDEFINED ROLE	DESCRIPTION
System Administrator	View and modify system role assignments, system role definitions, system properties, and shared schedules.
System User	View system properties, shared schedules, and enable use of Report Builder or other clients that execute report definitions.

**NOTE** To give users the ability to launch Report Builder from the Report Manager they would need both Report Builder and System User roles.

To get a better idea of all of these roles and how they work, take a look at the following example of a common security scenario that would require you to modify security. Imagine you have a new user that should have permission to view all reports. You want to leverage the inheriting folder structure that

Reporting Services natively has available. To ensure the user gets permissions to all current reports and to any new reports that may get added later you need to add the user as a Browser to the Home folder, which will automatically be granted access to all subfolders by default. To grant a user access to a folder, follow these steps:

1. Open a web browser and type in the Report Manager URL for your server. This is typically in the format of: `http://<servername>/Reports`.
2. Once the Report Manager opens, click the Folder Settings button under the Home folder. Home is the root folder.
3. To add new user permissions on the folder, click New Role Assignment.
4. Provide a username or group name and select Browser to give view-only permissions to the Home folder. This step is shown in Figure 9-9. Click OK. Remember, Reporting Services uses an inheriting security model, which means any folder created under Home will automatically have the same security applied to it unless you break the inheritance.



Figure 9-9

As you can see, applying Reporting Services security can be fairly simple. The most important aspect of security is the organization of your security model. Use the tips you learned in this section for keeping your security model simple and maintainable over time.

## Configuring and Publishing to SharePoint

The other location you can deploy your report to is SharePoint. This section explains how to properly configure a SharePoint integrated Report Server. You also walk through the steps for deploying your reports to a SharePoint integrated Report Server. Finally, you will see how a SharePoint-only feature called Data Alerts helps users know when changes have occurred on their balanced scorecards and dashboards.

## Configuring Reporting Services for SharePoint

If you have decided to go with a Reporting Services environment that integrates with SharePoint, then you will find that the configuration is more complex but only involves a one-time setup. Following the installation of both SQL Server and SharePoint, there are a few services that need to be configured before you see the full benefit of both servers working together. The objects that need your attention after installation are the following:

- Creating a Reporting Services service application
- Provisioning subscriptions and alerts
- Configuring e-mail for the services application
- Adding Reporting Services content types to a library

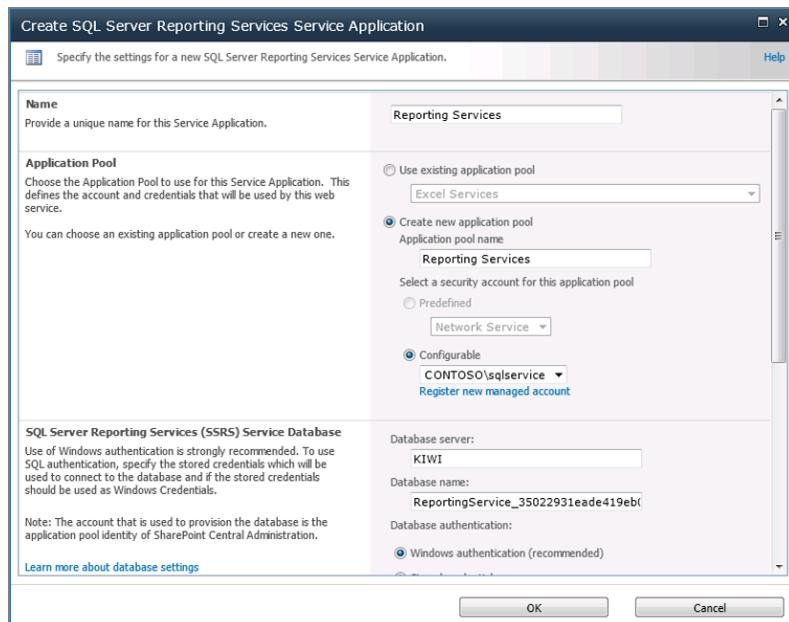
The following sections focus on how to configure each of these appropriately.

### *Creating a Reporting Services Service Application*

A major change in SQL Server 2012 was how Reporting Services integration with SharePoint is designed. When integrated with SharePoint, Reporting Services now uses a service application for administration called SQL Server Reporting Services Service Application. *Service applications* are a term to describe the model of deployment for shared services in SharePoint 2010. Much of the configuration of the service can be found in the service application.

Service applications can be managed from either PowerShell or the SharePoint Central Administration site. To create a new services application for Reporting Services using the SharePoint Central Administration site, follow these steps:

1. Open the SharePoint Central Administration site and select Manage Services Application under the Application Management group.
2. In the SharePoint ribbon, click New and then select SQL Server Reporting Services Application.
3. When the Create SQL Server Reporting Services Service Application page appears, provide a name for the application.
4. Select Create new application pool and provide a name. This is often configured with the same name as the service application, as shown in Figure 9-10.
5. Next, provide the database server and database name for where the Reporting Services repository database resides.

**Figure 9-10**

6. In the Web Application Association section, select the web application to be provisioned for access by the current Reporting Services service application. Then click OK.

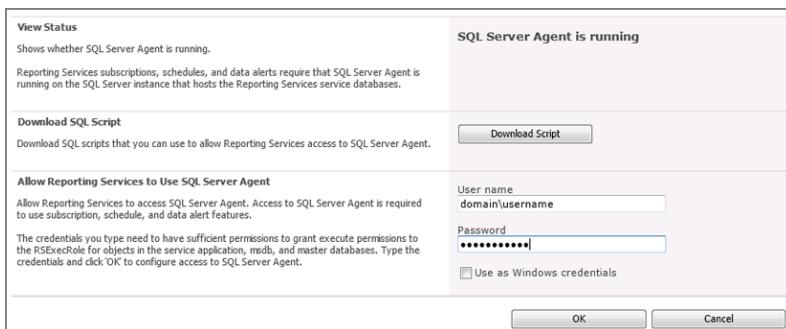
### *Provision Subscriptions and Alerts*

If you use Reporting Services subscriptions or alerts, the scheduling tool that is used is the SQL Server Agent, which is the main scheduling tool used within SQL Server. To ensure Reporting Services can properly communicate with the SQL Server agent, you might have to configure through a step called Provisioning Subscriptions and Alerts.

To provision subscriptions and alerts using the SharePoint Central Administration site, following these steps:

1. Open the SharePoint Central Administration site and select Manage Services Application under the Application Management group.
2. Select the Reporting Services application that you previously created.
3. Click Provision Subscriptions and Alerts to manage the settings.

4. Ensure that the View Status section says, “SQL Server Agent is running.” If not, open the SQL Server Configuration Manager and start it. This should be already running by default.
5. Click the Download Script button, which downloads the script you need to run for Reporting Services to communicate with the SQL Server Agent.
6. Open SQL Server Management Studio and connect to the database instance that has your Report Server database on it.
7. Click the File menu and select Open, then File.
8. Find the SQL script that was just generated and execute it.
9. Back in the SharePoint Central Administration site, provide a username and password meant for SharePoint to communicate with the SQL Server Agent, as shown in Figure 9-11. Click OK.



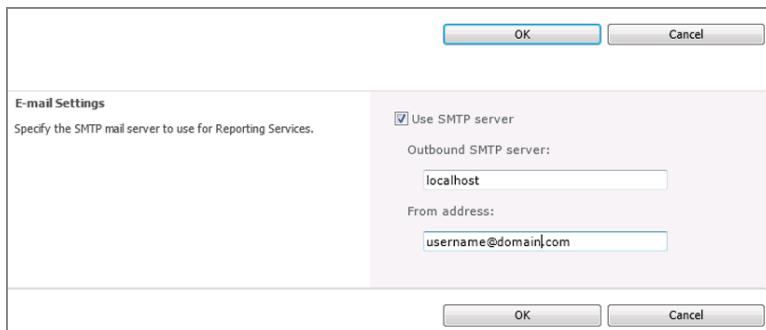
**Figure 9-11**

### Configure E-mail for the Services Application

Later in this chapter in the section “Data Alerts” you will learn how to configure *data alerts*, which notify your users about report data changes. Before you can alert them, however, you must properly configure the e-mail server.

To configure e-mail for a Reporting Services service application using the SharePoint Central Administration site, follow these steps:

1. Open the SharePoint Central Administration site and select Manage Services Application under the Application Management group.
2. Select the Reporting Services application that you previously created.
3. Click E-mail Settings to configure your e-mail server.
4. Select Use SMTP server and provide an Outbound SMTP server.
5. Last, provide an e-mail address in the From address field that will be used as the sender, as shown in Figure 9-12. Click OK.



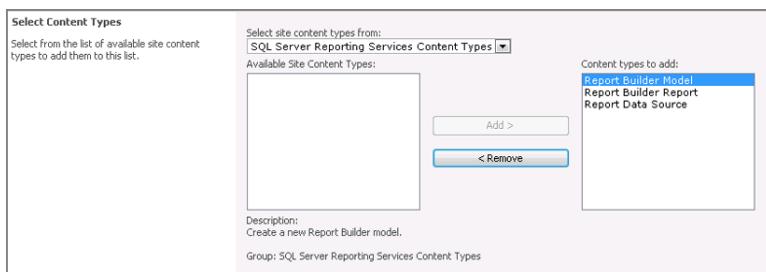
**Figure 9-12**

### Add Reporting Services Content Types to a Library

Once you are ready to begin storing Reporting Services content on your SharePoint site, you will likely want to enable the Reporting Services content types in a library. This enables your users to do things such as launch Report Builder directly from SharePoint, create Report Models, and configure Data Sources.

To add Reporting Services content types to a library, follow these steps:

1. Open the library in which you wish to enable Reporting Services content types.
2. In the SharePoint ribbon, click Library Settings under the Library tab.
3. Select Advanced Settings and ensure that Allow management of content types is set to Yes, then click OK.
4. Click Add from existing site content types.
5. In the Select site content types from dropdown box, select SQL Server Reporting Services Content Types. If this option is not available then either the Reporting Services add-in was not installed or the content types have already been added to this library.
6. Select Report Builder Model, Report Builder Report, and Report Data Source, and click the Add button. Figure 9-13 shows the end result. Click OK when this is complete.



**Figure 9-13**

## Deploying to SharePoint

Earlier in this chapter you learned how to deploy your balanced scorecards and dashboards to a native Report Server. In this section, the focus is on publishing to SharePoint, which has just a few subtle differences.

To configure SQL Server Data Tools for deployment to SharePoint, follow these steps:

1. With SQL Server Data Tools open to the Report Project you wish to deploy, right-click the project file and select Properties in the Solution Explorer.
2. On the General page, type the SharePoint site for the TargetServerURL. The TargetServerURL is typically a path similar to the following:  
`http://<SharePointServer>/.`
3. Next, populate the properties TargetDatasetFolder, TargetDataSourceFolder, TargetReportFolder, and TargetReportPartFolder with a SharePoint library to which you want to deploy your reports and other support objects, as shown in Figure 9-14.
4. Right-click the Report Project file again and select Deploy to send the files in your project to the SharePoint location you specified in the previous step.

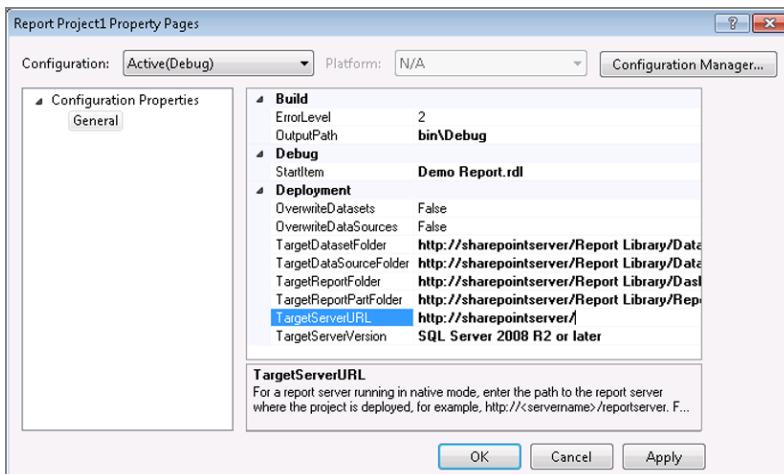


Figure 9-14

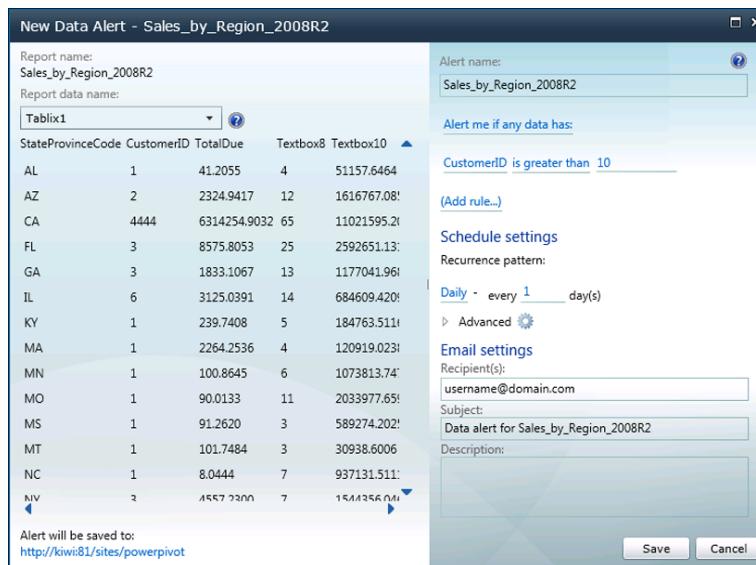
## Data Alerts

Once your dashboards and balanced scorecards have been deployed to a SharePoint library, there is a feature exclusively available in SharePoint 2010 called Data Alerts. Data Alerts is a data-driven alerting system that lets users know about changes in the report data that they care most about.

Consider a busy executive that makes little time for reviewing the dashboards you have developed. From her perspective, the data represented on the report doesn't change frequently enough to warrant her daily attention. Rather than abandoning the report altogether, you might ask her which elements of the report she finds interesting and how she would like to be notified to view the report only when those elements change. The report suddenly becomes more valuable to her because it saves time. This is a small scenario of how data alerts can be used in practice.

To configure Data Alerts on a report deployed to SharePoint, follow these steps:

1. In your web browser, open a Reporting Services report that has been deployed to your SharePoint site.
2. In the top left of the report navigation bar, select the Action drop down option and choose New Data Alert.
3. Create a rule, as shown in Figure 9-15, that triggers an alert after a significant change in the data. This example shows a rule that triggers when the CustomerID column, which is a count of customers, is greater than 10.
4. Next, schedule the alert to run your rule check at the appropriate frequency. This example shows the execution of the alert daily.
5. Finally, provide the e-mail(s) you want to receive the alert then click Save.



**Figure 9-15**

Data Alerts is a nice feature that is provided when you use SharePoint as your reporting hub. The alerts are quick to create but have many benefits from a user perspective in that they make the reports you design more relevant.

## Summary

---

Balanced scorecards and dashboards can be published to either a native Report Server or a SharePoint integrated Report Server. To properly configure each of these environments you must consider your uses. If you plan on having reports delivered to users' e-mails then you must configure the e-mail settings. If you plan on accessing remote files then you need to configure the execution account. You can also architect a scaled-out deployment to improve the performance of your report processing. Reporting Services integrated with SharePoint adds not only the closer integration with the rest of your environment but it also gives you new functionality. The SharePoint feature Data Alerts can help bring attention to reports based on user needs.



# **Dashboards and Balanced Scorecards in Action**

---

## **In This Part**

---

This section of the book focuses on case studies to demonstrate how dashboards, balanced scorecards, and other analytical reports are implemented using SQL Server Reporting Services. The case studies describe fictional companies and are intended to help you learn how these reports are produced from idea to implementation.

Each chapter is dedicated to a specific analytical report type including dashboards, balanced scorecards, process/strategy maps, decision-making charts, and finally, forecasting charts.

**Chapter 10:** Dashboards

**Chapter 11:** Strategy and Process Maps

**Chapter 12:** Balanced Scorecards

**Chapter 13:** Decision-making Charts

**Chapter 14:** Forecasting Trends and Data Smoothing



# Dashboards

In the Parts I and II of the book, you focused on learning and working with the different parts and pieces that make up Reporting Services to deliver a dashboard or balanced scorecard. In this chapter, you'll shift gears to bring together everything you've learned into something concrete and actionable as you build a dashboard for a retail sales company: Adventure Works.

## Case Study: Adventure Works

---

Adventure Works is a well-known international retailer that specializes in bicycles, bicycle components, accessories, and apparel. They have a well-established Internet presence from which they sell directly to consumers and a large reseller network, including specialty bike shops and other retailers that cater to the cycling lifestyle.

In an effort to capture and maintain the recent growth momentum, the executive management team has defined the following strategic objectives:

- Objective 1: Grow net profit on Internet sales
- Objective 2: Improve year-over-year reseller sales
- Objective 3: Grow Internet and reseller channels in the Midwest and southeastern United States

To monitor these strategic objectives, a project has been initiated to build an executive dashboard. The dashboard must capture metrics aligned with the defined objectives. In addition to the strategic objectives, the management team has existing key performance indicators (KPIs) that need to be monitored.

## What to Measure and When

For Adventure Works' executives to gauge performance against their defined strategic objectives, they need a valid and consistent method of measurement. The objectives in this case are fairly straightforward, as they relate directly to either reseller or Internet sales (sometimes both). In practice, this isn't always the case, so it's important to understand what you are measuring and when you measure it.

Table 10-1 breaks down the three objectives by source, measurement area, and interval:

**Table 10-1:** Objective Measurements

OBJECTIVES	SOURCE	MEASUREMENT AREA	INTERVAL
Objective #1	Internet Sales	Sales Amount & Total Product Cost	Daily
Objective #2	Reseller Sales	Sales Amount & Total Product Cost	Daily
Objective #3	Internet & Reseller Sales	Sales Amount & Total Product Cost	Daily

Because this is an executive-level dashboard, the interval of measurement is daily. This is typical for dashboards of which the audience is primarily executives. For performance dashboards, the interval can be more frequent, up to and including real time.

## Developing the Dashboard

Now that the measures and metrics required for the strategic objectives have been identified, it is time to begin building the physical implementation of the dashboard. Before diving into Reporting Services, however, start with a design or layout that has been sketched out on a sheet of paper or put together during a whiteboard session. By laying out the dashboard prior to implementation, it is easier to take into consideration the usability and audience for the dashboard.

## Design

As you begin, keep in mind the design principles covered in Part II, “Building and Extending Dashboards and Balanced Scorecards,” particularly those visual guidelines relating to layout and the use of white space.

For the Adventure Works dashboard, take care to make the three executive objectives the central focus of the dashboard (see Figure 10-1). The KPIs are deemphasized by pushing them to the lower-right corner, which is the least significant area for left-to-right readers.

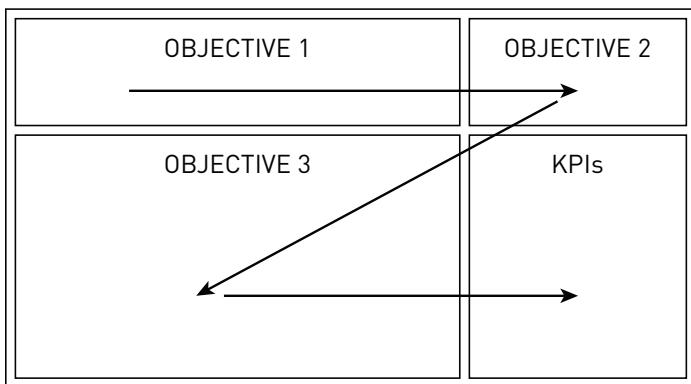


Figure 10-1

After the layout is finalized, select the most appropriate visualizations for each measure and metric. Since most of the information presented should show correlation and trends, the most appropriate data graphics for Objectives 1 and 2 are bar and line charts.

For Objective 3, the emphasis on geospatial data lends nicely to the use of a map with point and polygon layers. For the KPIs, indicators are the best choice since gauges or larger visualizations would distract the user from the primary objectives of the dashboard.

## Project Set-Up

To begin, start a new Visual Studio Report Server Project named AWExecutiveDashboard. Alternatively, because this dashboard is nothing more than another Reporting Services report, it can be included in an existing Reporting Services project, if you have one.

**NOTE** You need to have both the Adventure Works OLTP and Data Warehouse databases installed on your development SQL Server database instance. You can download the sample databases for this case study at <http://msftdbprodsamples.codeplex.com/>.

Before creating a new report for your dashboard, define the required data sources. Since your dashboard needs data from both the OLTP and Data Warehouse database instances, set up a data source using the following steps:

1. In the Solution Explorer, right-click the Shared Data Sources folder and select Add New Data Source.
2. Name the connection **dsAdventureWorks**, as shown in Figure 10-2 and click the Edit button next to the Connection String prompt.

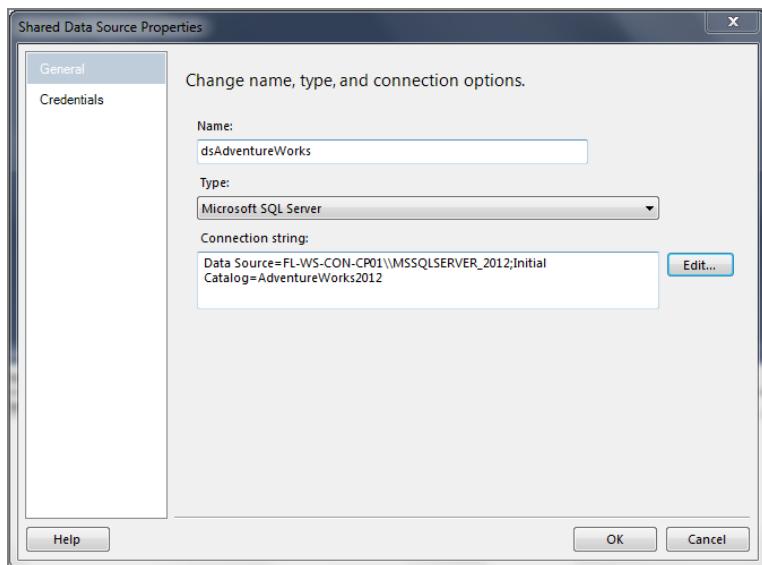
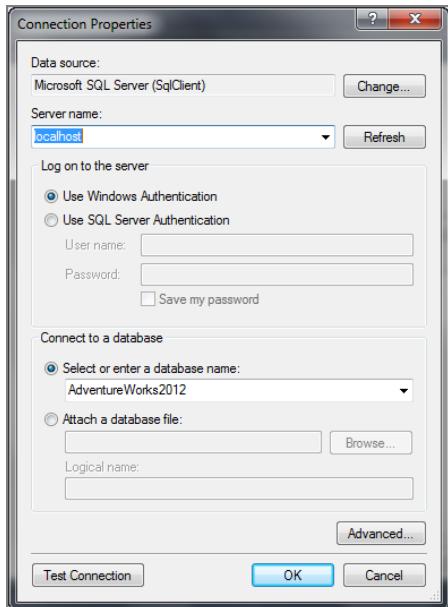
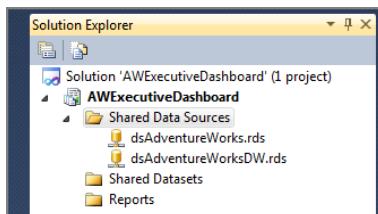


Figure 10-2

3. Enter **localhost** for your Server name and select the AdventureWorks2012 database (see Figure 10-3). Note that the server name and database name may vary if you are not using a local instance or if you are using a previous version of the Adventure Works database.
4. Click Test Connection to verify the connection configuration, and then click OK twice to finish.
5. Repeat the previous steps to create a connection that targets the Adventure Works Data Warehouse. Use the name **dsAdventureWorksDW** for the data source name. Your result should look like Figure 10-4.



**Figure 10-3**



**Figure 10-4**

After your data source connections have been created, it is time to create the datasets you need for the dashboard. Recall from Chapter 3, “Building your First Dashboard and Balanced Scorecard,” that there are two types of datasets available within Reporting Services: shared and embedded. To make the work you are doing reusable, use shared datasets whenever possible or practical.

For this dashboard, there are six datasets that need to be created:

**dsInternetSales**

**Source:** dsAdventureWorksDW

**Query:**

SELECT

```
d.MonthNumberOfYear,
d.EnglishMonthName,
SUM(s.SalesAmount) AS SalesAmount,
```

```
        SUM(s.SalesAmount) - SUM(s.TotalProductCost) AS GrossProfit
    FROM FactInternetSales s
    JOIN DimDate d ON (s.OrderDateKey = d.DateKey)
    WHERE d.CalendarYear = 2007
    GROUP BY
        d.MonthNumberOfYear,
        d.EnglishMonthName
```

**dsResellerSales****Source:** dsAdventureWorksDW**Query:**

```
SELECT
    d.MonthNumberOfYear,
    d.EnglishMonthName,
    SUM(s.SalesAmount) AS CurrentYear,
    MIN(ps.SalesAmount) AS PriorYear
FROM [dbo].[FactResellerSales] s
JOIN DimDate d ON (s.OrderDateKey = d.DateKey)
JOIN (
    SELECT
        d.MonthNumberOfYear,
        d.EnglishMonthName,
        SUM(s.SalesAmount) AS SalesAmount
    FROM [dbo].[FactResellerSales] s
    JOIN DimDate d ON (s.OrderDateKey = d.DateKey)
    WHERE d.CalendarYear = 2006
    GROUP BY
        d.MonthNumberOfYear,
        d.EnglishMonthName
) ps ON d.MonthNumberOfYear = ps.MonthNumberOfYear
WHERE d.CalendarYear = 2007
GROUP BY
    d.MonthNumberOfYear,
    d.EnglishMonthName
```

**dsCustomers****Source:** dsAdventureWorks**Query:**

```
SELECT
    soh.TotalDue,
    RTRIM(sp.StateProvinceCode) as StateProvinceCode,
    ad.City,
    ad.PostalCode,
    ad.SpatialLocation
FROM Sales.SalesOrderHeader AS soh
INNER JOIN Sales.Customer AS c ON
    soh.CustomerID = c.CustomerID
```

```

INNER JOIN Person.BusinessEntity AS b ON
    b.BusinessEntityID = c.PersonID
INNER JOIN Person.Person AS p ON
    p.BusinessEntityID = b.BusinessEntityID
INNER JOIN Person.BusinessEntityAddress AS a ON
    a.BusinessEntityID = b.BusinessEntityID
INNER JOIN Person.Address AS ad ON
    ad.AddressID = a.AddressID
INNER JOIN Person.AddressType AS at ON
    at.AddressTypeID = a.AddressTypeID
INNER JOIN Person.StateProvince AS sp ON
    sp.StateProvinceID = ad.StateProvinceID
WHERE c.PersonID IS NOT NULL
    AND at.Name = N'Home'
    AND sp.CountryRegionCode = N'US'
    AND YEAR(soh.OrderDate) = 2007

```

**dsResellers****Source:** dsAdventureWorks**Query:**

```

SELECT
    soh.TotalDue,
    st.Name AS Store,
    sp.StateProvinceCode,
    ad.City,
    ad.PostalCode,
    ad.SpatialLocation
FROM Sales.SalesOrderHeader AS soh
INNER JOIN Sales.Customer AS c
    ON soh.CustomerID = c.CustomerID
INNER JOIN Person.BusinessEntity AS b
    ON b.BusinessEntityID = c.StoreID
INNER JOIN Sales.Store AS st
    ON st.BusinessEntityID = b.BusinessEntityID
INNER JOIN Person.BusinessEntityAddress AS a
    ON a.BusinessEntityID = b.BusinessEntityID
INNER JOIN Person.Address AS ad
    ON ad.AddressID = a.AddressID
INNER JOIN Person.AddressType AS at
    ON at.AddressTypeID = a.AddressTypeID
INNER JOIN Person.StateProvince AS sp
    ON sp.StateProvinceID = ad.StateProvinceID
WHERE c.StoreID IS NOT NULL
    AND at.Name = N'Main Office'
    AND sp.CountryRegionCode = N'US'
    AND YEAR(soh.OrderDate) = 2007
AND soh.TotalDue > 10000

```

**dsCategorySales****Source:** dsAdventureWorksDW**Query:**

```
SELECT
    d.MonthNumberOfYear,
    d.EnglishMonthName,
    c.EnglishProductCategoryName AS Category,
    SUM(s.SalesAmount) AS SalesAmount,
    SUM(s.SalesAmount) - SUM(s.TotalProductCost) AS GrossProfit
FROM FactInternetSales s
JOIN DimDate d ON (s.OrderDateKey = d.DateKey)
JOIN DimProduct p ON (s.ProductKey = p.ProductKey)
JOIN DimProductSubcategory sc ON (p.ProductSubcategoryKey =
    sc.ProductSubcategoryKey)
JOIN DimProductCategory c ON (sc.ProductCategoryKey =
    c.ProductCategoryKey)
GROUP BY
    d.MonthNumberOfYear,
    d.EnglishMonthName,
    c.EnglishProductCategoryName,
    sc.EnglishProductSubcategoryName

UNION ALL

SELECT
    d.MonthNumberOfYear,
    d.EnglishMonthName,
    c.EnglishProductCategoryName AS Category,
    SUM(s.SalesAmount) AS SalesAmount,
    SUM(s.SalesAmount) - SUM(s.TotalProductCost) AS GrossProfit
FROM FactResellerSales s
JOIN DimDate d ON (s.OrderDateKey = d.DateKey)
JOIN DimProduct p ON (s.ProductKey = p.ProductKey)
JOIN DimProductSubcategory sc ON (p.ProductSubcategoryKey =
    sc.ProductSubcategoryKey)
JOIN DimProductCategory c ON (sc.ProductCategoryKey =
    c.ProductCategoryKey)
GROUP BY
    d.MonthNumberOfYear,
    d.EnglishMonthName,
    c.EnglishProductCategoryName,
    sc.EnglishProductSubcategoryName
```

**dsKPI****Source:** dsAdventureWorks**Query:**

```
SELECT 'Growth in Customer Base' AS KPI, 0.70 AS Value
UNION ALL
SELECT 'Net Income', 0.50
```

```

UNION ALL
SELECT 'Operating Profit', 0.80
UNION ALL
SELECT 'Return on Assets', 0.75
UNION ALL
SELECT 'Budget Variance', 0.70
UNION ALL
SELECT 'Channel Revenue', 0.50

```

**NOTE** Because the Adventure Works data source is static, the queries are designed to return a single year's worth of data. A better approach would be to either use a rolling 12-month period, or dynamic period, within the dashboard source.

Once all six datasets are created, you must create the report which is the foundation of the executive dashboard. To create the report, follow these steps:

1. Right-click the Reports folder in the Solution Explorer, then select Add, and then New Item.
2. Select Report from the Installed Templates, name the report ExecutiveDashboard, and then click the Add button (see Figure 10-5).

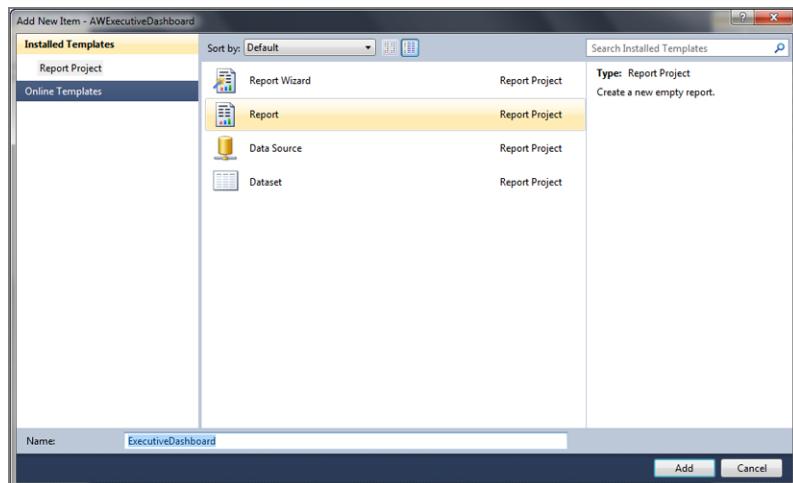


Figure 10-5

3. Finish the project set-up by adding the shared data sources and shared dataset using the queries from the preceding bulleted list. The process for creating these datasets is identical to those you previously created in Chapter 4.

### Defining Dashboard Layout

With the project set-up out of the way, you are ready to configure the layout for the dashboard. Keep in mind that one of the central tenets of good dashboard design is that all information fits on either a single page or screen, depending on the medium in which your dashboard will be delivered.

Adventure Works has decided that the executive dashboard should fit on a landscape, 8.5-inch x 11-inch sheet of paper. To set this configuration within Reporting Services, select the Report menu from the toolbar and select Report Properties.

On the Report Properties menu, select the landscape orientation (see Figure 10-6) and change all four margins to 0 inches.

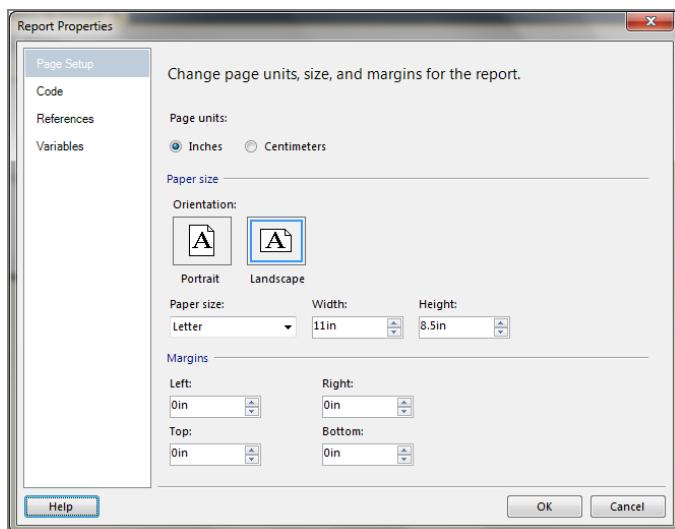


Figure 10-6

After the report properties are set, you can set up the initial report design surface. To easily manage the design surface, first make sure the ruler is visible. Right-click the design surface, select View and then Ruler to toggle the ruler. Once the ruler is visible, you can expand the design surface to 8.5 inches tall by 10 inches wide.

**NOTE** Alternatively, you can set the report size by adjusting the report body properties. To view and set these properties, the Properties Window needs to be visible. If this window is not visible, you can open it by selecting the View menu from the toolbar and then selecting Properties Window.

The remaining design elements, including fonts, color palettes, and visual boundaries, will be implemented as you build each report item.

### Graph: Internet Sales vs. Gross Profit

Displaying monthly trends between Internet sales and gross profit is best handled by a simple chart. As you saw in Chapter 6, “Visual Dashboards and Balanced Scorecards,” Reporting Services has robust charting capabilities. Use the following steps to build the Internet Sales vs. Gross Profit chart for the executive dashboard.

1. Add a Chart control from the Toolbox to the report design surface.
2. The basis of this chart will be a simple column chart. Select the column chart type and click OK.
3. Change the Chart Title to **Internet Sales vs. Gross Profit**.
4. This chart is going to display two different sets of values, both from the `dsInternetSales` dataset: Sales Amount and Gross Profit. Click the + icon to add both fields to the Values section of the Chart Data window.
5. To display the months along the x-axis, map the `EnglishMonthName` field as the Category Group.
6. The default behavior for the Chart control is to sort the category groups by name. For the months to display in the proper order, the sort order must be manually defined. Define the sort order by selecting the drop-down arrow on the `EnglishMonthName` and selecting Category Group Properties. In the Sorting window, define `MonthNumberOfYear` as the Sort by option (as shown in Figure 10-7).

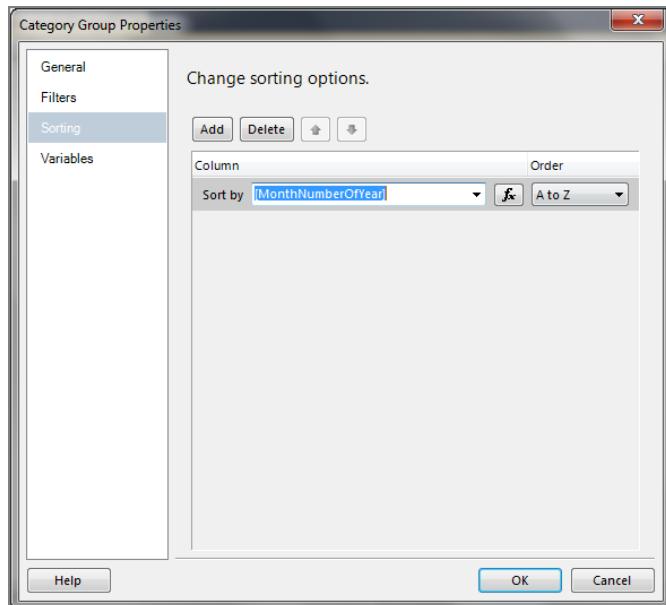


Figure 10-7

7. Next, make the profit trend easier to read by changing the chart type for the Gross Profit value from bar chart to line chart. Click the drop-down arrow on the Gross Profit and select Change chart type.
8. Using the techniques learned in Chapter 6, finish the chart by cleaning up the formatting with the following changes:
  - Move the chart legend to the bottom-center.
  - Format the vertical axis to use thousands formatting.
  - Hide the gridlines.
  - Hide the horizontal axis labels.
  - Remove the chart border.
  - Remove both the vertical and horizontal axis titles.
  - Change the chart palette to pastel colors.
  - Make the chart title 14 pt and soften the font color to gray.

After all the configuration and formatting changes are complete, your chart should look similar to Figure 10-8.

**NOTE** For the remainder of the chapter the basic formatting (axes, fonts, palettes, etc.) will not be repeated.

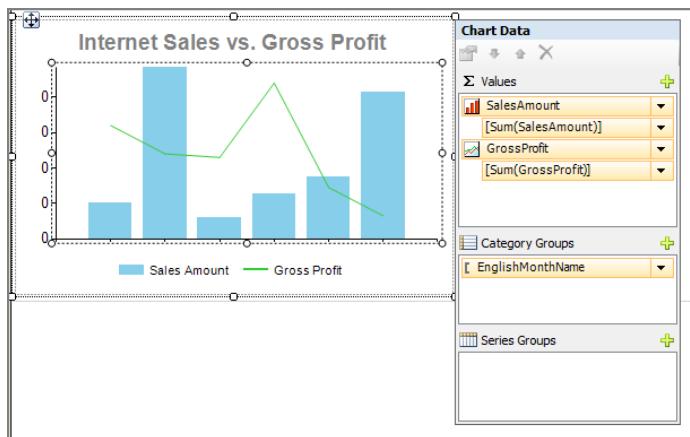


Figure 10-8

### Sparkline: Category Sales vs. Gross Profit

The second half to delivering the information required to monitor Objective 1 is a chart similar to the previous section only for the product category level. This scenario lends itself nicely to the use of a sparkline chart within a matrix.

To implement the sparkline visualization, follow these steps:

1. Drag a Matrix control from the Toolbox to the design surface.
2. Map the first column to the Category field in the `dsCategorySales` dataset.
3. Add a Sparkline control from the Toolbox to the second column of the matrix. Select Line as the Sparkline type.
4. Using the Report Data window, set the `SalesAmount` and `GrossProfit` fields as the Values and the `EnglishMonthName` as the Category Group. Don't forget to define the Sort by for the Category Group so that months are displayed in the correct order.
5. Add a Textbox control from the Toolbox to serve as a title for the matrix. Set the title of the visualization to **Category Sales vs. Gross Profit**.

After you have completed the steps and applied the dashboard formatting, the visualization should look like Figure 10-9.

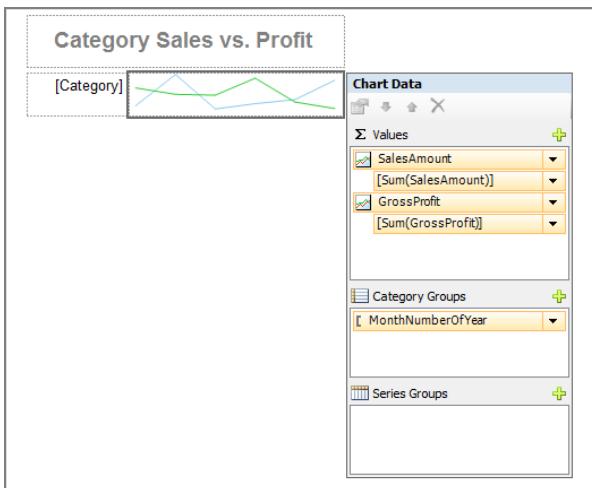
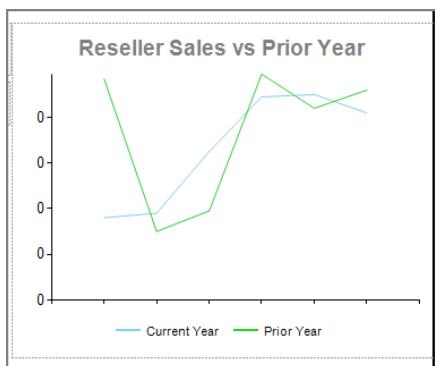


Figure 10-9

### Graph: Reseller Sales vs. Prior Year

Displaying performance against Objective 2 requires only a single chart. The chart is identical in implementation to the Internet Sales vs. Gross Profit chart you implemented previously, only it uses the `dsResellerSales` dataset. The Values for this chart are `CurrentYear` and `PriorYear`, and the Chart Type should be set to Line. The resulting chart should look like Figure 10-10.

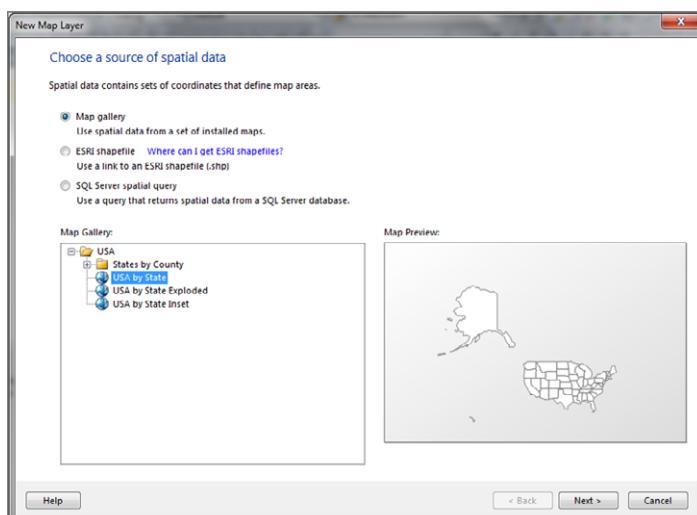
**Figure 10-10**

### Map: Internet & Reseller Sales

The third objective covered by the Adventure Works executive dashboard relates to geographical expansion of both direct customer sales via the Internet and sales through reseller partners. To visually provide a status for this objective, a map provides the optimal medium.

Robust mapping capabilities including Bing Map integration are natively built into SQL Server Reporting Services. Begin building your dashboard map by adding a Map control to the design surface and then complete the following steps:

1. The primary concern for the map you are building for the dashboard is the Midwest and southeastern regions of the United States. Begin the map by selecting the USA by State template from the Map Gallery (see Figure 10-11) at the first step of the Map wizard.

**Figure 10-11**

2. Zoom the map in so that the resulting view displays only the continental United States (as shown in Figure 10-12). For a dashboard map, typically you elect not to include Bing Map tile layers as the decorations since they do not provide any additional informative value.

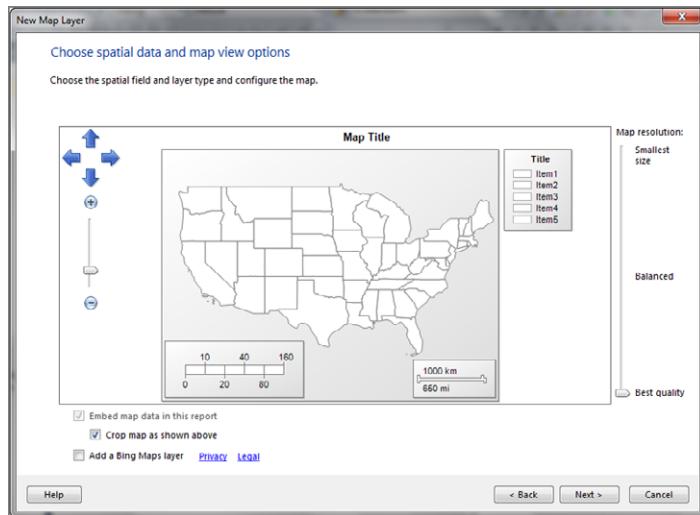


Figure 10-12

3. The dashboard map uses two different layers to illustrate both customer and reseller sales. Customer sales are visualized in a color, analytic, polygon layer and reseller sales use a point layer. Select Color Analytical Map (as shown in Figure 10-13) and click the Next button.

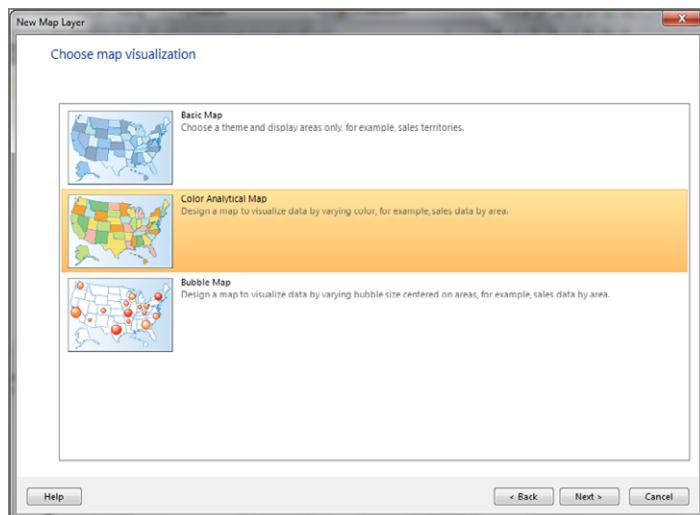


Figure 10-13

4. Using the `dsCustomer` dataset, map the STUSPS spatial dataset field to the `StateProvinceCode` field.
5. On the final step of the wizard, select the `Sum(TotalDue)` field as the field to visualize and change the color rule to White-Blue (as shown in Figure 10-14). You will make further changes to the color scheme to more accurately match the dashboard later.

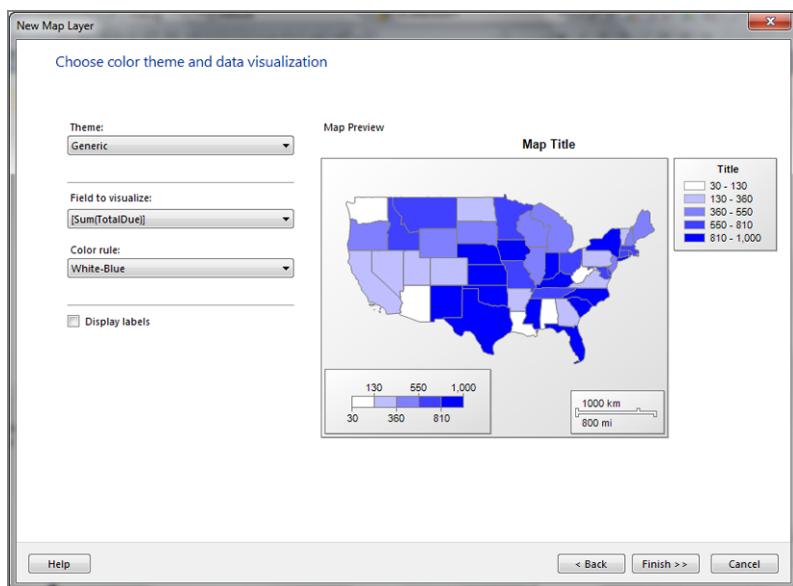


Figure 10-14

6. The Map Control wizard gets you most of the way there, but to finish the map from a data or information perspective, a point layer is still needed to chart the geographical dispersion. In the Map Layers dialog, select the Add Layer button and then Point Layer.
7. Using the drop-down arrow on the `PointLayer1`, select Layer Data. Map the `dsReseller` dataset to the new Point Layer, as shown in Figure 10-15.
8. Select Layer Wizard from the Layer Context menu. Select the Analytical Marker Map and click Next. Begin configuring the Analytical Marker Map by mapping the layer to the `dsReseller` dataset, then click Next.
9. Configure the layer so that resellers are marked with a star and that the marker color is used to visualize the data, as shown in Figure 10-16. Select White-Blue as the color rule.

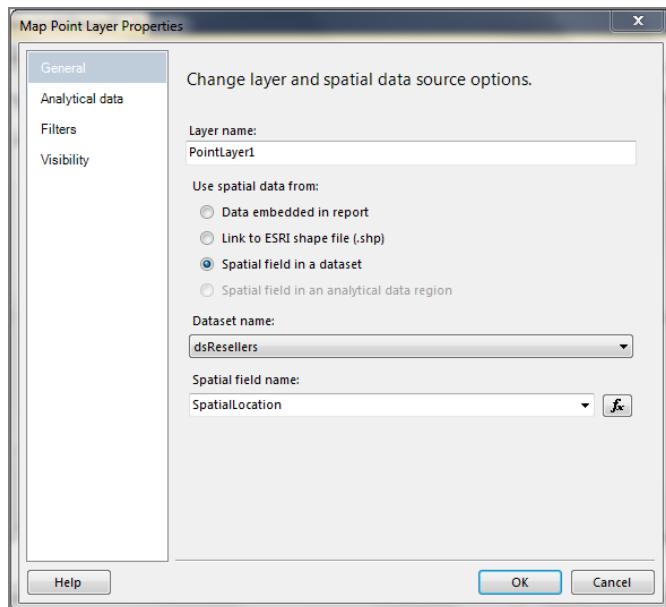


Figure 10-15

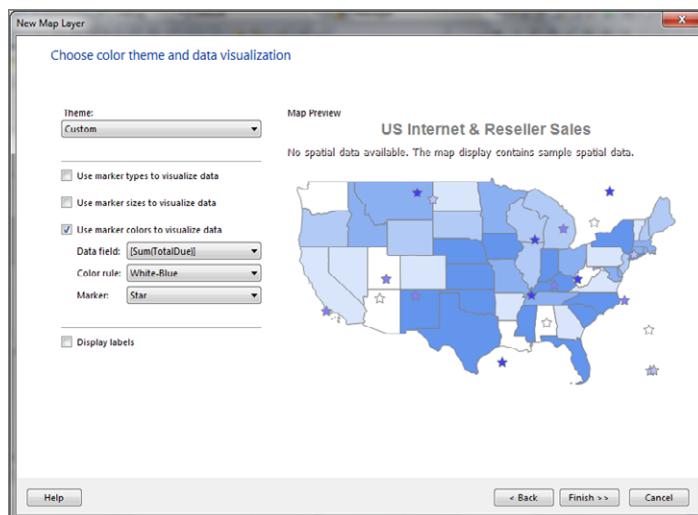


Figure 10-16

10. To finish the map, apply the following formatting:

- Title the map **US Internet & Reseller Sales**, then format the title so that it is consistent with the previous visualizations.
- Hide the Map Legend, Color Scale, and Distance Scale.

- Change the Map Viewport background to a solid white background. Remove the border and shadow to reduce visual noise.
- Change the Point and Polygon Color Rule for Start, Middle, and End colors to White, No Color, and Cornflower Blue, respectively.

The resulting map when rendered should look similar to Figure 10-17.

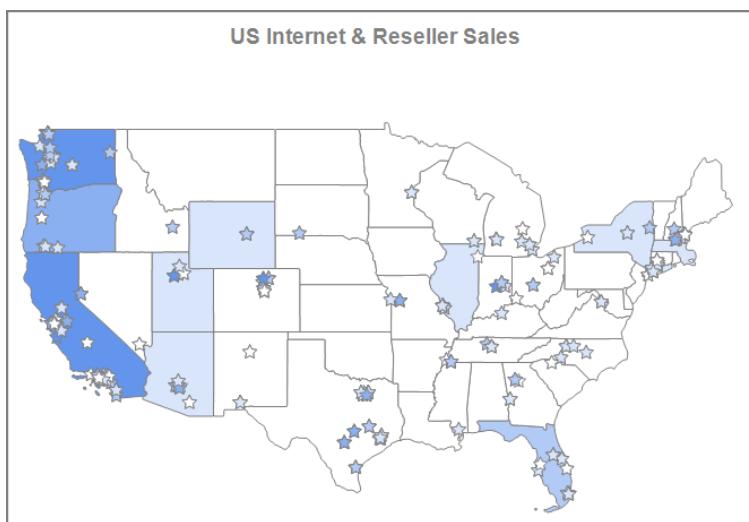


Figure 10-17

**NOTE** You can also change the centering point and zoom for the map by adjusting the view port properties.

### KPIs

The final section of the dashboard is a table that will be used to display an existing set of KPIs. Because the KPIs are not the main focus of the dashboard, avoid more elaborate visualizations such as gauges and thermometers in favor of indicators, which are subtler.

1. Add a Table control to the report design surface.
2. In the first column of the table, add the KPI field from the `dsKPI` dataset.
3. Add an Indicator control to the second column of the table and select the 3 Directional Arrows option.

4. Right-click the indicator in the table and select Indicator Properties.
5. On the Value and States page, select the `Sum(Value)` as the Value for the indicator. Change the indicator color to Cornflower Blue and set the state ranges to 0–33, 33–66, and 66–100 (see Figure 10-18).

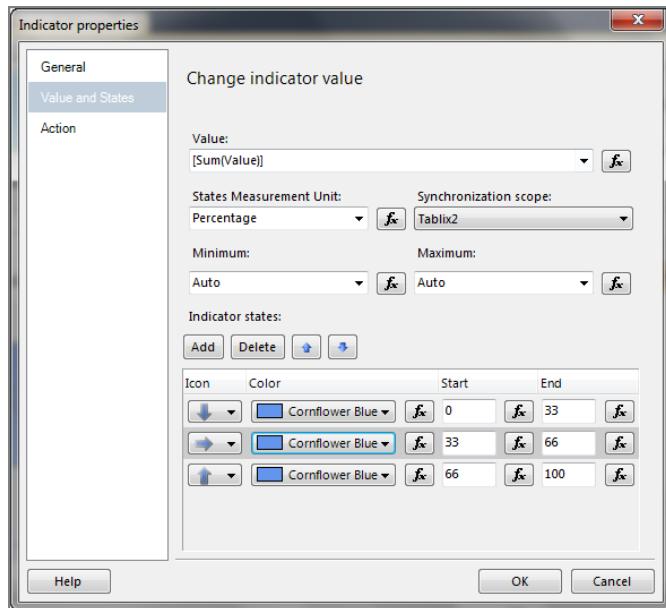


Figure 10-18

6. Remove the extra column and then merge the header row into a single column. Enter **Executive KPIs** as the title and then format for consistency.

After you are finished, preview the results.

### Finishing Touches

You are almost done, as the bulk of the dashboard work is complete. All that remains are finishing touches. Start by cleaning up and organizing the dashboard by resizing visualizations as necessary. Consider adding a title to the dashboard and footer with the report generation date if the dashboard is going to be printed.

Finally, use the Line control to establish a clear delineation between sections of the dashboard. When you are finished, the complete dashboard should look similar to the one shown in Figure 10-19.

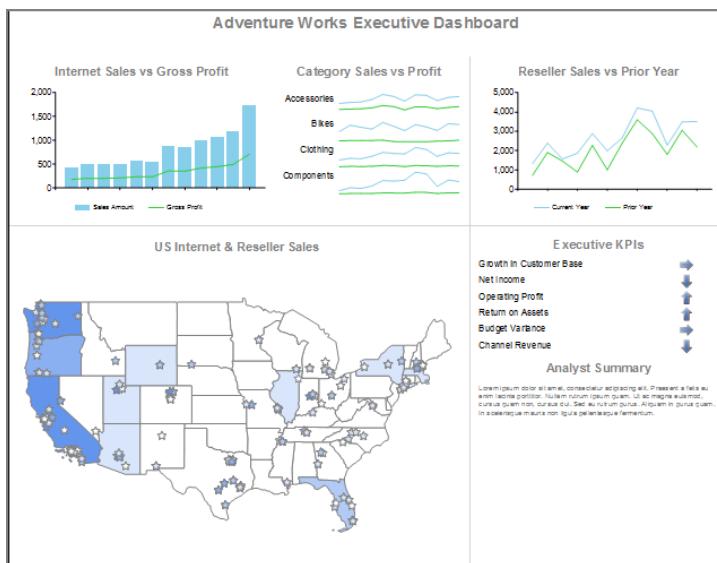


Figure 10-19

## Summary

Building on the techniques and skills learned in Part II, this chapter focused in on the techniques required to take your dashboard from concept to reality. Remember that building dashboards using SQL Server Reporting Services requires that you first identify organizational strategies, business objectives, and the metrics or measures needed to track them.

For each metric and measure, it is necessary to determine not only the visualization method that is most effective but how each metric and measure fits together into a single cohesive report or dashboard.

There are a number of different methods that are available to you, but letting the objectives and measures dictate the choices simplifies the process. Beyond the technical implementation demonstrated in this chapter, remember to refer back to the layout techniques discussed in Chapter 4, “Enhancing Your Dashboard and Balanced Scorecard,” and visualization techniques discussed in Chapter 6 before you start building out the data sources, datasets, and various report components.

In subsequent chapters, you will explore other analytical report types that are intended to enhance your understanding of how your business functions and performs.

# Strategy and Process Maps

You have a vision and you've gone through the painstaking steps to lay out a strategy to achieve it. Now what? For your strategy to ultimately succeed, it must translate to action. In other words, your strategy must be communicated clearly and effectively, in a way that is memorable and easy to understand.

The *strategy map* is a graphical tool that translates your strategy in terms of perspective, strategic theme, and objectives. *Process maps* compliment strategic maps by serving as a common language to simplify the complex business processes that drive strategic objectives. They are the foundation of process improvement.

This chapter introduces the history, concepts, and methodology behind both strategy and process maps. You will look at options for implementing each before diving into a detailed case study that focuses on building a simple process map for a plastics injection molding manufacturer.

## Strategy Maps

---

To understand strategy maps you must first start by understanding balanced scorecards. As you should recall from previous chapters, a *balanced scorecard* is a strategic performance report that focuses on the *vital few*, or the few metrics critical to measuring and monitoring the business objective or strategy at hand. Its primary purpose is to track strategic activities and their outcomes against defined targets in order to align strategy and action into a continual or iterative process.

For balanced scorecards to be successful, the correct strategic activities must be identified so that strategic objectives can be translated into organizational initiatives and projects. This is where strategy maps come into play. A strategy map (as seen in Figure 11-1) defines leadership's vision for the organizational strategy and the steps or paths necessary to realize it. The metrics from these activities are the essence of the balanced scorecard and the strategy map is their foundation.

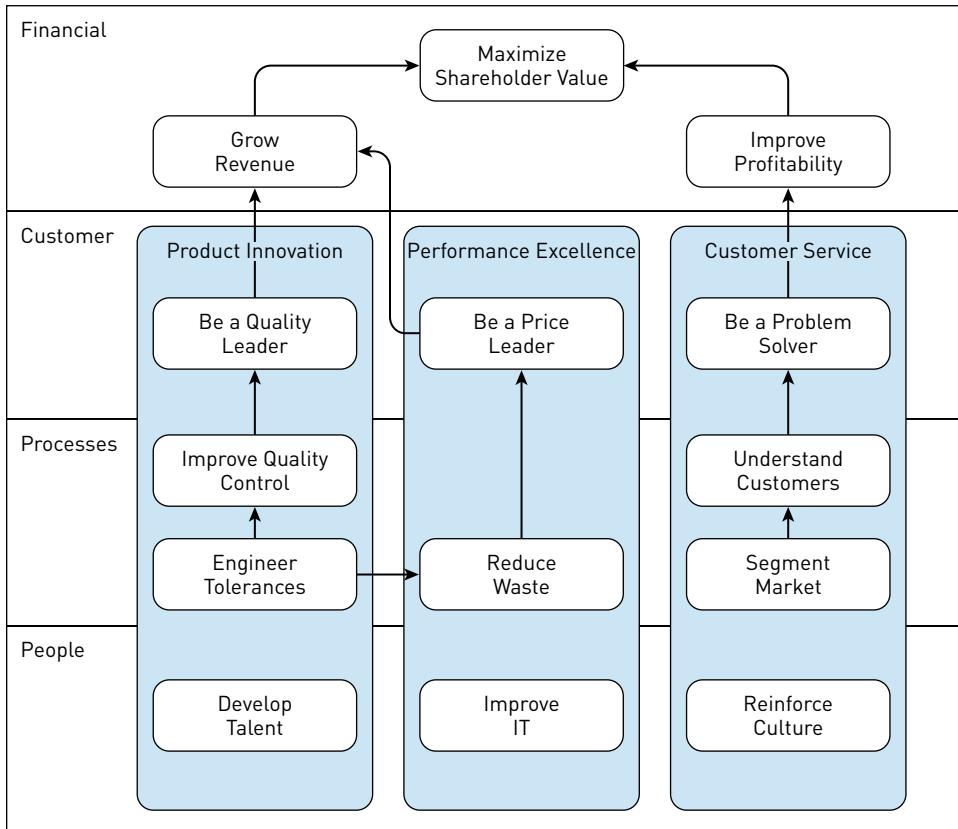


Figure 11-1

Robert Kaplan and David Norton first conceived balanced scorecards in the 1990s. It wasn't until 2001, however, when Kaplan and Norton made sweeping changes to balanced scorecards and introduced strategy maps as a management tool for helping organizations focus and align business objectives with goals. Strategy maps are useful for four reasons:

- They clearly and concisely communicate strategy throughout an organization.
- They provide a model for knowing what strategic themes drive the vision.

- They identify key objectives necessary to realize the strategic vision.
- They enable leadership to use key strategic objectives to drive initiatives that are in line with the overall strategy.

The implementation of a strategy map focuses on four perspectives: financial measures, customers, internal business processes, and innovation and learning. Strategic themes span these perspectives, creating a pseudo-grid on which bubbles, boxes, and other figures are used to define a small or manageable number of objectives that are critical to success. Arrows are used to show necessary causal links between objectives. The following sections cover these perspectives, themes, and designs in greater detail.

## Perspectives

*Perspective*, in the context of a strategy map, defines the aspects on which the company's activities and goals are evaluated. Perspectives are used to define goals or areas of focus for which metrics are developed, data is collected, and results are measured for the defined strategy. As mentioned previously, for-profit businesses typically look at four perspectives, as shown in Table 11-1:

**Table 11-1: Four Perspectives of Strategy Maps**

PERSPECTIVE	AREAS OF FOCUS/GOALS	METRICS
Financial Measures	Sales Growth Rate	Market Share
	Cross Selling Opportunities	% Revenue Growth
	Product Line Profitability	% Profit Growth
	Asset Utilization	Return on Capital
Customers	Customer Service	Customer Satisfaction Index
	New Products/Markets	Market Share
	Customer Retention	Repeat Orders/Purchases
Internal Business Processes	Manufacturing Excellence	Efficiency
	Streamlined Processes	Capacity Utilization
	Quality	% Defects
Innovation and Learning	Inventory Management	Inventory Turnover
	Technology Leadership	R&D as % of Sales
	HR/Team Development	Talent Acquisition
	Product Development	# of New Products

**NOTE** The four perspectives as defined by Kaplan and Norton might not be well aligned to your organization. This is fairly common, particularly in educational, government, and non-profit organizations. Feel free to alter, rearrange, or add perspectives to fit your organizational culture.

## Strategic Themes

Strategy maps use themes to transcend silos and to concentrate on specific points. *Strategic themes* clarify organizational strategy and allow for simplification of decision-making, budgeting, governance, and execution of tasks.

To identify strategic themes, an organization must find either gaps or opportunities between the current state of the business and the strategic destination. To achieve this, organizations typically start by using analytic tools to evaluate the business environment. Common analytic tools include: the PESTEL (Political, Economic, Social, Technology, Environmental and Legal) framework, SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis, and Porter's 5 Forces. These tools are discussed in the following sections.

### *PESTEL Framework*

The PESTEL framework was designed as an analytical framework of macroeconomic factors used to identify how each of six environmental factors influences business strategy. The six environmental factors are listed in Table 11-2.

**Figure 11-2:** Environmental Factors

ENVIRONMENTAL FACTOR	EXAMPLES
Political	Tax policies, trade regulations or agreements, stability of government, and social policies
Economic	Inflation, interest rates, living wage, unemployment, and credit market
Social	Demographics, lifestyle, education, and distribution of wealth
Technological	Platforms, innovations, obsolescence, and speed of changes
Environmental	Energy policies (consumption), waste/disposal laws, and popular attitude towards environment
Legal	Product regulations, employment laws, health and safety laws, and anti-trust laws

## *SWOT Analysis*

The SWOT analysis is a four-quadrant planning matrix that is used to define the strengths, weaknesses, opportunities, and threats for a product, place, project, or person with the goal of aligning business objectives with internal and external factors that will influence (helpful or harmful) the overall outcome. An example of a SWOT analysis is provided in Table 11-3.

**Table 11-3:** Sample SWOT Analysis

STRENGTHS	WEAKNESSES
Financial Reserves	Weak Brand Name
Location/Geography	Cash Flow
People/Processes/Systems	Project Deadlines
Price/Value/Quality	Employee Morale/Commitment
OPPORTUNITIES	THREATS
New Technology	Political Policies
Customer Demand	Seasonal/Weather
Industry/Population Trends	Economic
Competitor Vulnerabilities	Shift in Consumer Taste

## *Porter's 5 Forces*

The Porter's 5 Forces framework is an analytical tool used as a part of business strategy development to identify the competitive intensity of a market through five microeconomic forces. The five forces are broken out in Table 11-4.

**Table 11-4:** Porter's 5 Forces

FORCE	EXAMPLES
Threat of New Entrants	Brand equity, capital requirements, and customer loyalty
Threat of Substitute Product/Service	Switching costs, product differentiation, and quality
Bargaining Power of Customers	Price sensitivity, information availability, and availability of substitutes
Bargaining Power of Suppliers	Differentiation of inputs, substitute inputs, and distribution channel
Intensity of Competitive Rivalry	Firm concentration ratio, advertising trends, and online versus brick-and-mortar

### Common Themes

As you work to identify your themes, it's important to select broad areas that affect each perspective. These themes should reach each part of your organization and should be critical to achieving the strategic vision. At first glance, strategic themes can seem vague and meaningless, but they will come into focus when objectives are added, as discussed in the next section. Some of the most common strategic themes are as follows:

- Performance/Service Excellence
- Sustainability
- Customer Service
- Innovation
- Quality
- Safety
- Clinical Leadership

Typically, you want to limit the number of themes you select to only 3 or 4. This enables your organization to focus both its efforts and resources, without incurring the law of diminishing returns—resources spread too thin—and ultimately risking failure.

Within the strategy map, themes make up the columns on the pseudo-grid described previously, as they intersect each silo or identified perspective. Themes serve as the foundation or broad areas in which your organization must excel and are used to identify objectives.

### Objectives and Causal Links

The goal of the strategy map is to fulfill the strategic vision or mission that defines it. Perspectives enable you to set goals, define metrics, and identify themes, which help focus in on the activities that must be executed successfully for the strategic vision to be realized. These activities take the form of *objectives* and provide substance to strategic themes on the strategy map.

To begin identifying objectives, there are two good rules to follow. First, as you begin identifying objectives, take care to not over-fit the strategy map to the daily or performance activities of your business. Over-fitting results in a performance map that defines where the organization is today, including any weaknesses or problem areas versus focusing on realizing the vision for the future.

Second, the objectives on the strategy map should identify the critical few activities that can be measured and monitored. Typically, large corporations have fewer than 15 objectives, and research has shown that having less than 20 objectives improves the probability of success.

After objectives are defined and placed on the pseudo-grid created by the intersection of perspectives and strategic themes, arrows or lines are used to connect the objectives to illustrate relationships, or causal links. *Causal links* are defined when there is a cause-and-effect relationship between objectives. For example, a manufacturing company might identify an objective that focuses on Engineering Tolerances, which would have causal links on the Improve Quality and Reduce Waste objectives. Furthermore, a relationship between Reduce Waste and Price Leader objectives could be shown.

Diagramming these links is important to illustrate how objectives work together toward a theme and, when successfully achieved, toward the strategic vision.

## Designing a Strategy Map

The process of designing a strategy map that accurately reflects the organizational strategy is an iterative group or team process. It typically involves members of the executive team coming together with a facilitator to hash out strategic themes and objectives.

In his book, *Balanced Scorecards & Operational Dashboards with Microsoft Excel*, (Wiley, 2013) Ron Person proposes two techniques for designing strategy maps. The first technique, the *Straw Dog Approach*, involves building a draft strategy map to serve as a sacrificial starting point. The draft enables the executive leadership team to begin thinking about themes, objectives, and causal links. The downside to this technique is a lack of commitment, otherwise known as the *rubberstamp effect*, named for the fact that executives will take the path of least resistance and prefer the draft instead of actively participating.

The second technique proposed by Person is the *Brainstorming and Intensive Discussion Approach*. This approach is more intensive and requires more involvement and commitment from the leadership team as they meet to brainstorm a strategy map on the fly. The downside of this approach is the increased requirement of executive time and the need for an experienced facilitator to lead the session.

Regardless of the technique chosen, be sure to document each step and the artifacts generated throughout the process. Keeping notes, drawings, and even digital photos of whiteboard sessions will be useful as you move forward.

### CHOOSE AN EXPERIENCED AND CERTIFIED FACILITATOR

An experienced facilitator can make the difference between success and failure when designing strategy maps. The right facilitator will not only help guide the process but will ask the right questions and provide support through case studies and practical business experiences.

## Process Maps

---

Where the strategy map visualizes the path to realizing a company's strategic vision, the *process map* is used to visualize the steps or activities necessary to deliver a service or produce a product. Process maps come in a variety of types and can range from a very high level that abstracts many of the details to the highly detailed maps that show activities minute to minute. These maps are generally created with the intent to either monitor or improve a business process. Creating these maps can help point out important parts of the process, such as key inputs and outputs, including identifying potential bottlenecks, redundancies, and waste.

Before you begin a project to map out a performance process or business activity, it's important to understand context. Mapping projects are typically done as a key function of process improvement. In fact, improvement methods such as Six Sigma and Lean Manufacturing require an effective process map. To understand the context, ask yourself the following questions: What is the purpose of the project? How does it benefit the company? Does it align with the organization's strategic objectives? Understanding the context helps avoid some of the pitfalls, particularly when making a case for change.

**NOTE** Six Sigma and Lean Manufacturing are frameworks for process improvement that identify and remove defects, waste, and variation to increase product or service quality while reducing costs. In addition to process maps, these frameworks depend on many other analytical reports, most notably dashboards. For additional information on Six Sigma and Lean Manufacturing refer to:

- International Society of Six Sigma Professionals: [www.isspp.com/](http://www.isspp.com/)
- Lean Enterprise Institute: [www.lean.org/](http://www.lean.org/)

To better understand process maps, take a look at three specific types of maps found in the organizational environment: process flow, SIPOC map, and value stream map. As different as each of these map types are, it's important to remember that conceptually they share a common base. Each map type strives to provide details or insight into the business process it models, specifically seeking to identify inputs, outputs, bottlenecks, and points of risk.

### Process Flow Map

The *process flow map* is a process visualization tool with which you are most likely already familiar. This map type diagrams each step or stage in the relevant business process and in addition can be used to illustrate relationships between processes or in a cross functional manner between processes and people. The process map is built using rectangles, ovals, triangles, and other

shapes (shown in Figure 11-2), linking each with a line called a connector. If you're visualizing a flowchart, you are correct; the process map is commonly referred to as a process flowchart.

### REQUEST FOR QUOTE CENTRALIZATION PROCESS

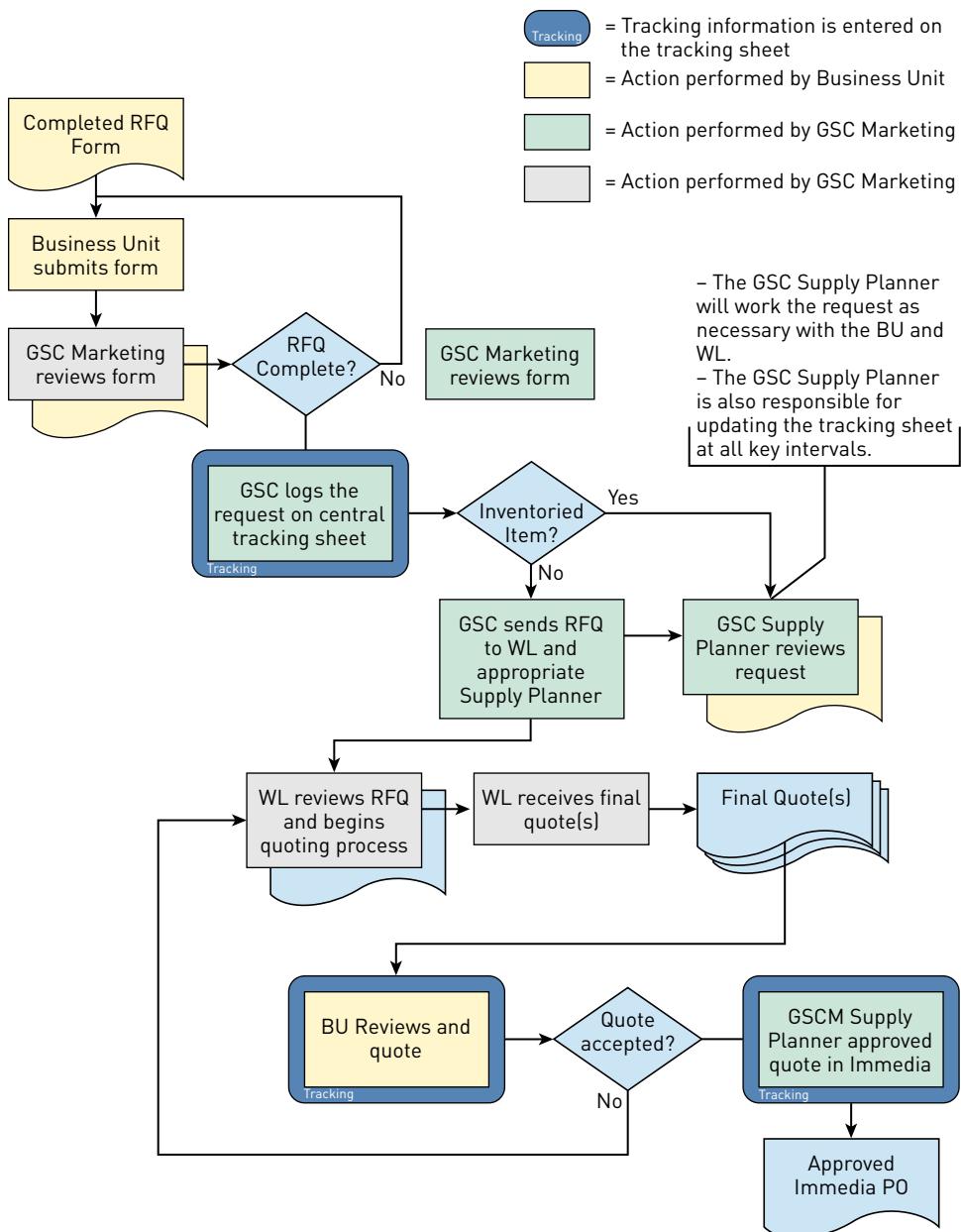


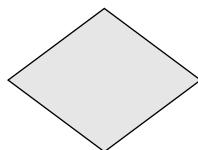
Figure 11-2

The following figures provide examples of common process map shapes and their corresponding usages.



Process or Activity  
(Rectangle)

**Figure 11-3**



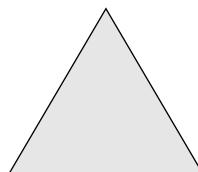
Decision  
(Diamond)

**Figure 11-4**



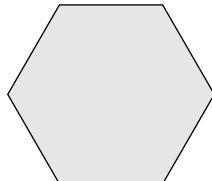
Flow Continuation  
(Circle)

**Figure 11-5**



Storage/Inventory  
(Triangle)

**Figure 11-6**



Set-Up/Preparation Activity  
(Hexagon)

**Figure 11-7**



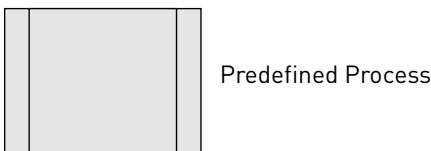
Wait/Delay  
(Half Oval)

**Figure 11-8**

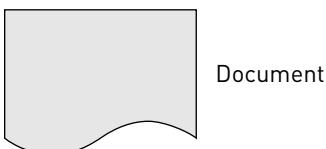


Terminator/Start/End  
(Oval)

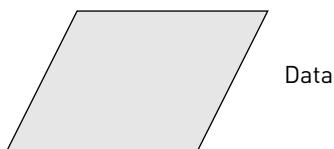
**Figure 11-9**



**Figure 11-10**



**Figure 11-11**



**Figure 11-12**

Process flow maps are useful for simplifying complex processes and can facilitate better understanding of the entire activity through visualization. They also excel at identifying pain points within a process and are thus useful in process improvement projects.

Building a process flow map can be simplified to three steps:

1. Determine the process boundaries (frame the process). Identify where the process starts and ends, including the level of detail required by the map.
2. Identify the process steps. The process steps should include inputs, outputs, activities, and decisions and should adhere to the level of detail identified in step 1.
3. Sequence the steps. Arrange the steps so that they follow the order in which they occur within the process.

**NOTE** During brainstorming sessions, use sticky notes as you identify process steps. Sticky notes enable you to easily and visually arrange each step during sequencing.

After you have completed your map, it should be thoroughly tested to validate the data flow. Take care to ensure the required level of detail is correct, all process tasks are accounted for, and no open-ended flows exist.

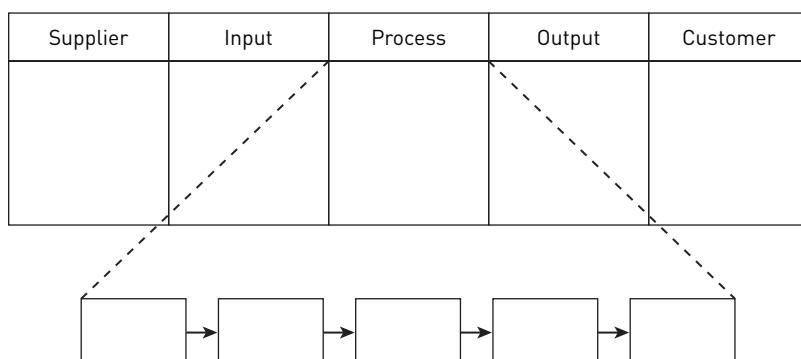
## SIPOC Map

The *SIPOC* (Supply, Input, Process, Output, Customer) map is a tool that was born out of the Six Sigma framework and is typically used in process improvement. Starting by identifying the customer, who can be either internal or external to the company, the SIPOC map is built backwards by recording the outputs, then the process, inputs, and ultimately the supplier (internal or external).

The purpose of the map is to document the in-scope process at a high level, including the process boundaries, which are defined by the inputs and outputs. The SIPOC map is a table-based, text diagram. The process description is typically broken out to show major milestones from start to finish within the process. To further illustrate, Table 11-5 provides a sample SIPOC map for a mortgage process that is then graphically illustrated in Figure 11-13.

**Table 11-5:** SIPOC Map for Mortgage Process

SUPPLY	INPUT	PROCESS	OUTPUT	CUSTOMER
Listing of all suppliers to the process, internal or external	All inputs required by the process including any quantifiable expectations	A description of the process including major milestones from start to finish	Outputs of the process	Customers of the process
■ Appraisers ■ Title Companies ■ Insurance Companies ■ Fannie Mae	■ Basis Points ■ Interest Rate ■ Loan Value ■ Loan Type ■ Loan Application	■ Prepare ■ Process ■ Underwrite ■ Clear Conditions ■ Close	■ Loan Documents ■ Mortgage	■ Mortgage ■ Bank



**Figure 11-13**

## Value Stream Map

*Value stream mapping* originated at Toyota as a Lean Manufacturing technique to analyze and design how material and information flow to bring a product or service to a customer. Whether it is used during the design (concept to customer), build (order to delivery), or sustain (lifecycle) portion of the value stream, this mapping tool can help companies improve processes and reduce waste.

Unlike the process maps discussed previously, the value stream map (see Figure 11-14) does not necessarily focus on a single process. Instead, this type of map focuses on either a single product or service, and in some cases an entire product line.

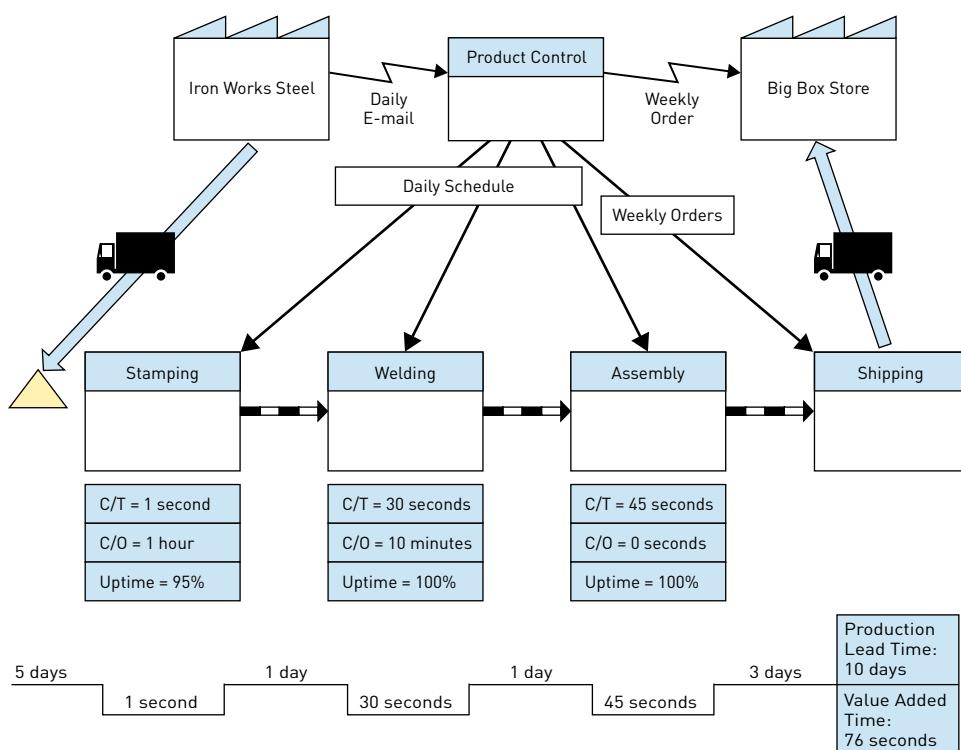


Figure 11-14

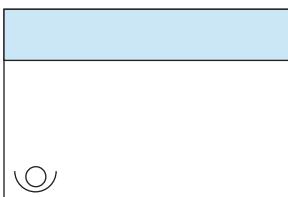
The value stream map is a diagram that is similar to a flowchart but has its own set of special symbols known as “the language of the lean.” Within the value stream, processes or activities required to deliver the product or service are recorded, as is the data metric needed to observe the process. Some of the common data metrics recorded for each task are described in Table 11-6.

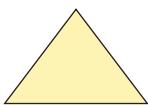
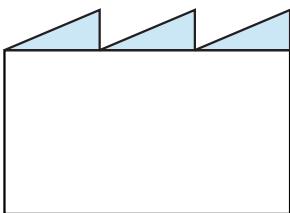
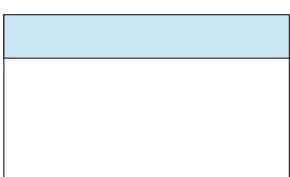
**Table 11-6:** Value Stream Data Metrics

NAME	ABBREVIATION/ EXAMPLE	DESCRIPTION
# of Process Operators	30 Operators	Describes the number of operators required for the process.
Error Rate	1% Error	The average expected error rate.
Set-up/Change Over Time	C/O - 1 day	Set-up time required to begin a new task.
Cycle or Task Time	C/T or T/T - 3 days	The time needed to complete the task.
Wait Time	W/T - 2 days	Wait time before next process starts.
Total Task Time	TC/T - 5 days	The total time from when a task is started until it leaves the process.
Value-Added Time	VAT - 2 days	Time needed to complete a required task.
Non-Value-Added Time	NVA - 1 day	Calculated as C/T - VAT. Represents tasks such as time waiting on operators, mail time, or time waiting on external sources.

Actors such as customers, suppliers, external sources, and operators, as well as inventory are all diagrammed using the symbols described in Table 11-7. Multiple types of flows are recorded, including manual and electronic as well as pushes and shipments.

**Table 11-7:** Value Stream Map Symbols

ICON	DEFINITION	DESCRIPTION
	Process	Can be a process, operation, department, or machine through which materials flow.

ICON	DEFINITION	DESCRIPTION
	Inventory	Inventory held between two processes.
	Push Arrow	The movement of materials from one process to another.
	Customer/ Supplier	This icon represents the supplier when on the left side of the diagram and the customer when on the right side.
	Shipment Arrow	Movement of raw materials from suppliers or finished goods to customers.
	Shipment Truck	Shipments from suppliers or to customers.
	Production Control	A central production scheduling department, operation, or person.
	Manual Info Flow	Records memos, reports, or conversations, including relevant frequency.
	Electronic Info Flow	Records e-mails and electronic data exchanges.

*Continued*

**Table 11-7:** *(continued)*

ICON	DEFINITION	DESCRIPTION
	Data Flow	Metrics required to analyze and observe the adjacent process.
	Supermarket	An inventory supermarket.
	Timeline	Used to compare value-added and non-value-added time throughout the process.
	Timeline Total	Total value-added and non-value-added time.
	Safety Stock	Inventory hedge to buffer unexpected events.

## Case Study: Contoso Plastics

The remaining portion of this chapter walks through a case study for a fictional company called Contoso Plastics. The case study is intended to reinforce the techniques for implementing both a process and strategy map discussed previously.

## Background

Contoso Plastics was founded in 1990 in the southeastern United States and is the leading manufacturer of custom, injection-molded plastics in North America, with revenues that exceeded \$500 million in 2012. The small but powerful company employs a highly skilled workforce of just over 190 employees who support the company vision of first-rate customer service while offering high quality, environmentally friendly products.

Contoso's primary customers are located within the United States and are mostly industrial. Many of Contoso's finished products serve as inputs or raw materials to finished commercial and consumer products.

## The Situation

In 2012, the founders of Contoso Plastics sold the company to an investment group. The new executive leadership team noted that while the company had been proactive in acquiring and implementing new technology, a number of performance inefficiencies lead to excessive waste and intermittent quality problems. These issues contributed to overall flat growth over the prior five years, while competitors continued to increase market share.

New leadership was concerned that even though the company had been highly successful, there was little visibility into the manufacturing process. Few if any metrics existed to track manufacturing, and the data or reports that were available were locked away in Excel silos. Understanding that change was necessary to ensure viability, executive leadership engaged a Six Sigma (process improvement) consultant.

One of the key concepts of the Six Sigma process is known by the acronym DMAIC. This acronym is short for:

- Define
- Measure
- Analyze
- Improve
- Control

Because the project was in the early definition phase, the consultant recommended putting together a SIPOC (Supply, Input, Process, Output, Customer) map to document and summarize the manufacturing process at a high level that is at the core of Contoso's business.

## Process Improvement

The project charter for the process improvement project focused primarily on the manufacturing process, since it is the heart of Contoso Plastic. The project objectives fell in line with those set forth in the Six Sigma doctrine. The primary objectives for the project were:

- Continuously achieve stable and predictable manufacturing results.
- Develop a core set of metrics that can be measured, analyzed, controlled, and improved.
- Sustain quality improvements.

## Design Sessions

On the first day of the project, the process improvement consultant began constructing a rough draft SIPOC map, which included a high-level view of the overall process picture to identify the customer and where the process starts. He started at the loading dock where the completed customer orders are loaded on trucks and worked his way up the process stream to identify each trigger, until the customer order is identified.

**NOTE** It's not uncommon to find that the generally accepted or documented process is either incomplete or just wrong.

For Contoso Plastics, the consultant was able to track down the origination of the customer order, which occurred electronically when a customer uploaded an engineering (CAD) file. From that point, the consultant was able to walk through and sketch a drawing of the entire process, including each step, metrics associated with each step, and notes about any observed inefficiencies, bottlenecks, or waste.

Table 11-8 lists the information identified by the consultant as part of his discovery.

**Table 11-8:** SIPOC Players

CATEGORY	PLAYERS
Suppliers	Raw Materials (Raw Plastic Bead) Supplier
	Quality Control Department
	Production Management

CATEGORY	PLAYERS
Inputs	Raw Plastic
	Skilled Production Personnel
	Quality Control
	Plastic Die Mold
Process	Injection Molding
Output	Finished Product
	Plastic Waste
	Quality Control Records
Customer	Customer/Warehouse
	Recycler/Regrinder
	Quality Control Department

The defined steps of the injection molding process were identified as:

1. Raw plastic pellets are fed into the machinery through the hopper.
2. Plastic pellets are heated and melted in the cylinder and transported by screw.
3. Molten plastic is forced into the die.
4. Refrigerant is circulated around the die to cool the plastic.
5. A moveable plate retracts and the plastic is ejected to a storage container.

After the draft map was created, the consultant reviewed it with Contoso's performance managers. The goal of this review was to verify and normalize the high-level process and make adjustments, including any steps or metrics that may have been overlooked.

To finalize the draft of the map, the Six Sigma consultant met with a wide array of Contoso personnel who are subject matter experts or are familiar with either the whole process or parts of the process. The goal of this step was to document additional information such as insight into processes and detailed metrics.

## Anecdotal Results

While the SIPOC map is an input into the larger landscape of a Six Sigma process improvement project, the inclusion of individual contributors and Contoso personnel increased organizational buy-in, lending support for success of the project. Mapping the injection molding process was a key first step to identifying which process tasks had the greatest impact and thus the most potential gain.

## Implementing Strategy and Process Maps

---

Although this book is about making use of SQL Server Reporting Services, building strategy and process maps on this platform requires a fair amount of additional work. At issue is Reporting Services' limited support for custom shapes and symbols, which are typically required to implement such tools. Instead of hammering the square peg into the round hole, you should learn about more traditional techniques for producing these maps.

### Working in Microsoft Visio

Microsoft Visio is the tool of choice for building most business diagrams or analytical charts. Out of the box, Visio supports typical flowchart symbols, numerous libraries of shapes and symbols, multiple types of containers, custom shapes, and free-form drawing tools. There is also a ready-to-use template for the value stream map and its custom symbols.

New documents are started in Visio by selecting a drawing template. For strategy maps, process flow maps, and SIPOC maps, a good starting point is the Basic Flowchart template. The template for the value stream map can be found under the Business template category.

To orient and prepare for working in Visio, familiarize yourself with the shapes and Shapes Library. Shapes are added by dragging and dropping them onto the drawing surface. If your drawing requires that shapes be connected, as is typical when connecting objectives within something like a strategy map, connectors can be used. Connectors attach to shapes at various connection points.

**NOTE** There is an extensive collection of shape stencils available out of the box within Visio. You can search from shapes within the Shapes Library. If you don't find a particular shape you need for your map, try browsing the Internet for Visio Stencil collections.

Beyond the library of shapes available, there is extensive support for custom drawing in Visio. Tools for drawing rectangles, ellipses, lines, free-form shapes, and arcs are also available. While this isn't an extensive tutorial on working with Visio, the following steps provide a good foundation for building the SIPOC map defined in the Contoso case study.

1. Open Visio and create a new drawing using the Basic Flowchart as the selected Template (see Figure 11-15).
2. Using the rectangle shape tool found on the toolbar, construct a grid with two rows and five columns. Resize the rectangles so that the top header row has a height of approximately 0.5 inches and the bottom row is about 2.25 inches.

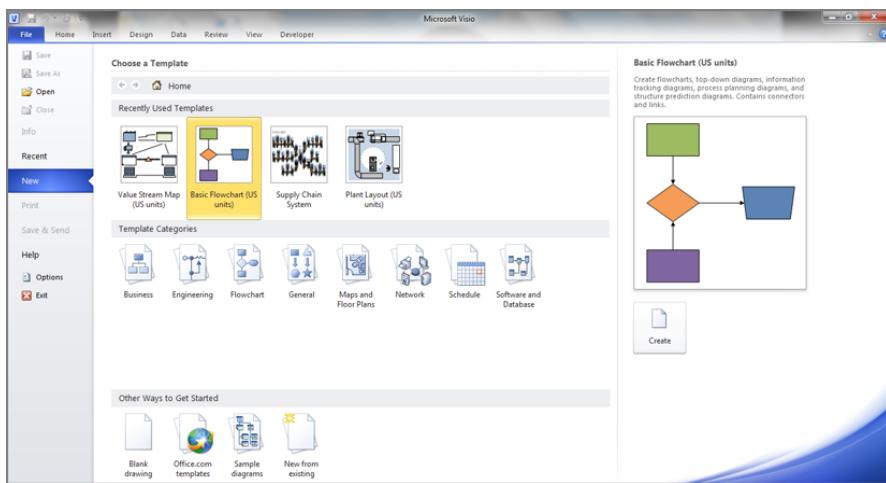


Figure 11-15

**NOTE** When shapes need to be kept together, consider grouping them. This will enable you to move, resize, and format them as a single unit rather than individually. To group objects in Visio, use Ctrl+Click to select each object and then right-click the selected objects and choose Group. To ungroup, right-click the grouped objects and select Ungroup.

3. Select all the rectangles and then change the font size by right-clicking, choosing Format, and then Text. Change the font size from 8 pt to 12 pt, making the header style Bold.
4. From left to right, enter the text for the column headers: **Supplier**, **Input**, **Process**, **Output**, and **Customer**.
5. In the data row, enter the players identified by the consultant during discovery from Table 11-8 as shown in Figure 11-16.

Supplier	Input	Process	Output	Customer
Raw Plastic Supplier Quality Control Department Product Management	Raw Plastic Skilled Product Personnel Quality Control Plastic Die Mold		Finished Product Waste Plastic Quality Control Report	Customer/ Warehouse Recycler/ Regenerator Quality Control Department

Figure 11-16

6. Next, add five Process shapes from the Basic Flowchart Shapes stencil. With all five shapes selected, click the Auto Align & Space button on the Home tab to evenly space the process shapes.
7. Enter the text identified for the five injection molding process steps from Table 11-8 as shown in Figure 11-17.

Supplier	Input	Process	Output	Customer
Raw Plastic Supplier Quality Control Department Product Management	Raw Plastic Skilled Product Personnel Quality Control Plastic Die Mold		Finished Product Waste Plastic Quality Control Report	Customer/ Warehouse Recycler/ Regrinder Quality Control Department

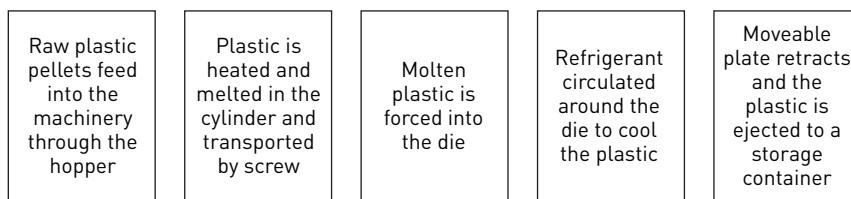


Figure 11-17

8. Use the Connector tool to connect each step to the next in the process flow.
9. Draw two lines using the Line Tool. If the Line Tool is not immediately visible, click the drop-down arrow next to the Rectangle Tool and then select the Line Tool to make it active.
  - Draw a line from the bottom left of the Process header rectangle to the top left of the process flow.
  - Draw a line from the bottom right of the Process header rectangle to the top right of the process flow.
10. Finish by right-clicking each line and choosing Format and then Line. Configure the line so that it is dashed with a line weight of 1.25 pt and has 60-percent transparency. Figure 11-18 displays the resulting image.

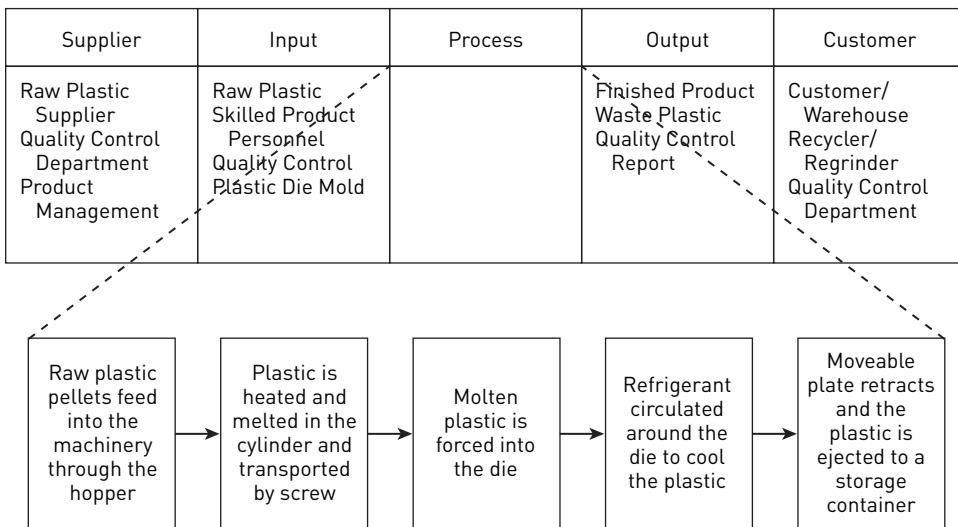


Figure 11-18

## Alternative Implementation Options

While Microsoft Visio is the standard-bearer in terms of business diagrams, there are a number of other options available within the Microsoft Office suite. Word, PowerPoint, and to a lesser extent, Excel can be used to build process and strategy maps. Excel, Word, and PowerPoint all support robust drawing capabilities through the Illustrations section on the Insert tab in Office 2010 as seen in Figure 11-19. (In prior versions the same features are available through the Drawing toolbar.)

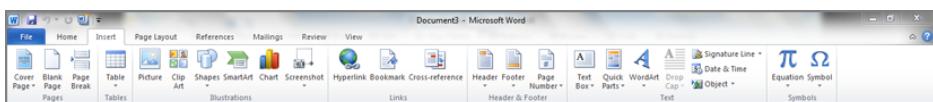


Figure 11-19

When the shape is selected, the Drawing Tools tab displays. The Drawing Tools tab allows for easy configuration of your shape, including:

- Applying styles
- Setting the shape's fill, line format, and text properties
- Applying shape effects

- Adding shape connectors with edges that can be attached
- Adjusting shape size, alignment, positioning, and grouping
- Creating custom shapes

In addition to the built-in capabilities, plug-ins are available that extend the base functionality of these Microsoft Office products to include broad features such as flowcharting and more specialized features such as value stream maps.

#### SHAREPOINT PERFORMANCE POINT AND VISIO SERVICES

To take your strategy and process maps to the next level, consider Performance Point, available in SharePoint 2010, which has a built-in strategy map report. The name strategy map in Performance Point is a bit of a misnomer since it's capable of much more than simple strategy maps.

Behind the scenes, the report uses Visio Services to include a hosted Visio drawing whose shapes can be linked to metrics or key performance indicators (KPIs) on a balanced scorecard. Tying the Visio shapes to metrics or KPIs enables the shapes to visually provide a status for each process or objective on the map.

---

## Summary

---

Whether bringing clarity to a complex process or communicating an organization's strategic vision, both strategy and process maps offer a common language to simplify your efforts. Use the well-established techniques laid out for defining business strategies and mapping business processes.

As you design, build, and ultimately implement your map, regardless of the technology you use, stay true to the goals and ideals of each map type and you will be positioned for success.

## Balanced Scorecards Case Study

So far in this book you have learned how balanced scorecards can be used to plan out your company's "big picture" strategies. You've learned how to set clear objectives and define measurements, so you have some way to quantify progress. You also learned about setting appropriate goals or targets, so you can define your initiatives as successful or unsuccessful.

A well-planned, balanced scorecard focuses on four business perspectives. First is the learning and growth perspective, which includes metrics for employee satisfaction, retention, training, and development. Next is the internal business process perspective, which includes metrics for operations management, order fulfillment, and quality control. Then there is the customer perspective, which includes metrics for customer satisfaction and retention, brand strength, and market share. Last is the financial perspective, which includes metrics such as revenue, growth, productivity, and others related to increasing shareholder value. These perspectives build on each other, if you consider a statement like this: Training your employees will help the business run smoother, which provides more time to spend with customers, who will likely make more purchases.

Previous chapters in this book covered these perspectives in great detail. The goal of this chapter is to take what you have learned about balanced scorecards and apply it to a real-world example. You will walk through a scenario of building a balanced scorecard for a marketing firm that is looking to grow its business.

## Designing a Balanced Scorecard for a Marketing Firm

When designing a balance scorecard for any industry you must consider what that business deems most important within the four perspectives discussed previously. For the purpose of this case study, you'll focus on a marketing firm. Consider what objectives this firm could have to impact overall strategy. In the next sections, you'll focus on putting these objectives into the context of a balanced scorecard.

### Defining Objectives

Strategic objectives should be chosen carefully, because they are the key to understanding how each business perspective builds on one another. To illustrate, take a look at the objectives for each perspective in the marketing case study, shown in Table 12-1.

**Table 12-1:** Marketing Firm Objectives

PERSPECTIVE	OBJECTIVES
Learning and Growth	Provide training opportunities for marketing team
	Provide a job competency test for current employees
Internal Business Process	Leverage hosted e-mail marketing services
Customer	Establish a local retail presence
	Be customers' first choice for marketing services
Financial	Increase gross profit

From the learning and growth perspective, the marketing team will have training opportunities so they can continue to update their skills with the latest techniques used by the most successful firms. Current employees will also take a competency test to ensure they are the right people for the job.

For a marketing firm, technology is integral to making the business run smoother. The next objective will help the group's efforts from an internal business process perspective by leveraging new e-mail marketing services. Taking advantage of this technology will lessen the effort needed internally for producing e-mail campaigns.

The customer perspective focuses on making customers aware of the business and making those that are aware of the business happy with it. The marketing

company plans to accomplish this with local events to help promote brand awareness and by conducting surveys to see if the company is really the public's first choice for marketing services.

From a financial perspective, the marketing firm has a much more traditional objective. Like most companies, except non-profit organizations, they would like to increase gross profit each year.

Now, consider the entire list of objectives together in the following statement: The marketing firm will utilize its well-trained employees to leverage a new hosted service for its e-mail marketing campaigns. These campaigns should make their customers think of the marketing firm first, above its competitors, which should lead to an increase in gross profit for the company.

Figure 12-1 shows a strategy map, which details how the marketing firm's objectives build on each other to reach the ultimate goal of increasing gross profit. Creating a strategy map can clarify the purpose of a balanced scorecard to the business.

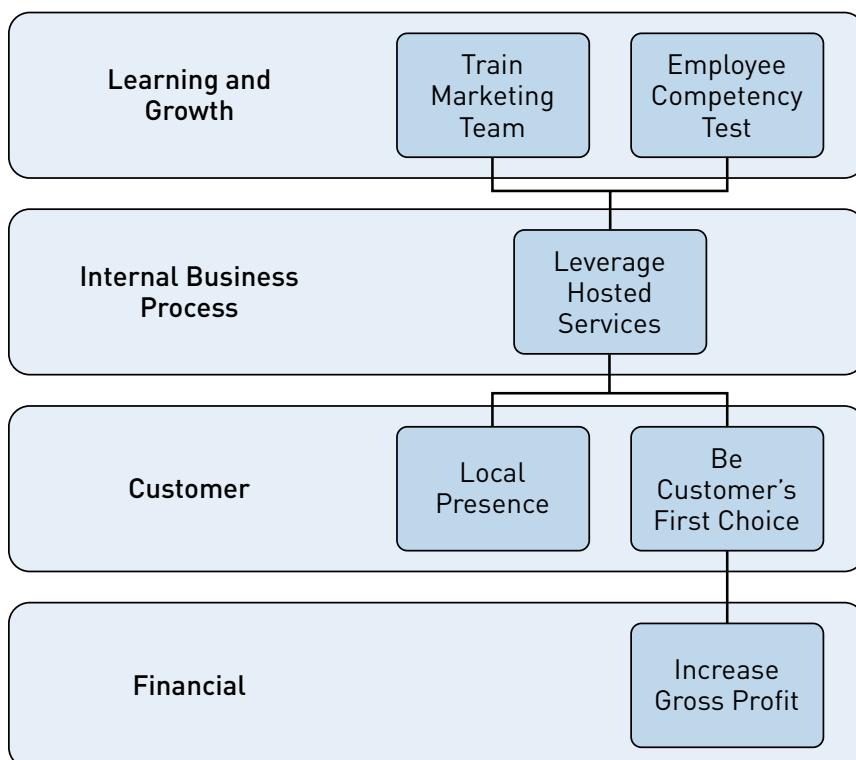


Figure 12-1

## Selecting Metrics

With the objectives for the balanced scorecard defined, you can now look at how to measure performance against these objectives. Measuring the success of performance initiatives is the most important aspect to consider when planning a balanced scorecard. Without the ability to quantify the results of your work, you'll find it difficult to justify the time spent on the efforts. Select metrics that help identify the success or failure of your objectives in each perspective.

You should have one or more metrics for each of the objectives on your balanced scorecard but, as discussed previously, be wary of too many metrics. While it may seem like a great idea to have as many measures as possible to analyze your business, too many can actually create problems. When this happens, it might seem as though every metric has the same importance, and it becomes difficult to focus your attention on addressing the critical few that can really impact a business turnaround.

Table 12-2 is an updated view of where you are in the planning process for this case study, this time providing measures for each objective.

**Table 12-2:** Marketing Firm Measures

PERSPECTIVE	OBJECTIVE	MEASURES
Learning and Growth	Provide training opportunities for marketing team	Training Days Available
		Average Training Days Used
	Provide a job competency test for current employees	Percentage of Employees Taking Tests
		Average Test Scores
Internal Business Process	Leverage hosted e-mail marketing services	Percentage of Customers on Hosted Service
Customer	Establish a local retail presence	Community Events
	Be customers' first choice for marketing services	Percentage of Brand Awareness
Financial	Increase gross profit	Gross Profit
		Profit Margin

In the learning and growth perspective you may have a couple measures for each of the objectives. One objective is to provide job competency testing, and you could measure the success of this testing a couple ways. The first metric is

total percentage of employees who have taken the test. This will help ensure each employee is actually attempting the test. The second metric will reflect the actual test scores. It is not only important that the employees take the test but that they also perform well on it.

The other learning and growth objective is to provide more training opportunities for the marketing team. The metric in this case is the number of days that are made available to employees for training opportunities. Measuring the number of days provided to employees is great, but are the employees really using the time provided to them? The company wants well-trained employees, therefore a second metric measures the average number of training days taken.

Looking at the internal business process perspective, there is just one objective: to leverage new hosted e-mail marketing services. Doing so will streamline e-mail campaigns, but it takes time to migrate customers to using it. Therefore, the metric for this objective is the total percentage of customers who are on the hosted e-mail service. The closer this metric is to one hundred percent, the smoother the business runs.

For the customer perspective, both objectives focus on awareness of the company in some way. One is to establish a local retail presence by running local events, which can be measured by the number of events run each year. The other objective is to be the first firm that customers and potential customers think of when they need marketing services. The way to measure this is through public surveys that the company has already implemented. Through these surveys the company gathers the number of survey takers that are aware of the company. This metric gauges brand awareness.

Finally, from a financial perspective the company has a simple but important objective of increasing gross profit. The metrics are also rather straightforward in this case: gross profit and profit margin. With a proper accounting system, this information should be easy to gather.

These measures will help the marketing firm objectively quantify the work it has done for each initiative. In the next section you focus on choosing targets, which helps you analyze the actual success of these initiatives.

## Choosing Appropriate Targets

Selecting measures is only half the job. To truly define success or failure with a balanced scorecard, you must select targets against which to compare these measures. This can be a difficult process because no one wants their work to be seen as falling short of preset goals.

You should choose targets that are attainable but that stretch the efforts of the people leading the company's initiatives. For example, if you have a growing company and one of your measures is profit, then you might set the target at 10 percent above the previous year. For a growing company, this is likely obtainable but provides a challenge.

Table 12-3 provides a view of the targets for each perspective in the marketing firm case study.

**Table 12-3:** Marketing Firm Targets

PERSPECTIVE	OBJECTIVE	MEASURES	TARGETS
Learning and Growth	Provide training opportunities to marketing team	Training Days Available	15
		Average Training Days Used	10
	Provide a job competency test for current employees	Percentage of Employees Taking Tests	95%
Internal Business Process		Average Test Scores	85%
	Leverage hosted e-mail marketing services	Percentage of Customers on Hosted Service	90%
Customer	Create a local retail presence	Community Events	4
	Be customers' first choice for marketing services	Percentage of Brand Awareness	70%
Financial	Increase gross profit	Gross Profit	\$1.5 Mil
		Profit Margin	40%

From a learning and growth perspective, you might need to reach a goal of 95 percent of employees completing the competency test and returning an average test score of 85 percent. You might also set a goal of making 15 days of training opportunities available to your employees, coupled with a goal of them taking advantage of at least 10 days of training on average.

The internal business process perspective is focused on migrating to a hosted e-mail service and so your goal might be to get 90 percent of your customers on the new service by the end of the year. Making this goal would allow the company to be much more efficient with e-mail marketing campaigns.

From the customer perspective, your emphasis is on brand awareness. You might target four local events to help grow the awareness of your company. You might also want to score 70 percent or higher on the survey of brand awareness—a good indication that your company's brand is doing well.

Finally, from a financial perspective, your goal might be to achieve a gross profit of \$1.5 million this year with a profit margin of 40 percent.

Remember, each of these measures continues to build off one another, so as you meet your goals for learning and growth, you will be helping your internal business processes run more smoothly, which will make your customer happier, and then your finances will flourish.

With these targets now defined, your planning process is almost complete. The next step is to bring all the components together and define a visual cue called an indicator.

## Bringing the Balanced Scorecard Together

Before you can start pulling together the data and placing it on a report, there is one last component that must be defined in the planning process. Traditionally, scorecards use indicators to show the report viewer whether the company met its goal. There are a variety of indicators available in Reporting Services, as shown in Figure 12-2, and you must define which indicator is shown and when.

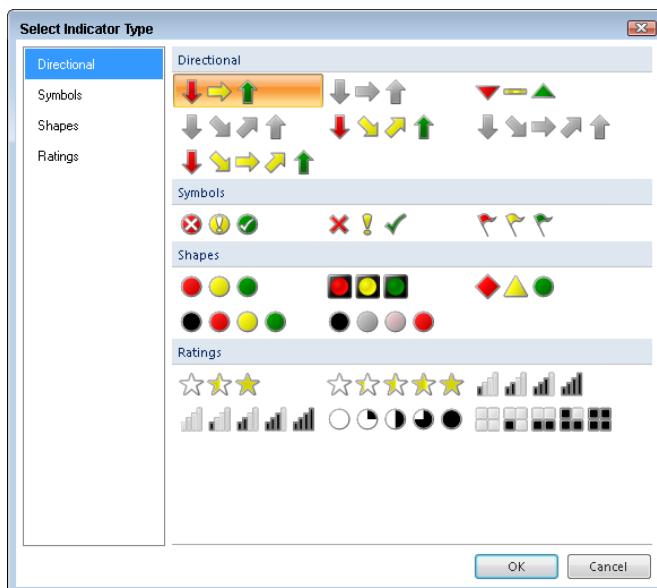


Figure 12-2

For example, one of your metrics is profit and your target is \$1.5 million. How do you want the indicator to display it when you meet this goal? How do you want the indicator to display it when you miss the goal? You must define ranges so the proper indicator is displayed under different scenarios.

Think about your profit metric again. You may decide that if the company reaches 100 percent of the target then the indicator should display green. Considering these targets were stretch goals, you may decide to make 90 percent of the target display the yellow indicator and everything below 90 percent displays a red indicator.

Creating these ranges enables you to have total control over the visual perception that the report conveys. At a glance, if you see a lot of red indicators then you know you're not doing well, but if you see green indicators then you know you're doing well. These ranges are most often defined by the business and should be collected during the requirements gathering process. You will walk through the configuration of an indicator in the section "Building the Balanced Scorecard."

## Wrangling the Data

As you can imagine, pulling together all the data required for a balanced scorecard is no small task. Often it requires multiple databases, files, and other types of data sources to bring together all the necessary fields for the report.

This design process can be lengthy and is generally driven by IT. A technical team creates an extraction process and brings the data into a newly designed database called a *data warehouse*. A data warehouse is a database that is used for reporting and analytics. It serves as a central repository for all the possible data sources mentioned previously. The data is usually organized in a way that makes life simpler for the report writer, because there are often fewer tables than traditional transactional databases making database relationships simpler to navigate when writing queries.

Designing and loading a data warehouse is out of the scope of this book, but as you go through the process of finding the data necessary for your balanced scorecard, you will likely find that building a data warehouse is a logical next step for organizing your data.

## Building the Balanced Scorecard

---

With the planning stages now complete, you are ready to turn your attention to the actual development of the balanced scorecard within Reporting Services. Building scorecards using native tools in Reporting Services is a fairly new concept. In fact, only since the release of SQL Server 2008 R2 are developers provided the indicator tool that enables them to properly represent scorecards. Prior to that, a report developer had to use images and write Reporting Services expressions to control which image would show that the company reached or didn't reach its goals.

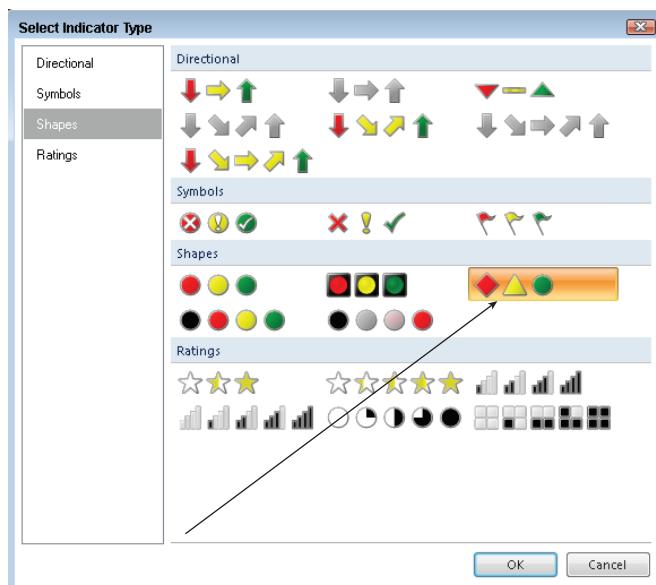
This section walks you through the steps required for pulling together the metrics you planned in the earlier part of this chapter to design a balanced scorecard in Reporting Services. As mentioned, bringing together all the data that is required for your balanced scorecard is quite a feat, and is out of the scope of this book. A query with all the columns and data required to build out this example is provided for you on the downloads page for this book at Wrox.com.

Reporting Services' design tool has all the necessary components built in to enable you to develop impressive scorecards. To build out the marketing firm balanced scorecard, follow these instructions:

1. Open SQL Server Data Tools and create a new Report Project.
2. Right-click the Reports folder and select Add ➔ New Report Item. Name the report **Balanced Scorecard**.
3. Create a Shared Data Source to the AdventureWorksDW2012 sample database and then add the Shared Data Source to the report in the Report Data pane.
4. In the Report Data pane, right-click the Datasets folder and select Add Dataset.
5. Select Use a dataset embedded in my report and the data source defined earlier.
6. Find the query provided to you for this chapter at [www.wiley.com/go/performancedashboards](http://www.wiley.com/go/performancedashboards) and place it in the Query window for this dataset. Once complete, click OK.
7. From the Toolbox, drag a Table onto the design surface of the report.
8. Place the Perspective field above the Details group in the Row Groups section of the Grouping Pane.
9. Next, place the Objective field between the Perspective group and the Details group in the Row Groups section of the Grouping Pane. Figure 12-3 shows steps 8 and 9 completed.

**Figure 12-3**

10. From the Report Data pane, drag the field `MeasureName` from the dataset onto the table and to the right of Objective.
11. Open the Toolbox and drag a Rectangle on the table next to the `MeasureName` column. This will be used to help keep the size of the indicator image from growing and appearing stretched.
12. Next, drag the Indicator tool inside the Rectangle.
13. You will be prompted to select the indicator type. Choose the red diamond, yellow triangle, and green circle indicator found under the Shapes category, then click OK, as shown in Figure 12-4.



**Figure 12-4**

14. Resize the objects so they are appropriately proportioned in the cell.
15. Right-click the indicator in the table and select Indicator Properties.
16. On the Values and States page, click the *fx* button next to the Value property to build an expression that will represent the indicator value.
17. Use the following expression in the Expression window and then click OK, as shown in Figure 12-5:  

$$=Fields!MeasureValue.Value/Fields!MeasureTarget.Value$$
18. Change the States Measurement Unit to Numeric and set the Indicator states property so that Red starts at 0 and ends at 0.89, Yellow starts at 0.9 and ends at 0.99, and Green starts at 1 and ends at 1.5. Click OK when these steps are complete. Figure 12-6 shows this step completed.

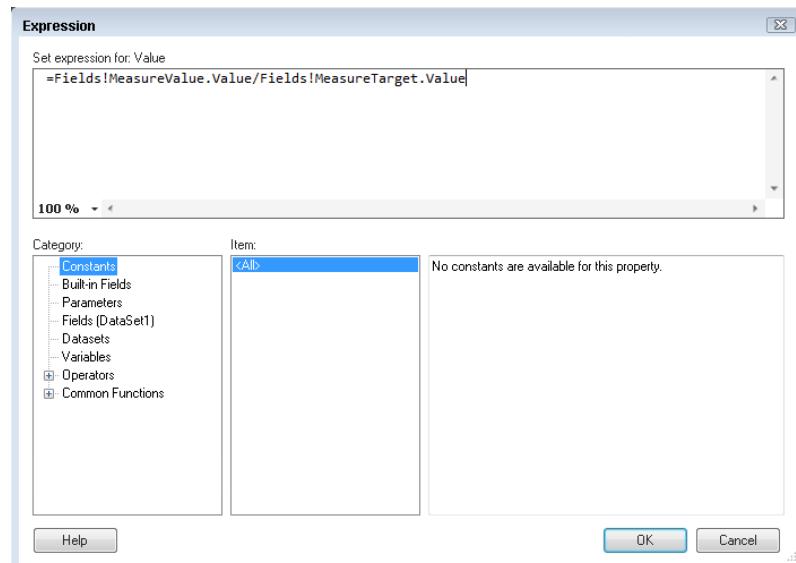


Figure 12-5

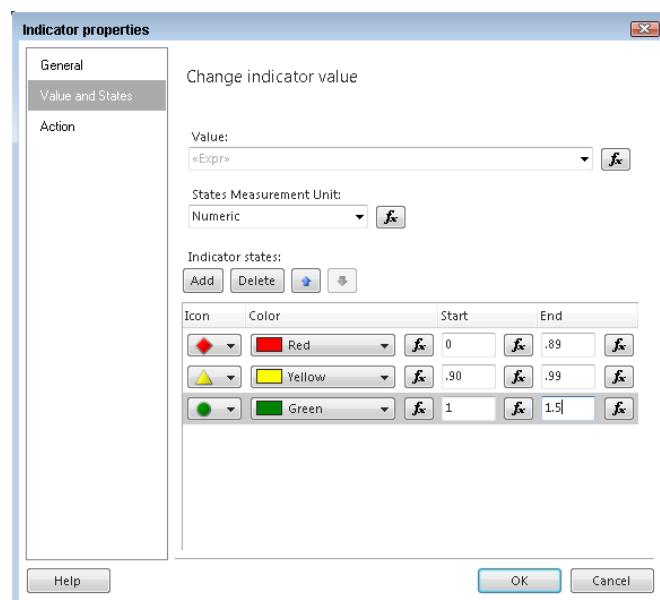
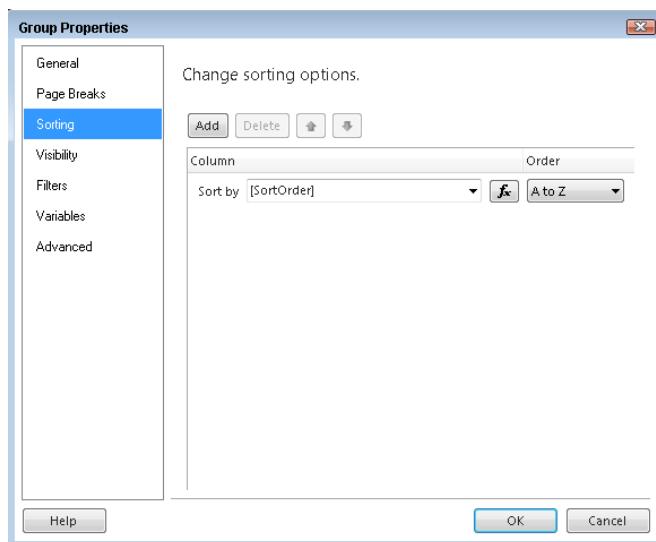


Figure 12-6

19. To ensure your data is ordered properly, right-click the Perspective group and select Group Properties.

20. On the Sorting page, change the Sort by field from Perspective to SortOrder, and then click OK. See Figure 12-7.



**Figure 12-7**

21. Delete the extra column in the table that has no fields in it.  
 22. Preview the report to see the final results. You may find it necessary to resize some of the other columns to get the proper aesthetics, but once complete the report should look like Figure 12-8.

Perspective	Objective	Measure Name	
Learning and Growth	Current Employees Take Marketing Competency Test	Test Taken	
		Average Test Scores	
	Provide Training Opportunities for Marketing Team	Training Days Available	
		Average Training Days Used	
Internal Business Process	Leverage Hosted Email Marketing Service	Customers on Hosted Service %	
Customer	Be Customers First Choice for Marketing Services	Brand Awareness %	
	Have a Local Presence Near Retail Stores	Community Events	
Financial	Increase Gross Profit	Gross Profit	
		Profit Margin	

**Figure 12-8**

## Summary

---

In this chapter you walked through a case study of a marketing firm that was looking to grow its business. The planning process can be lengthy, but through it you defined objectives that set the stage for the entire scorecard. You need a way to measure how your initiatives are being executed, and this is done through metrics. Too many metrics for each objective can be a problem, however, because it makes it difficult to know which are the most important. Once metrics are set, there must be a way to define the success of each objective, which is accomplished by setting targets that stretch your efforts and help grow the business. During the design of your balanced scorecard, you also learned that setting indicator ranges is important to provide visual cues to the viewer of the report on the overall success of the company.



## Decision-making Charts

By now, you've identified your strategic objectives, selected the metrics, and started designing and implementing performance dashboards, balanced scorecards, and other analytical reports to support decision-making within your organization.

This chapter continues to build on what you've learned as you are introduced to four types of powerful decision-making charts that can help your audience make better, faster, and more efficient business and organizational decisions.

### Box-and-whisker Charts

---

It's likely that at some point you have come across a *box-and-whisker chart* in one form or another. They are common for charting stock price movement throughout the day and are used to represent a series of values through basic statistics using quartiles. These charts are a form of descriptive statistics and go by many names including *box plots* and *candlestick charts*.

## Box-and-whisker by Example

To better explain this type of chart, five basic statistics must be defined first:

- **Min:** The minimum value found in a set of values; used to plot the bottom of the whisker.
- **Max:** The maximum value found in a set of values; used to plot the top of the whisker.
- **Median:** The center or midpoint of a sorted set of values, which divides the set of values in half. If there is an even number of values, the two center points are averaged to calculate the median value. The median can be plotted as either a line or a point and should fall within the box.
- **Upper Quartile:** Using a sorted set of values, the center point of the set of points to the right of the median point; used to chart the top of the box.
- **Lower Quartile:** In a sorted set of values, the center point of the set of points to the left of the median point; used to chart the bottom of the box.

**NOTE** The *interquartile range* is an alternative to using the median value and is less influenced by the minimum and maximum values. To calculate, subtract the upper quartile from the lower quartile.

Using those definitions, the following example better illustrates each statistic. Assume the following set of values:

12, 17, 22, 27, 32, 37, 42, 48, 54, 61, 68, 73, 78, 80, 83

Since the set of values is already sorted from smallest to largest, finding the minimum and maximum values is a trivial task.

- Minimum Value: 12
- Maximum Value: 83

To find the median, follow a few simple steps:

1. First you must find the midpoint of the set. The example set has 15 data points, making the midpoint the eighth data point, or 48.
2. Next, calculate the quartiles in similar fashion. Starting with the upper quartile, take the seven data points to the right of the median value and find the midpoint.

54, 61, 68, 73, 78, 80, 83

3. After resolving the upper quartile as 73, repeat the exercise using the seven data points to the left of the median (48) to find the lower quartile, or 27.

12, 17, 22, 27, 32, 37, 42

Using these basic statistics, you can construct a box-and-whisker chart, as shown in Figure 13-1.

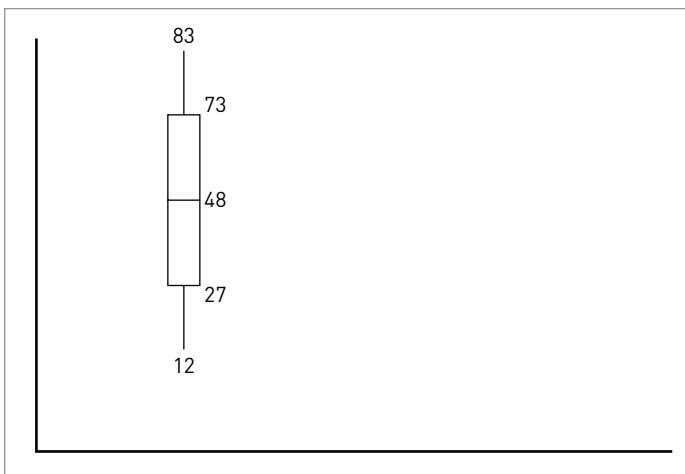


Figure 13-1

**NOTE** It is common to remove outliers and plot them separately using a point or marker.

## Building a Box-and-whisker Chart

Out of the box, Reporting Services supports three primary forms of box-and-whisker charts: stock, candlestick, and box plot. Although each chart has subtle differences both in chart format and in the data points plotted, the stock and candlestick charts are very similar.

The stock and candlestick charts support four data points. The open and close (typically used to represent stock prices) corresponds with the quartile values while the high and low corresponds to the minimum and maximum values. The box plot chart goes further by adding data points for the median and mean values, which are charted as dashed and dotted lines through the box, respectively.

To build a box-and-whisker chart, you will first need to calculate the required statistics over your set of data for some period of time. For a long time, T-SQL had limited abilities beyond the `MIN`, `MAX`, and `ROW_NUMBER()` functions to help you calculate these statistics. With the introduction of the `PERCENTILE_CONT()` function in SQL Server 2012, these statistics can now be done. An even better option than manually calculating these set statistics though is to use a tool like SQL Server Analysis Services to facilitate the calculation of these statistics.

**NOTE** The PERCENTILE\_CONT() function calculates a continuous distribution over the values found within a target column. The query below provides an example based on the Adventure Works database that finds the 25th, 50th (Median), and 75th percentile for employee salaries.

```
SELECT DISTINCT Name AS DepartmentName
    , PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY ph.Rate)
        OVER (PARTITION BY Name) AS Percentile25th
    , PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY ph.Rate)
        OVER (PARTITION BY Name) AS Percentile50th
    , PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY ph.Rate)
        OVER (PARTITION BY Name) AS Percentile75th
FROM HumanResources.Department AS d
INNER JOIN HumanResources.EmployeeDepartmentHistory AS dh
    ON dh.DepartmentID = d.DepartmentID
INNER JOIN HumanResources.EmployeePayHistory AS ph
    ON ph.BusinessEntityID = dh.BusinessEntityID
WHERE dh.EndDate IS NULL;
```

After you identify, source, and prepare your data, you are ready to build your chart using the following steps:

1. Create a new report. Set up your data source and dataset using the following source query, and then add a Chart Control to the Report Designer surface.

```
SELECT '3/21/2013' ReportDate, 3 Min,
      5 LowerQuartile, 10 Median, 15 UpperQuartile, 20 Max
UNION ALL
SELECT '3/22/2013', 1, 4, 7, 9, 13
UNION ALL
SELECT '3/23/2013', 10, 13, 17, 19, 23
UNION ALL
SELECT '3/24/2013', 2, 5, 7, 9, 10
```

2. In the Select Chart Type dialog, select the box plot type from the Range chart category (shown in Figure 13-2).
3. In the Chart Data region, select the ReportDate column as the grouping field in the Category Groups. This will result in dates being plotted along the x-axis.
4. Add the maximum column from your dataset to the Values region. This causes a new series to be created (see Figure 13-3) with placeholders to map the High whisker (max), Low whisker (min), High box (upper quartile), Low box (lower quartile), Mean, and Median.

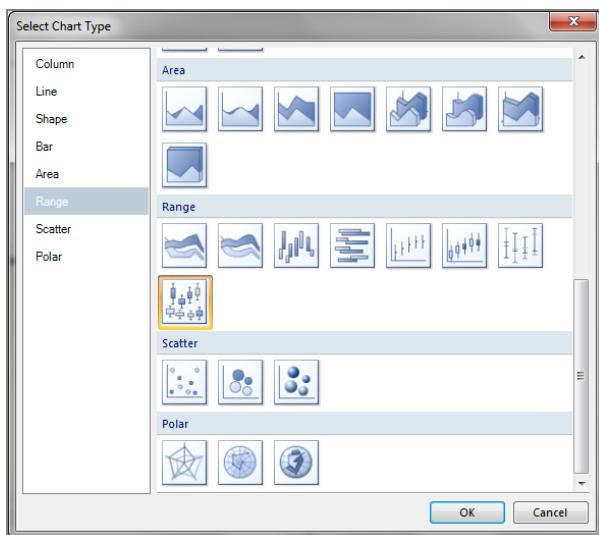


Figure 13-2

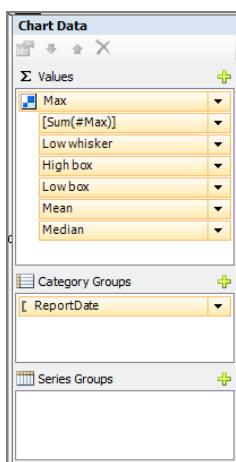
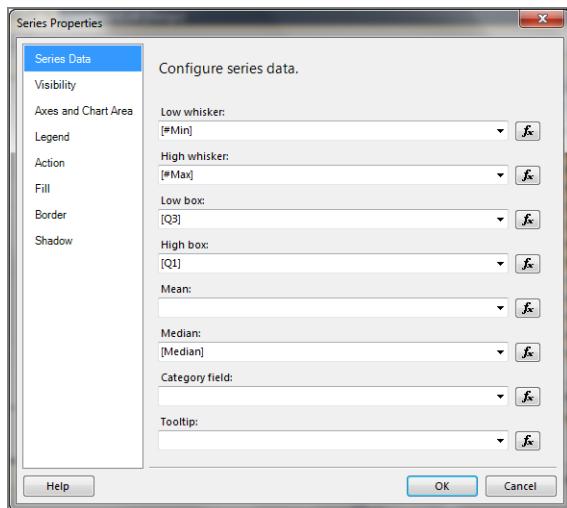
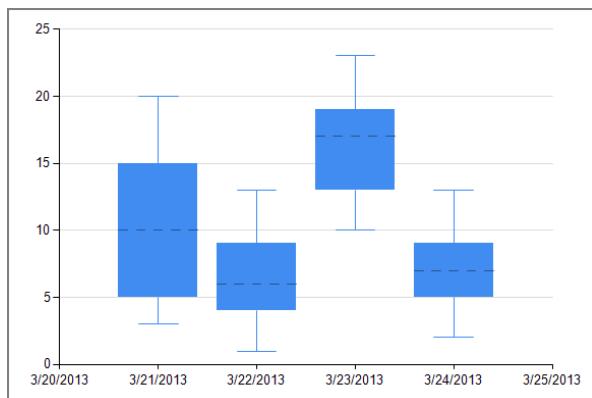


Figure 13-3

5. Open the Series Properties dialog and update the mapping for the Series Data as illustrated in Figure 13-4. Note that if your dataset has multiple rows for each data point, it is necessary to remove the `Sum()` function that is added automatically when mapping fields through the dialog. Click OK when you are finished.

**Figure 13-4**

The basic configuration of the chart is complete. If you wish, you can adjust the format, including making changes to the y-axis and adding or removing various titles and legends. When finished, preview the resulting box plot (shown in Figure 13-5).

**Figure 13-5**

This example can be extended easily to include additional information such as an interquartile range or outlier points. The interquartile range is the most straightforward, since it could be plotted as either the median or mean value within the series data. Outliers, on the other hand, require the following additional steps:

1. Add a new series to the chart by clicking the green plus icon in the Value region of the Chart Data window. Map the new series to the Outlier column in your dataset.

2. Change the Chart type for the new series from box plot to candlestick and then open the Series Properties window.
3. Select and configure a Marker type and Marker color on the Markers page, as shown in Figure 13-6.

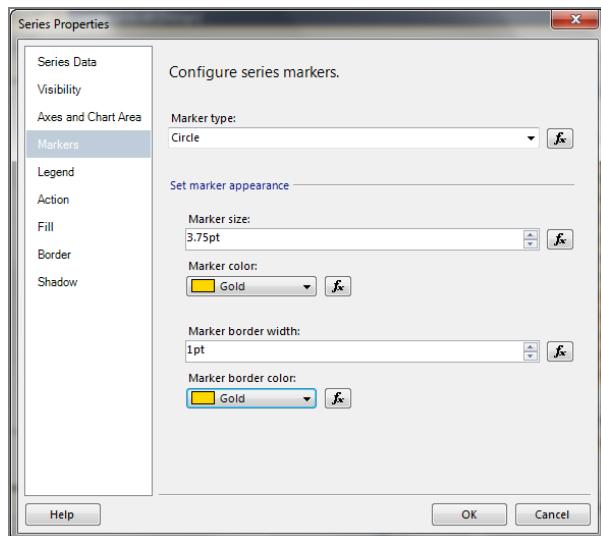


Figure 13-6

4. On the Fill page, change the Color option to No color. This hides everything but the outlier point marker. Click OK to finish the changes.

Figure 13-7 is a sample of the final result. Your preview or finished chart may vary and is dependent on the dataset used.

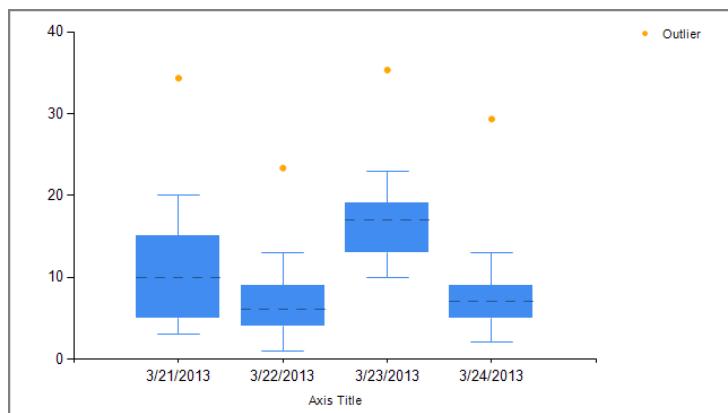


Figure 13-7

## Pareto Charts

---

In the 19th century, Vilfredo Pareto observed that 80 percent of all the land in Italy was owned by only 20 percent of the people. This and supporting observations led to the discovery of a very powerful microeconomic principle known formally as *Pareto* or *power law probability distribution*. This concept later spawned the *Pareto Principle*, or more informally, the *80-20 Rule*.

### 80-20 Rule

Although Pareto was primarily interested in income distributions through society, his principle has been validated over time as a useful decision-making tool for focusing resources on the areas where they will have the greatest impact. The 80-20 Rule can be generalized into the following examples:

- 20% of customers account for 80% of total sales revenue.
- 80% of complaints come from 20% of customers.
- 80% of sales are for 20% of your products.

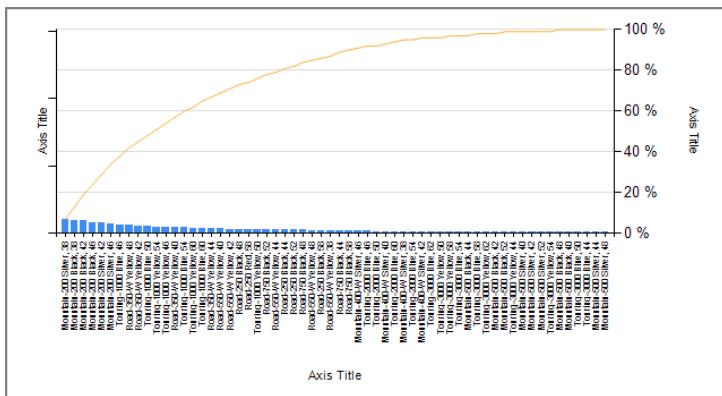
Each of these examples points out opportunities that exist to channel resources to customers and products to maximize benefits. This is a powerful tool for those responsible for making business and organizational decisions.

**NOTE** For Pareto's Principle or the 80-20 Rule to hold true, a sufficiently large sample of data must be available.

### Pareto Chart

Building on the Pareto Principle, a *Pareto Chart* visualizes the most important items among a set of factors or values for a given measurement. It consists of both bar (column) and line charts to graphically illustrate two series of data:

- **Bar Chart:** The bar chart represents each factor's value for a given measure. In the case of product sales, the set of products would be plotted along the x-axis, and their sales volume or revenue on the y-axis. The set of products are sorted by their value in descending order such that the products with the largest values are first.
- **Line Chart:** Using the product sales example, the line chart graphs each product's contribution to the percent of total. The line starts at 0 and approaches 100 percent as it moves to the right on the y-axis, as shown in Figure 13-8.



**Figure 13-8**

Together, these two charts enable the decision maker to quickly identify the factors, or products in this example, whose impact is greatest on the chosen measurement.

## Building a Pareto Chart

While Reporting Services is capable of rendering Pareto Charts, you will notice that when using the Chart Control there is no option to select Pareto as the Chart Type. Instead, a bar chart is used with several extra steps needed to implement this chart type.

Start by preparing the data for use in your report. The dataset required can be as simple as a list of products and their total sales, as shown in the following query based on the Adventure Works Data Warehouse:

```

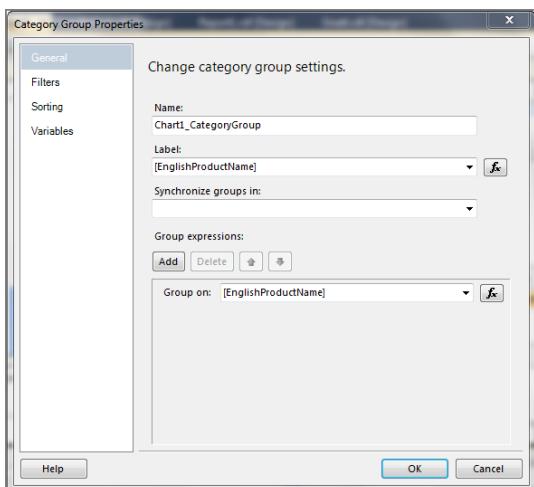
SELECT
    SUM(s.SalesAmount) AS TotalSales,
    p.EnglishProductName
FROM FactInternetSales s
JOIN DimProduct p ON (s.ProductKey = p.ProductKey)
JOIN DimProductSubcategory sc
    ON (p.ProductSubcategoryKey = sc.ProductSubcategoryKey)
JOIN DimProductCategory c
    ON (sc.ProductCategoryKey = c.ProductCategoryKey)
JOIN DimDate d ON (s.OrderDateKey = d.DateKey)
WHERE c.ProductCategoryKey = 1
AND d.MonthNumberOfYear = 4
AND d.CalendarYear = 2008
GROUP BY p.EnglishProductName
ORDER BY SUM(s.SalesAmount) DESC

```

With your data in hand, create your report and then perform any set-up required, including defining the data source connection and dataset.

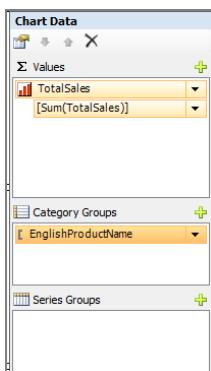
Next, follow these steps to build the Pareto Chart:

1. Add a Chart Control component from the Toolbox to the Reporting Services design surface. Since the Bar or, more accurately, the Column chart is the base of this chart, select it from the Chart Type dialog.
2. The products in this example should be plotted along the x-axis. To make this change, map the EnglishProductName column to the Category Group by right-clicking and opening the Category Group Properties from the Chart Data window, using the settings displayed in Figure 13-9. Even though this example pre-sorts the data in the dataset source query, it is not strictly necessary to either pre-sort the data or configure the sort order on the Category Group.



**Figure 13-9**

3. Because you are charting total sales per product, map the TotalSales column to the Values region of the Chart Data window, as shown in Figure 13-10.



**Figure 13-10**

4. If you were to preview the chart at this point, a standard bar (column) chart is rendered, as shown in Figure 13-11. To convert the chart into a Pareto Chart, the Chart Series properties must be adjusted. Select the Chart Series either in the Chart Data window or by clicking on the bars within the chart. Then press the F4 button.

**NOTE** You will need to format the horizontal chart axis to duplicate the layout of the Figure 13-11. In the horizontal chart axis properties, change the interval to 1.

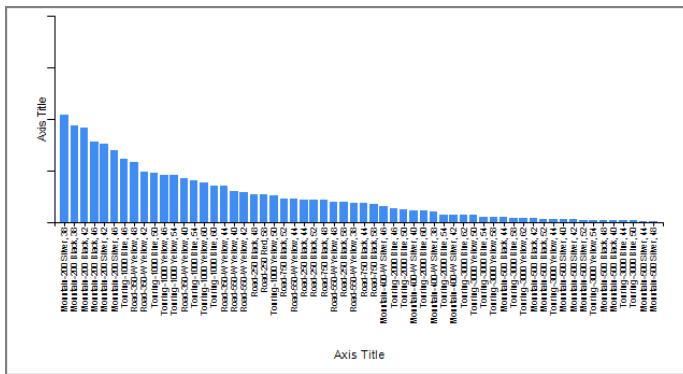


Figure 13-11

5. In the Properties window, find the `CustomAttributes` property. If you are viewing the properties using the categorized view, as shown in Figure 13-12, the `CustomAttributes` property is found under the General grouping. Expand the `CustomAttributes` property collection.

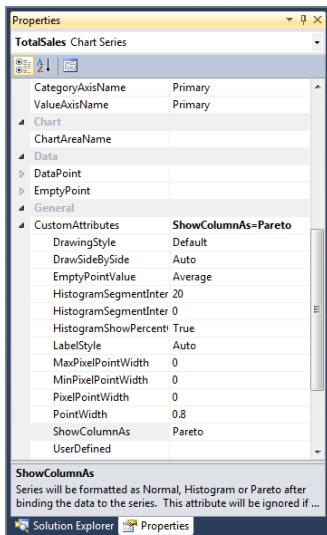


Figure 13-12

6. Find the ShowColumnAs property and change the value from Normal to Pareto.
7. At first glance, the change does not appear to have made any noticeable difference in the design view. The magic behind this property occurs when you preview the report, as shown in Figure 13-13.

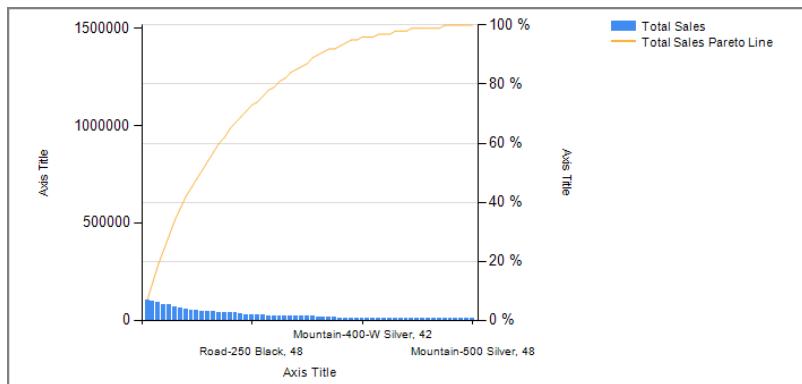


Figure 13-13

Although diving into the inner-workings of how Reporting Services handles the ShowColumnAs property is beyond the scope of this chapter, it's important to point out that the line chart series and the secondary y-axis are added automatically for you. Not all of these parts can be configured, but those that can are only accessible by digging into the properties for each portion of the chart.

## Variance Charts

The goal of each chart discussed to this point is to aid the decision maker by providing meaningful and actionable information, and the *variance chart* is no different. This niche chart is both a powerful and useful decision-making tool when the business objective is the management of budgets and/or sales forecasts that consist of targeted or budgeted values and actual performance values.

Although you could simply plot target and actual values on a chart, as in Figure 13-14, the result would force your user to mentally tally or calculate the variance for each data point. This is a tedious and error-prone exercise that can hinder rather than facilitate decision making.

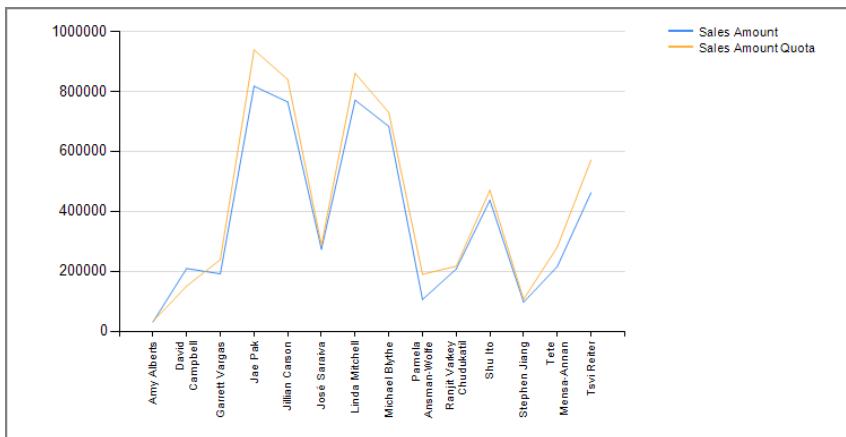


Figure 13-14

The variance chart calculates and plots the difference between the budgeted and actual value as either a percentage or a difference between the two. The x-axis is used as a visual divider to quickly enable the user to see those areas that are either over or under budget, or over or underachieving, as the case and measures dictate.

Budget and sales forecast/quotas are the data basis of the variance chart. In this example, employee sales quotas and sales data for Adventure Works from the following query are used:

```

SELECT
    s.EmployeeKey,
    e.FirstName + ' ' + e.LastName AS Name,
    SUM(SalesAmount) AS SalesAmount,
    MIN(SalesAmountQuota) AS SalesAmountQuota,
    SUM(SalesAmount) - MIN(SalesAmountQuota) AS Variance,
    SUM(SalesAmount)/MIN(SalesAmountQuota) AS VariancePercentage
FROM FactSalesQuota q
JOIN FactResellerSales s ON (q.EmployeeKey = s.EmployeeKey)
JOIN DimDate d ON (s.OrderDateKey = d.DateKey)
JOIN DimEmployee e ON (s.EmployeeKey = e.EmployeeKey)
WHERE q.CalendarYear = 2007
AND q.CalendarQuarter = 1
AND d.CalendarYear = q.CalendarYear
AND d.CalendarQuarter = q.CalendarQuarter
GROUP BY
    s.EmployeeKey,
    e.FirstName + ' ' + e.LastName
  
```

The example dataset calculates the variance both in terms of dollars and by percentage. Percentages are typically used when showing budgets versus actual values, while dollars or units are more advantageous when charting sales quotas or forecasts.

The following steps walk you through implementing a variance chart based on the preceding dataset:

1. Add a new Chart Control from the Toolbox to the Reporting Services design surface and then select the Bar (Column) chart type.
2. Open the Chart Data window and set the Employee Name as the Category Group field.
3. Set the Variance column as the Data value for the Series in the Values region.
4. Preview the chart (see Figure 13-15) to reveal that the x-axis is not set at zero as you have typically seen.

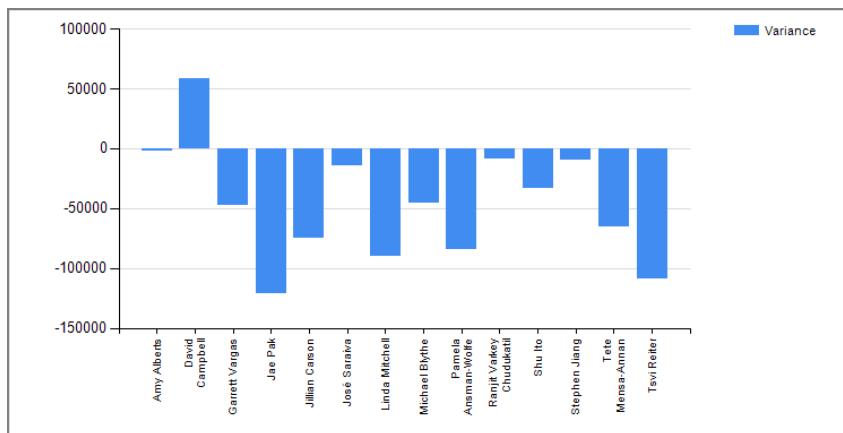


Figure 13-15

5. Next the x- and y-axes needs to be formatted so that the x-axis is anchored and crosses the y-axis at zero. Select the y-axis in the chart and then press F4 to open the Properties window, if it is not already open.
6. Find the `CrossAt` property in the Scale property category and change the value from `Auto` to `0`, as shown in Figure 13-16.
7. Now, preview the chart (see Figure 13-17) and you should see the x-axis crosses the y-axis at zero and the variance chart is complete.

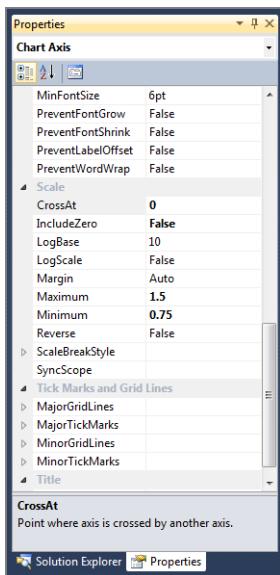


Figure 13-16

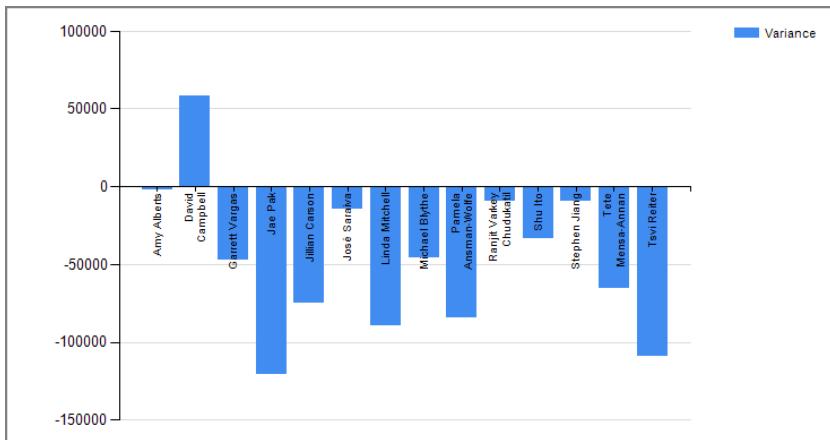
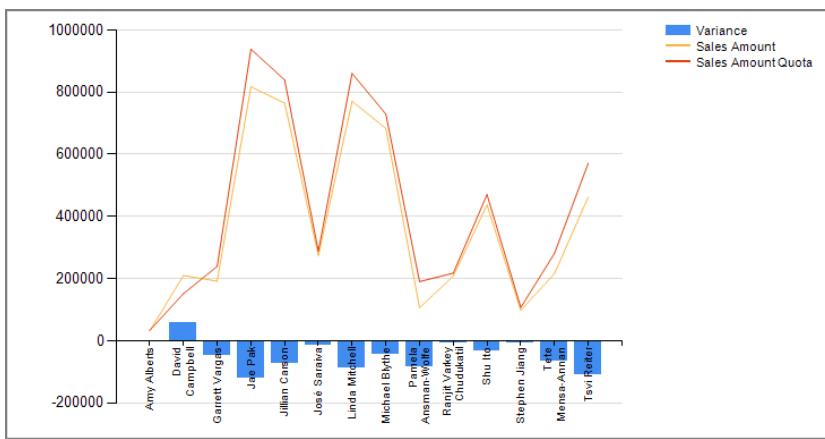


Figure 13-17

Further refinements to the variance chart can improve its utility:

- Restrict the y-axis scale, particularly when showing variances in percentages.
- Add a new series to the chart, as shown in Figure 13-18, when trends may be apparent. This is particularly important when plotting budget/forecast versus actual values across time. Use the Line chart type to avoid obscuring the meaning of the variance chart.

**Figure 13-18**

## Gantt Charts

*Gantt charts* were developed by Henry Gantt in 1910 as a method of visualizing a project schedule. This type of chart, although not the most common, is a useful tool when the dashboard audience consists of managers or project managers.

The foundation of the Gantt chart is charting project activities or tasks across time. Tasks are plotted on the y-axis and are typically sorted with those that occur first on top and those that occur last on bottom. The x-axis plots the time series and can take the form of specific dates, weeks, or months.

Microsoft Reporting Services supports the Gantt chart layout out of the box in the form of the Range bar chart. The data required to build this chart is straightforward, requiring only a set of tasks, start dates, and end dates.

Use the following steps to build a Gantt chart:

1. Create a new report. Then, add the report data source and a dataset that contains a list of tasks, start dates, and end dates. For this example, a simple, static dataset that mimics a book-writing schedule is used:

```

SELECT 'Chapter 1' AS Task, '1/08/2013' AS StartDate, '1/20/2013' AS EndDate
UNION
SELECT 'Chapter 2', '1/19/2013', '1/30/2013'
UNION
SELECT 'Chapter 3', '1/12/2013', '2/2/2013'
UNION
SELECT 'Chapter 4', '1/31/2013', '2/15/2013'
UNION
SELECT 'Chapter 5', '2/05/2013', '2/20/2013'

```

2. Add a Chart Control to the design surface and select the Range bar chart type.

3. In the Chart Data window, define the Category Group as the Task column.
4. Open the Category Group Properties and define the Sort by field on the Sorting page as startDate (see Figure 13-19) using an order of Z to A.

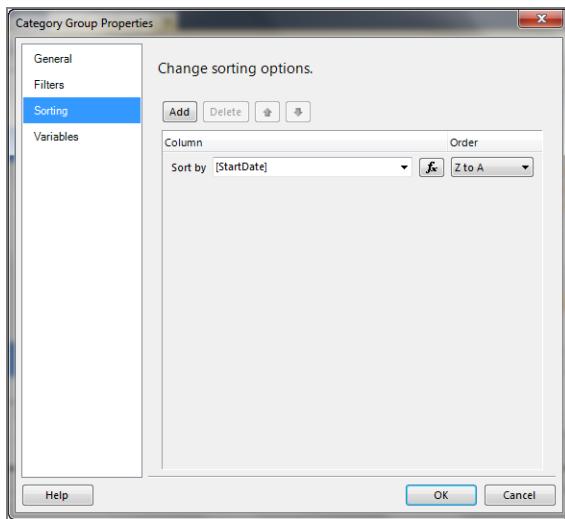


Figure 13-19

5. Add the `StartDate` column to the Value group and then open the Series Property window.
6. Map the Top and Bottom range values, as shown in Figure 13-20. Note that it is necessary to remove the `Count()` function.

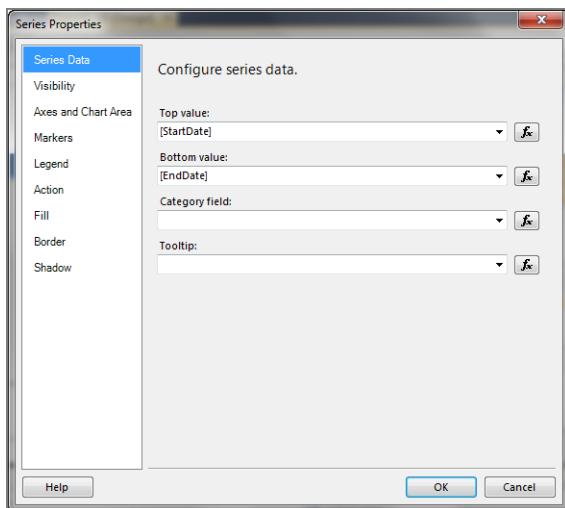


Figure 13-20

7. Delete the Chart legend, as it is not useful in this chart.

A quick preview of the chart (see Figure 13-21) reveals a rough Gantt chart. To clean up the formatting and condense the chart, apply the following formats:

- In the Vertical Axis Properties, hide the axis labels (Labels page).
- Add the Task Name to the Range bar. First, open the Properties window with the Chart Series selected. In the Labels category, set the `UseValueAsLabel` to True and then define the label as `=Fields!Task.Value`. Finally, you can alter the position of the label in the `CustomAttributes` using the `BarStyleLabel` property. This example sets the value to `Left`.

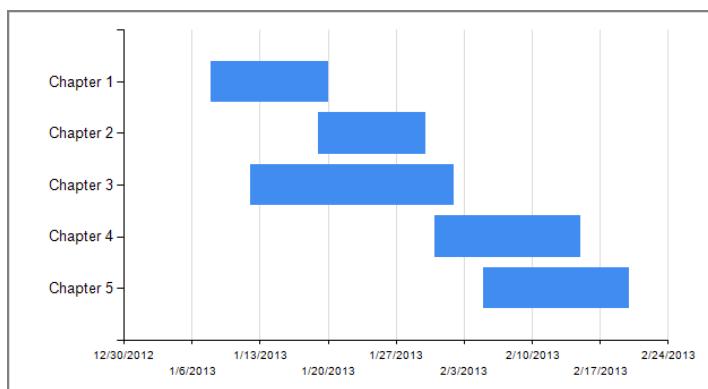


Figure 13-21

Figure 13-22 shows the resulting chart, which is more condensed and more suitable for inclusion as a component of a performance dashboard.

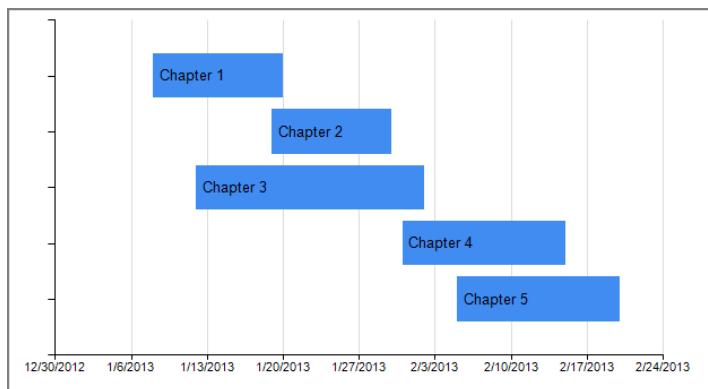


Figure 13-22

## Summary

---

In this chapter, you were introduced to box-and-whisker, Pareto, variance, and Gantt charts. All four of these charts are powerful and useful tools intended to aid and simplify the decision-making process. They will be valuable resources as you begin building and implementing both performance dashboards and balanced scorecards.



# Forecasting Trends and Data Smoothing

People rely on forecasts every day. From weather and traffic to those made for investment purposes, forecasts are a part of life. Businesses, regardless of size, sector or industry, rely on them as well for planning unit sales, managing inventory and workforce, budgeting, and estimating marketing campaign response.

In this chapter, you take a pragmatic first look at trend analysis. You learn how to smooth erratic data using three of the most common models to better identify trends before diving into forecasting implementations using both Microsoft Excel and SQL Server Analysis Services.

Before you get started, note that while Reporting Services itself does not have built-in forecasting ability, most of what is discussed can be accomplished using either Microsoft Excel or SQL Server Analysis Services. Neither experience nor sophisticated statistical tools beyond basic mathematics are required to be successful in this chapter.

## Smoothing Data

---

When data is erratic, trends can be difficult if not impossible to accurately spot. To overcome this issue, you can easily smooth erratic data to highlight any underlying trend that would be otherwise obscured. Three of the most common methods for smoothing erratic trends involve calculating either the average or

mean of some series of data points across a moving or rolling time period. The *rolling time period* is a window or range of dates that you will use to smooth data over and is an important feature to understand. Some of the more common rolling time periods used are 30-day, 3-month, 6-month, and 12-month. As you get started, take time to evaluate the data and select the correct window size for the trend or data movement you are attempting to observe.

Selecting the correct window size is completely relative to the task at hand and you should be careful not to select a window that is either too small or too large. For example, if you attempt to observe an annual or seasonal trend in sales for a retail organization, daily movements in the data may create too much noise, potentially obscuring any observable trend. Likewise, if you are interested in exploring hourly call center data, a 12-month window would likely obscure the hour-to-hour movements. Note that selecting the correct window size is critical as it enables the forecast to react to trends or changes.

In this section, you look at understanding and calculating simple moving averages, weighted moving averages, and exponential moving averages. Each method has its strengths and weaknesses, and understanding how each is built will help you select the most appropriate method when you begin building your forecast to identify the trend.

## Simple Moving Average

The *simple moving average* (SMA) is the most common and straightforward method or implementation used to forecast trends. It consists of nothing more than simple arithmetic in which a series of data points over time is added up and then divided by the number of time periods. The number you arrive at is a mean value for the rolling time period use.

To better illustrate this concept, take a look at the following formula and Table 14-1:

$$\text{Simple Moving Average (SMA)} = (\text{Jan} + \text{Feb} + \text{Mar}) / 3$$

**Table 14-1:** Simple Moving Average

DATE	SALES	SIMPLE MOVING AVERAGE
Jan 2012	300	
Feb 2012	175	
Mar 2012	220	232
Apr 2012	225	206
May 2012	230	225
Jun 2012	210	222

DATE	SALES	SIMPLE MOVING AVERAGE
Jul 2012	205	215
Aug 2012	200	205
Sep 2012	215	207
Oct 2012	230	215
Nov 2012	275	240
Dec 2012	290	265

Using a sliding window of 3 months, you simply take an average of the preceding two months' sales with the current month to arrive at the average value. Note that for the first two months you cannot calculate a value; this results in a lag. A *lag* occurs because of the shift or sliding window size and results in data movements typically being delayed by the same number of periods as the window size. Figure 14-1 shows how the simple moving average smooths the otherwise erratic sales data once it is charted.

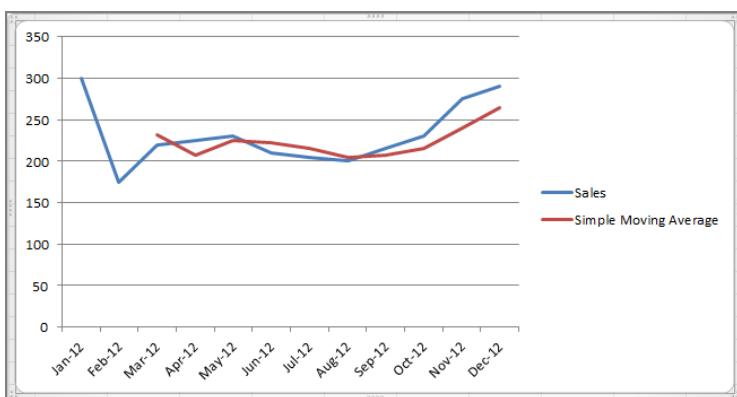


Figure 14-1

While easy to build and work with, the simple moving average is not without its shortcomings. The first and most obvious of these are the missing data points for the January and February periods and the cyclical lag, which is normally about half of the window size or about 1.5 months in this example. Other issues that stand out include:

- Data that is highly volatile or erratic will require a bigger window to smooth, such as 6, 9, or even 12 months instead of 3.
- Equal weight is given to all data points, meaning that the 300 units sold in January are weighted equally to 220 sold in March.

## Weighted Moving Average

In the previous section you saw the simple moving average is nothing more than a mean value of points within a sliding window. Equal weight is given to every point within the window whether it was the first point or the last, which causes a rather significant amount of lag (about half the window size).

To correct this issue, you can use the *weighted moving average* (WMA), which accounts for where the data point occurs within the window by assigning a weight to each time period's data value. The most recent data point is given a bigger weight than older points, where all weights must be equal to a sum of 1.

Continuing the example from the previous section, the math to calculate the weighted moving average for March using the weighting schema (3 + 2 + 1) requires that you divide by 6, and looks something like this:

$$\text{Weight Moving Average (WMA)} = ((3 \times \text{Mar}) + (2 \times \text{Feb}) + (1 \times \text{Jan})) / 6$$

Plug this formula into the example used previously (see Table 14-2) and you get a better idea of the smoothing effect of the weighted moving average.

**Table 14-2:** Weighted Moving Average

DATE	SALES	SIMPLE MOVING AVERAGE	WEIGHTED MOVING AVERAGE
Jan 2012	300		
Feb 2012	175		
Mar 2012	220	232	218
Apr 2012	225	206	215
May 2012	230	225	227
Jun 2012	210	222	219
Jul 2012	205	215	210
Aug 2012	200	205	203
Sep 2012	215	207	208
Oct 2012	230	215	220
Nov 2012	275	240	250
Dec 2012	290	265	275

When you chart this example as shown in Figure 14-2, you notice that the trends within the data react far more quickly and that you have effectively eliminated the lag that was present in the simple moving average.

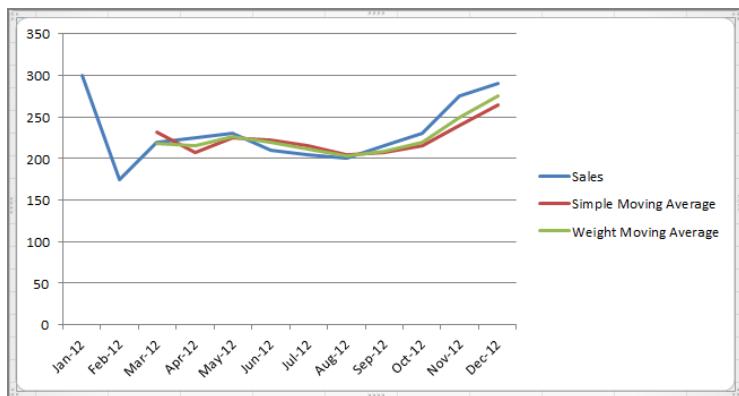


Figure 14-2

## Exponential Weighted Moving Average

Another tool in your tool belt to handle erratic data is the *exponential weighted moving average* (EMA). This formula also handles the weight of data points but goes further to isolate lag within the moving average. Calculating the exponential moving average is done in three steps:

1. Calculate the simple moving average for the first time period (March in the preceding example):

$$\text{SMA} = (\text{Jan} + \text{Feb} + \text{Mar}) / 3$$

2. Calculate or apply a smoothing factor, also called  $\alpha$  Alpha. This factor adjusts for error within the average. If you are calculating a large enough window use the following formula:

$$\alpha \text{ Alpha} = (2 / (1 + \text{Number of Time Periods}))$$

For small windows like that used in the example, you can manually use a factor between 0.1 and 0.3.

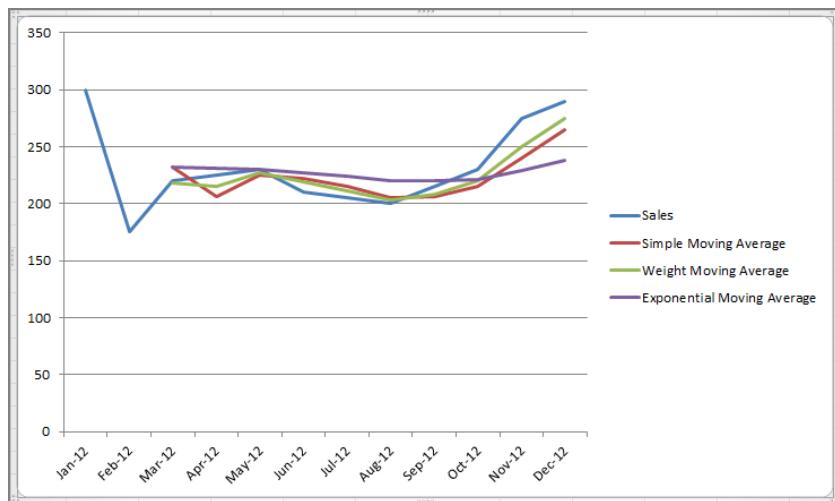
3. Calculate the exponential weighted moving average using the following formula:

$$\text{Exponential WMA (EMA)} = \text{SMA March} * (1 - \alpha \text{ Alpha}) + \text{April} * \alpha \text{ Alpha}$$

Applying the calculation to the example in Table 14-3, you can see the data is further smoothed (shown in Figure 14-3) and the overall trend is more evident. Note that if your data is cyclical by nature, this calculation would obscure those trends.

**Table 14-3:** Exponential Weighted Moving Average

DATE	SALES	SIMPLE MOVING AVERAGE	WEIGHTED MOVING AVERAGE	EXPONENTIAL WEIGHTED MOVING AVERAGE
Jan 2012	300			
Feb 2012	175			
Mar 2012	220	232	218	232
Apr 2012	225	206	215	231
May 2012	230	225	227	231
Jun 2012	210	222	219	228
Jul 2012	205	215	210	224
Aug 2012	200	205	203	221
Sep 2012	215	207	208	220
Oct 2012	230	215	220	221
Nov 2012	275	240	250	229
Dec 2012	290	265	275	238



**Figure 14-3**

## Forecasting with Moving Averages

In the previous sections, you looked at using simple, weighted, and exponential weighted moving averages to smooth out erratic data to better identify trends that exist within. Beyond doing trend analysis, the same formulas can also be applied to forecasting.

To apply these formulas in a forecasting exercise, you simply shift the window. Using the example you have followed throughout this chapter results in a forecast that is based on the moving average calculated from the previous months. It's no coincidence, then, that the moving averages calculated for December 2012 would be used as your forecast for January 2013.

## Forecasting

---

It's a pretty safe bet that the ability to predict the future would be high on just about everybody's list of desirable business traits, if it were possible. After all, the ability to predict sales or even stock movements would make you and your company a lot of money.

Since this is not consistently possible, you are left relying on meaningful forecasts of future events, or the use of statistical analysis to base future events on prior performance.

As noted at the beginning of this chapter, Reporting Services does not have explicit, built-in support for forecasting but can generate trendlines that can be added to charts and graphs. The next few sections are devoted to introducing alternative forecasting methods in both Microsoft Excel and SQL Server Analysis Services (SSAS). Both tools are capable of providing solid forecasts that can ultimately make their way into your dashboard.

## Forecasting with Microsoft Excel

Forecasting with Excel is a common activity in the business community. Every day your coworkers in Finance, Marketing, and potentially IT use the built-in formulas to forecast things like market response rates, budgets, and more. These tasks can be accomplished in Excel and fall into either linear or exponential forecast categories.

### *Forecasting Linear Growth*

Linear or straight-line growth is based on the notion that, at least in the short or near-term without outside interference, momentum will carry a trend forward. To build this forecast, you fit or draw a straight line through prior data points to predict where future data points will fall using the mathematical concept known as linear regression or least squares.

Excel has three formulas available out of the box to help accomplish this without requiring you to understand the underlying mathematics:

- **FORECAST()**: Calculates projected values using a best-fit, least-squares linear regression.
- **TREND()**: Calculates values along a linear trend using the least-squares method.
- **LINEST()**: Uses the least-squares method to calculate statistics for a straight line through your data.

To illustrate the use of these functions, take the data from the previous section and use the **TREND()** formula to forecast unit sales for the January to March 2013 period. The basic format of the **TREND()** function follows:

```
TREND(known_y's, [known_x's], [new_x's], [const])
```

Plugging in the sample data, the formula in Excel can be represented using pseudo-code as:

```
TREND([Sales Jan - Dec 2012], [Jan - Dec 2012], [Jan - Mar 2013])
```

**NOTE** The Excel Trend function returns multiple values or an array as the function result. Entering the formula correctly requires the selection of the cell range that will contain the results, entering the formula in the formula bar, and then using the **CTRL+Shift+Enter** keystroke sequence to apply the formula. When the formula is correctly applied it will display in the formula bar with curly-brackets “{}”.

The results are forecasted points that are plotted along a trendline as shown in Figure 14-4.

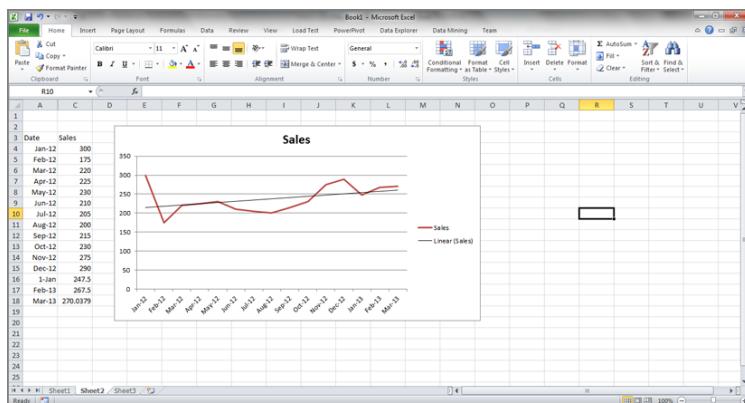


Figure 14-4

**NOTE** Figure 14-4 can be created by first selecting the Month and Sales data range and then using Line Chart function on the Insert toolbar. A Linear Forecast trendline is also displayed and can be inserted by navigating to the Layout toolbar under Chart Tools.

### *Forecasting Exponential Growth*

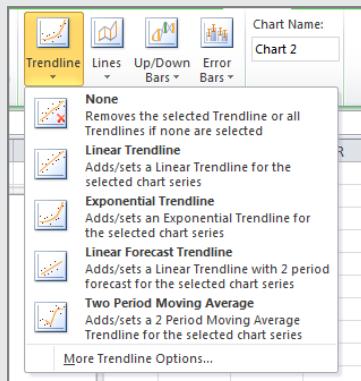
Linear forecasting can be problematic in many scenarios since most growth (or decline) does not follow a straight line. To solve this challenge you use exponential growth to forecast data points that follow a more natural pattern.

Excel has two functions to help you:

- **GROWTH()**: Calculates predicted values along an exponential growth curve.
- **LOGEST()**: Calculates an exponential curve to fit the supplied data.

### **ADDING TRENDLINES TO CHARTS**

In Microsoft Excel 2010, you can easily add trendlines to graphs and charts using built-in tools. To add a trendline to your chart, use the Trendline menu (shown in Figure 14-5) on the Chart Layout ribbon.



**Figure 14-5**

There are four trendline options available out of the box:

- **Linear Trendline**
- **Exponential Trendline**
- **Linear Forecast Trendline**
- **Two-Period Moving Average**

## Forecasting with SQL Server Analysis Services

In the proper hands, Excel is a power forecasting tool. Analysis Services, on the other hand, is a whole different beast: it supports not only a built-in Time Series data mining algorithm capable of accurately forecasting trends and values (as seen previously), but can also perform much more advanced tasks, such as handling seasonality (or periodicity), making interdependent or correlated predictions (finding relationships between items over time), and performing what-if analysis.

You read that right; the SSAS time series function is actually implemented in the form of data mining. Although the advanced features of the Microsoft Time Series algorithm is beyond the scope of this book and this chapter, this section introduces you to building and using a simple forecasting tool using Analysis Services.

**NOTE** In addition to working directly in Analysis Services, you can use the Data Mining Add-In for Excel to work with the Microsoft Time Series algorithm (and all the other built-in algorithms as well).

To do so, download the plug-in from: <http://office.microsoft.com/en-us/excel-help/data-mining-add-ins-HA010342915.aspx> and complete the following steps:

1. After the plug-in is installed, connect to your Analysis Services instance by clicking the Connection button and then enter the server/host address and select a catalog or Analysis Services database.
2. With your connection configured, select the cells that contain the time series data that will be used for the forecast and click the Forecast button.
3. The Forecast wizard enables you to configure your selected data range; select which column of data represents the date time range and name the mining structure that will be created within the cube on the Analysis Services instance.
4. Click Finish to deploy and process the mining structure and view the results as seen in Figure 14-6.

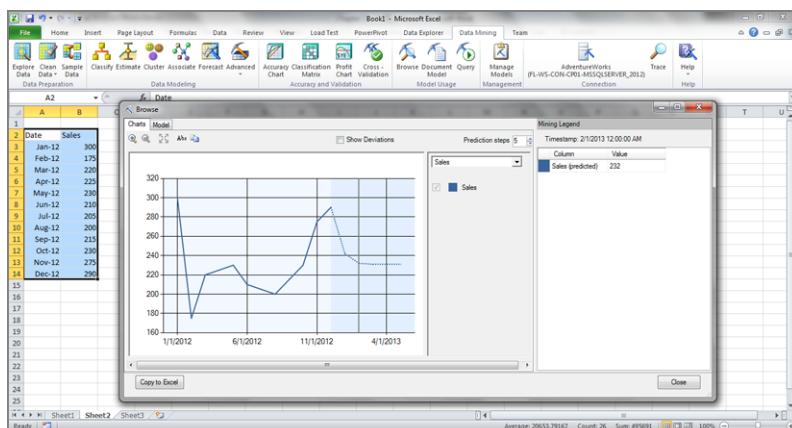


Figure 14-6

## Getting Started with Data Mining in Analysis Services

If you have not previously worked with Analysis Services, this section quickly introduces you to the various parts and pieces that are needed to work specifically with the Microsoft Time Series data mining algorithm. It is not an exhaustive discussion on data mining, but it should help you get started using time series forecasting.

**NOTE** Before starting this section you will need to have an instance of Analysis Services installed and set up for Multidimensional mode.

### Analysis Services (Multidimensional) Project

The basis of this section is an SSAS project. Using SQL Server Data Tools (SSDT, formerly known as BIDS), you will find an Analysis Services Project template installed with the Business Intelligence templates (see Figure 14-7). To create and work with an SSAS project, you will need to create a new project using this template.

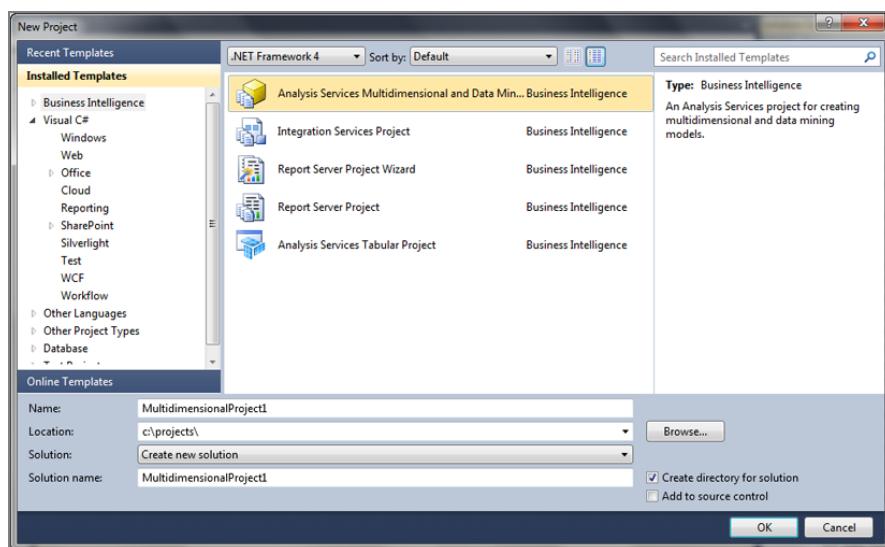


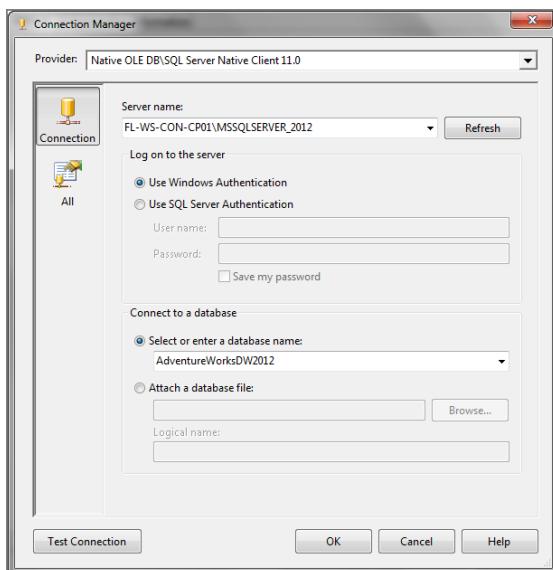
Figure 14-7

**NOTE** In SQL Server 2012, Microsoft introduced a new mode or version of Analysis Services called Tabular. When starting a new project make sure you select the Multidimensional mode, as SSAS Tabular does not support data mining.

### Data Source

The *data source* is the most basic object within SSAS. Its sole purpose is to act as a connector to the data. The data source is your connection to a wide variety of sources beyond SQL Server. Sources of data can include other RDMS systems as well as non-traditional sources, such as Microsoft Access.

When you set up a new data source, a wizard (shown in Figure 14-8) is provided for you. The wizard consists of two primary steps. The first is the identification of the data source, and the second involves the specification of the id or proxy used to securely connect to your source.



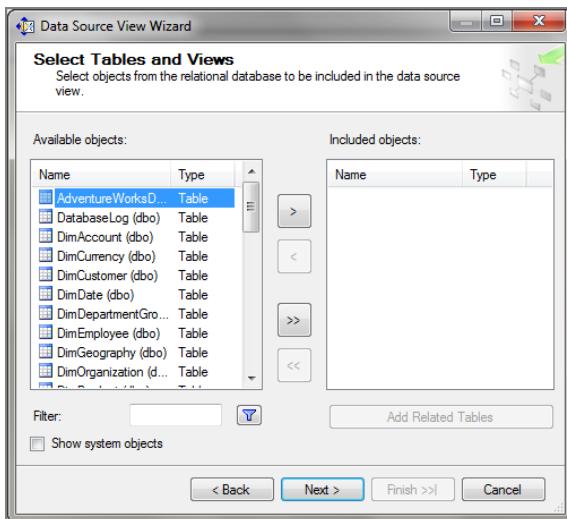
**Figure 14-8**

The data source connection wizard should look and feel familiar. It is identical to what you have worked with throughout this book as you have built out your dashboard and balanced scorecards in Reporting Services.

### Data Source View

The *data source view* is a layer that sits on top of your data source. It can be used in a rather sophisticated way, functioning as an abstraction layer that allows you to shape your source of data without actually modifying the underlying data source. For the purpose of data mining, however, it is only necessary to reference the tables and views that are required for you to complete the task at hand.

When creating a new data source view, you are presented with a wizard (shown in Figure 14-9) similar to that used for creating a new data source. The wizard requires that you select the data source described previously and then select the tables and/or views that you want to include and make available in the project.



**Figure 14-9**

### DATA MINING DATA PREPARATION

One of the most important parts of data mining, which is not addressed specifically within this chapter, is the time and effort required to prepare your data for mining.

Data mining is not only dependent on clean, quality data, but missing values or measurement can be especially problematic if not handled properly. Analysis Services provides a `MISSING_VALUE_SUBSTITUTION` parameter on the Time Series algorithm that allows you to use a numeric constant, the previous value, or a mean value as a replacement. The default value for this parameter is set to `none`.

### Mining Structure

The *data mining structure* is a bucket or container that defines the shape of the forecast you are creating. Its functions include a number of pieces and, like the previous steps, it can be built using a wizard. The most relevant structural pieces to be familiar with are:

- The bindings to your data source, including information such as each column's data type, whether the column should be used as a key column during forecasting, and the column's content type (discrete or continuous).
- Your selected mining model, discussed next.
- Compressed and cached data that can be made available to the model.

### Mining Model

The *mining model* is where the rubber meets the road for your time series forecast, since it is the actual object that is responsible for taking your data, applying the machine learning algorithm (Microsoft Time Series Analysis) and returning the forecasted results.

To this point, there has been no real discussion on the principles of this algorithm or what this built-in algorithm actually does. Without getting into the mathematical explanations, this supplied algorithm is capable of making predictions using one of three modes or techniques:

- **Auto-regression Trees with Cross Prediction (ARTxp):** This function is a regression tree algorithm such as those implemented for decision trees and excels at short-term forecasting. It breaks down when used for long-term prediction.
- **Auto-regression Integrated Moving Averages (ARIMA):** This is the more general algorithm for predicting values in time series. It is not as accurate as the ARTxp in the near term but is more stable in longer range forecast. Microsoft's implementation of ARIMA actually finds the optimal set of algorithm parameters for your data and handles seasonality or periodicity.
- **Blend between ARTxp and ARIMA:** Blending both models together is the default mode for the time series algorithm and results in a weighted average of the two algorithms, where more weight is given to ARTxp in the near term and to ARIMA in the long term.

Now that you have a functional understanding of both mining structures and models, add both to your project by right-clicking the Mining Structure folder in the Solution Explorer and selecting New Mining Structure... Yet another wizard will walk you through the set-up:

1. Select your source of data. You can typically use an existing relational database or data warehouse as your source of data. Before you get started with the wizard, you will need to configure a data source connected to the Adventure Works Data Warehouse and then set up the data source view to include all the tables within the project.
2. Select Microsoft Time Series as the data mining technique (shown in Figure 14-10).
3. Select the data source view you previously created in Step 1.
4. Find your source data table or view and define it as the Case (shown in Figure 14-11).

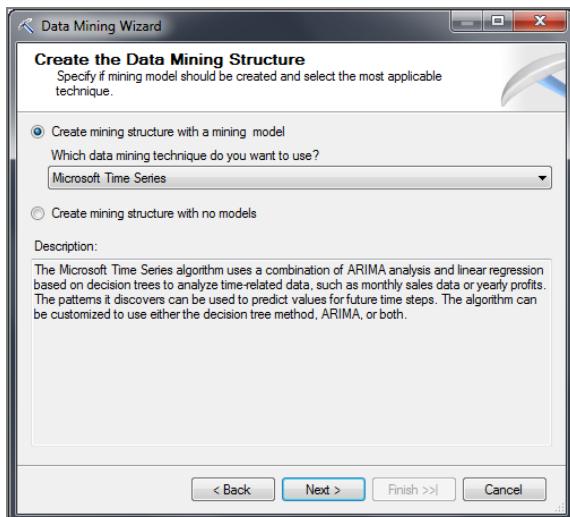


Figure 14-10

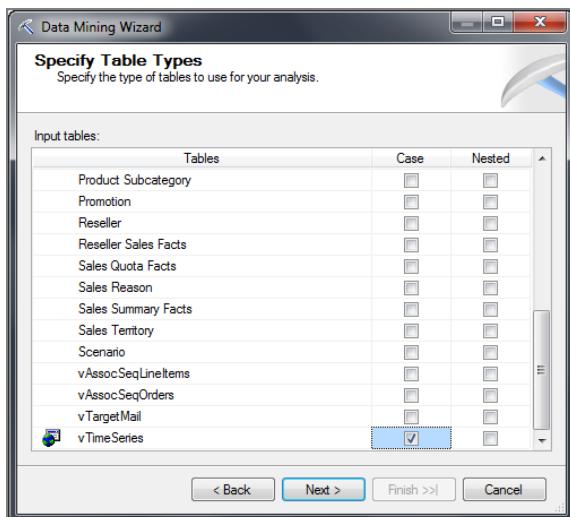
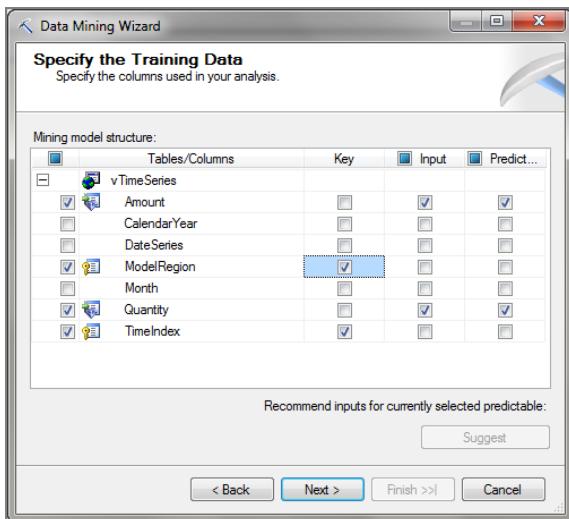
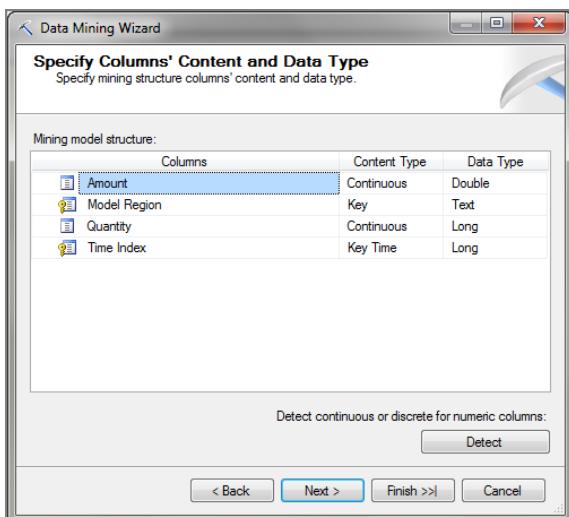


Figure 14-11

5. Now comes the most difficult part: Identify each data column's role within the mining structure either as a key, an input, a prediction, or both input and prediction (as seen in Figure 14-12). SSAS can help you identify which columns are correlated using the Suggest button, but at this point you should already be familiar with the data you are using.

**Figure 14-12**

- The wizard will detect the content types (as shown in Figure 14-13) for each attribute or column specified. Note that the time series algorithm only functions on continuous data and cannot be used if the content type is discrete.

**Figure 14-13**

7. Name both your mining structure and mining model so you can easily use the Data Mining Expression (DMX) language later.

You may notice that explicitly missing from the preceding steps is an Analysis Service multidimensional cube. The data mining structures and models are in fact independent of the cube and can function without the time or complexity needed to implement one. This is how the Excel Data Mining add-in is implemented under the covers.

### Processing and Predicting

With your mining structure and mining model complete, it is necessary to deploy and then process the model before you can use it to make your forecast predictions.

Before you move forward, open the Project Properties (shown in Figure 14-14) and set the deployment server target to the Analysis Services instance. Now, you can right-click your mining structure to open the Context menu and then click the Process... option.

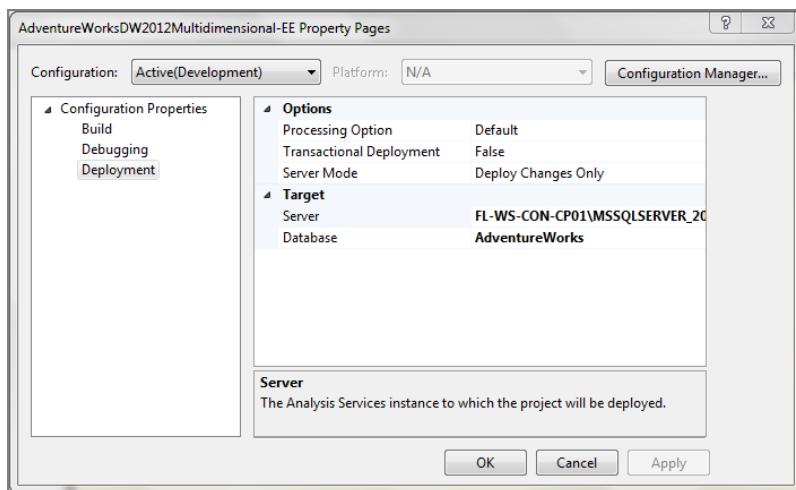


Figure 14-14

You will see several prompts indicating that the cube is out of date and needs to be redeployed before the Process dialog opens (shown in Figure 14-15). Click Run to process your mining structure and then close all dialog windows once it completes.

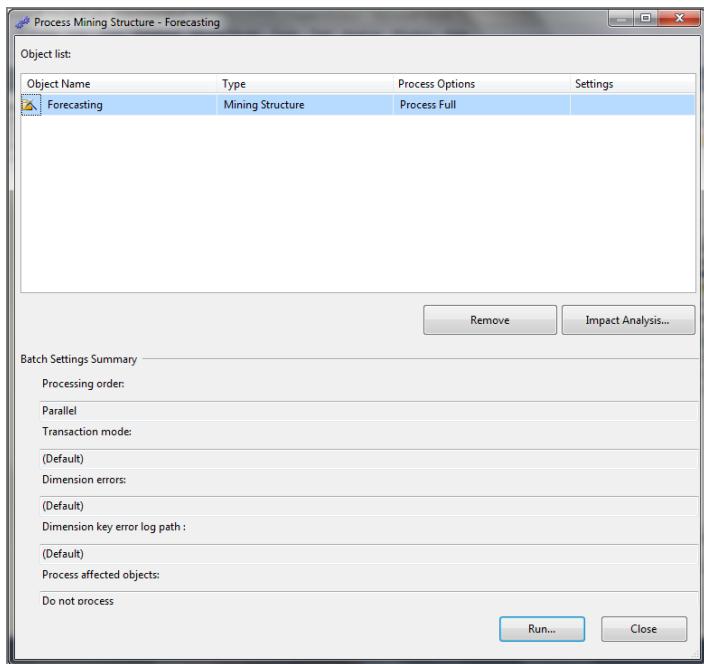


Figure 14-15

Next, open the Forecasting mining structure from the Solution Explorer and use the Mining Model Viewer (shown in Figure 14-16) built into the SQL Server Data Tools (formerly known as BIDS) development environment to preview your forecast, including charting up to 100 periods out with deviations calculated at each point.

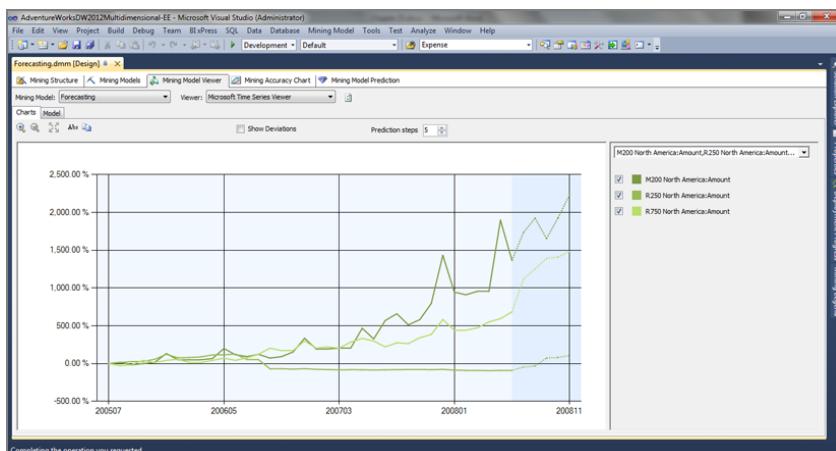


Figure 14-16

Although the preview is helpful to determine if your forecast mining structure is configured correctly to produce meaningful results, it is not that helpful in producing forecast results that can be consumed elsewhere, such as in your dashboard or other analytical report.

To overcome this, you must use the Data Mining Expression (DMX) language to query the mining model. Since a full explanation of DMX could consume an entire book by itself, a sample (shown in Figure 14-17) that predicts the next six periods for two products in North America is presented only for illustrative purposes using SQL Server Management Studio.

```
SELECT FLATTENED
[Forecasting].[Model Region],
PredictTimeSeries([Forecasting].[Quantity],6) AS PredictQty,
PredictTimeSeries ([Forecasting].[Amount],6) AS PredictAmt
FROM [Forecasting]
WHERE [Model Region] = 'M200 North America' OR
[Model Region] = 'R250 North America'
```

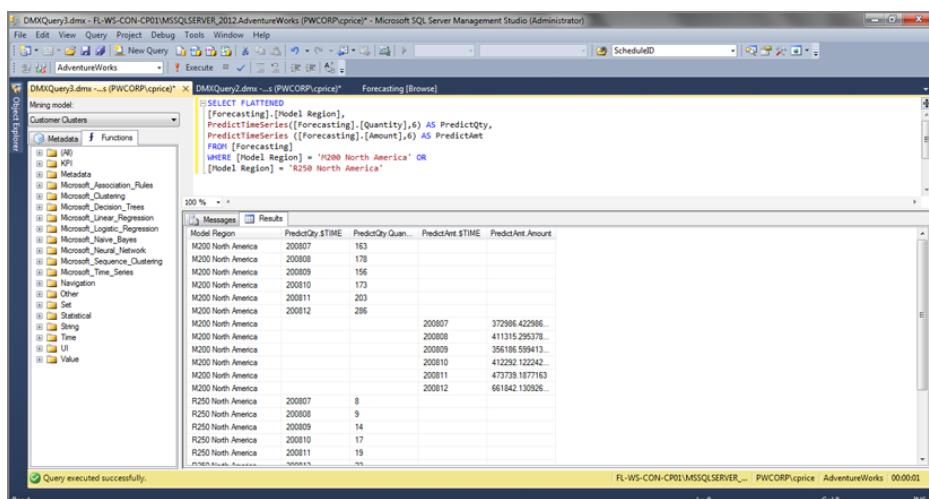


Figure 14-17

### Using Trendlines in Reporting Services

Although trendlines are not forecasting in the truest sense, they can be useful to smooth data to identify a trend visually. Reporting Services supports this feature in the form of a calculated series.

Calculated series can be added to any series within a chart and they support a wide range of formulas (see Figure 14-18), including the simple moving

average, weighted moving average, and exponential weighted moving average that you learned about previously.

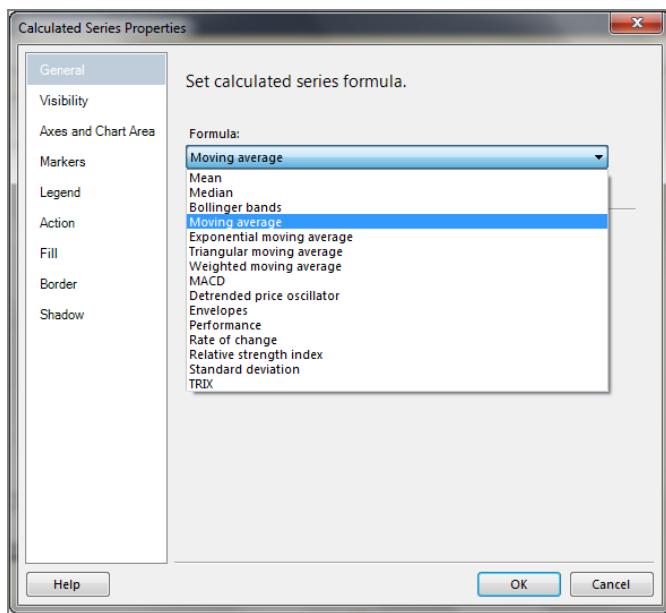


Figure 14-18

To add a calculated series to a Reporting Services chart, follow these steps:

1. Select the chart in the design view to show the Chart Data window.
2. Right-click the series that the new calculated series will be based on to bring up the Context menu. Then select the Add calculated series... option.
3. In the Calculated Series Properties window, you can select the formula you wish to use to define a trendline. For moving average formulas there is a period parameter that you can use to define the size of the sliding window you want to use. Note that other formula options may have additional parameters, such as a standard deviation for the Bollinger Bands formula.
4. Click OK to create the calculated series, as shown in Figure 14-19.

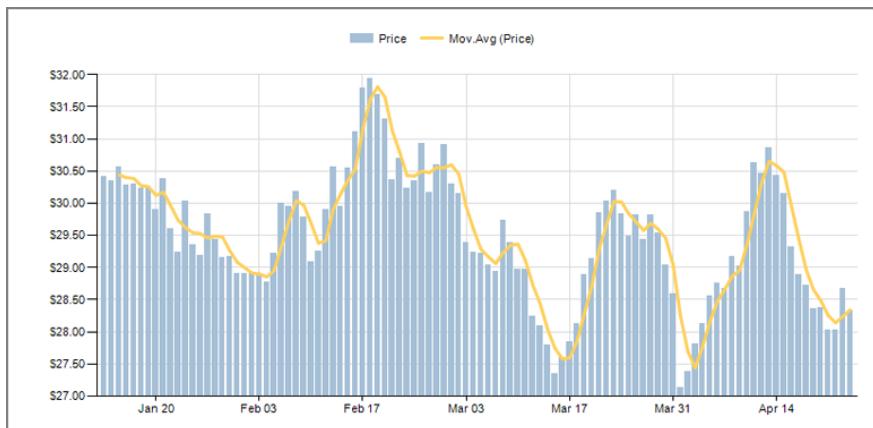


Figure 14-19

## Case Study: Healthcare Forecasting

Forecasting healthcare demand is no different from what is done day in and day out for IT budgets, sales projections, marketing response rates, and even call center volumes. Instead of dealing with dollars and cents or widgets, the services or incidents of disease (or accidents) is predicted. This type of forecasting is useful in a number of different scenarios for healthcare organizations ranging from budgeting and resource planning to marketing.

In this case study, the Contoso Health System is interested in expanding and building a new primary care facility to service its expanding market. As part of the planning activities, a forecast of projected primary care needs has been requested for each ZIP code within its market. After the data is gathered, Analysis Services is used to build a Time Series data mining model for use in the predictions.

### Gathering and Preparing the Data

Gathering data from health information systems can be difficult, particularly when information that is protected by HIPAA is involved. Luckily, the data required for this activity consists of little more than a count-by procedure (identified by an industry ICD-9 procedure code) at the ZIP code level.

After doing some initial analysis of the data, Contoso analysts have determined the following:

- The best level of granularity at which to aggregate the data is month of year. Using month-of-year as the grain allows for cyclical trends to be identified and minimizes the need to handle missing data points.
- Although there are thousands of procedures possible, 25 have been identified as accounting for nearly 80 percent of all primary care visits. The analysis and forecasting will be limited to these 25 ICD-9 codes.

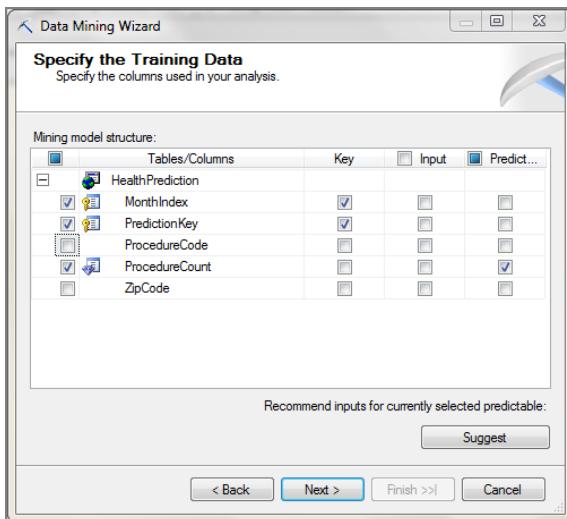
To prepare for forecasting, the data has been exported to a table on a SQL Server instance using the data schema provided in Table 14-4.

**Table 14-4:** Forecast Data Schema

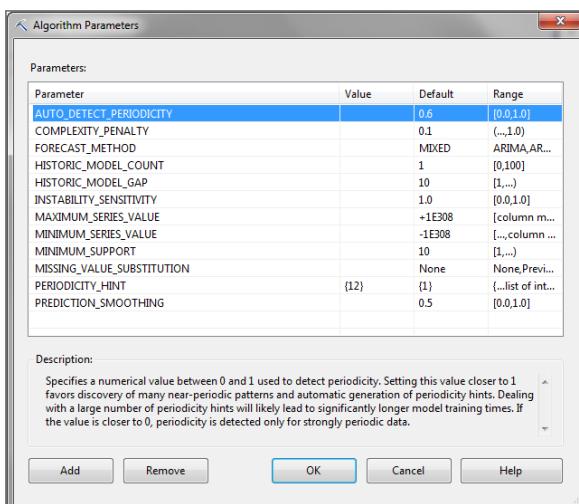
COLUMN NAME	DATA TYPE	DESCRIPTION
MonthIndex	Int	The measured month encoded in YYYYMM format
ProcedureCode	Varchar(5)	The ICD-9 procedure code representing the measured procedure
ZipCode	Varchar(5)	The 5-digit U.S. ZIP/postal code
PredictionKey	Varchar(11)	A concatenation of the ProcedureCode and ZipCode separated by a dash (-)
ProcedureCount	Int	The observed aggregated count for the procedure in a ZIP code during the defined period of time

## Building the Time Series Model

Building the mining structure and Microsoft Time Series mining model is a straightforward endeavor. Using the table created to hold the prepared data as the source, Contoso is only concerned with three columns: the `MonthIndex` and `PredictionKey` function as the structure keys and will be used to predict the `ProcedureCount` (as shown in Figure 14-20).

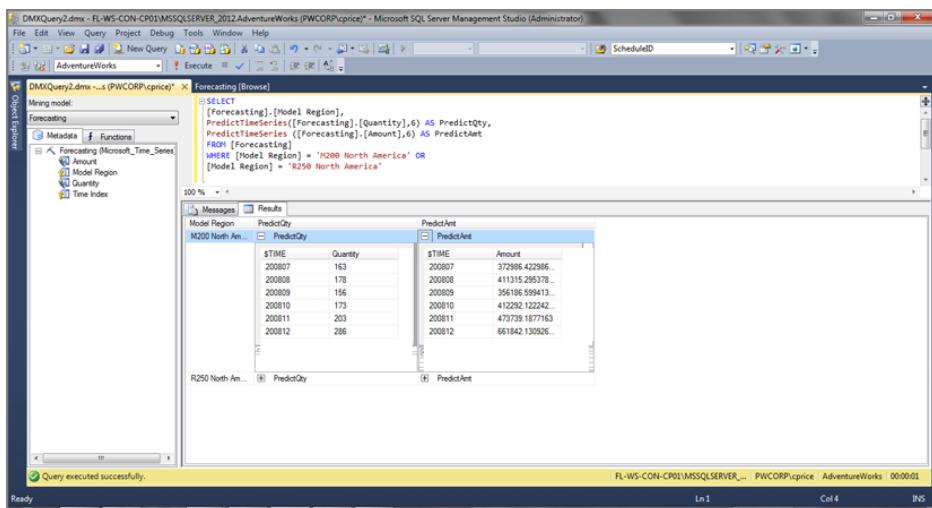
**Figure 14-20**

Because Contoso is interested in observing seasonal or cyclical trends, it's necessary to provide the algorithm with a hint as to the periodicity of the data. This hint is found and configured in the algorithm parameters, as shown in Figure 14-21, by setting the `PERIODICITY_HINT` algorithm parameter to a value of `{12}` to represent the 12 months of a year.

**Figure 14-21**

## Making the Forecast

After the structure and model are processed and evaluated, the Contoso Health System analyst uses a DMX query to make a 24-month forecast for each case (zipCode/ProcedureCode). The flattened forecasts results are exported to a working table that can be rejoined with the training data to present a concise set of results (shown in Figure 14-22) to the project team.



**Figure 14-22**

## Summary

In this chapter you looked at techniques for identifying trends by smoothing erratic data and techniques for making forecasts. Multiple tools are available for this, from the built-in functions available in Microsoft Excel, to the Time Series data mining model in Microsoft SQL Server Analysis Services. Applying these techniques is useful when it's necessary to highlight trends within data or deliver forecasts on your dashboards or other analytical reports.

# INDEX

## NUMBERS

3-D charts vs. line charts, 90–91  
80-20 Rule, 240

## A

Adobe Kuler, 102  
Adventure Works. *See* dashboards, building (Adventure Works)  
Agile Development Methodology, 29–30  
alerts  
    Data Alerts, 170–171  
    when publishing, 167–168  
Analysis Services. *See* forecasting with SQL Server Analysis Services  
Analytic Grids and Charts (PerformancePoint), 132  
area charts, 92  
ARIMA (Auto-regression Integrated Moving Averages), 266  
ARTxp (Auto-regression Trees with Cross Prediction), 266  
assemblies, 146–153  
    deploying, 150–151  
    in reports, 151–153  
.NET, building, 147–150  
    pros and cons of, 146–147  
audiences  
    data visualization and, 90  
    designing for, 10–11  
planning mobile BI and, 125–126

Auto-regression Integrated Moving Averages (ARIMA), 266  
Auto-regression Trees with Cross Prediction (ARTxp), 266  
averages. *See* moving averages  
AWExecutiveDashboard.  
    *See* dashboards, building (Adventure Works)  
axes in charts and graphs  
    formatting, 97–99  
horizontal and vertical, 95  
secondary, configuring, 105–106

## B

balanced scorecards  
    basics of, 4–5  
    benefits of, 7–9  
    building. *See* dashboards and balanced scorecards, building  
    defined, 195  
enhancing. *See* dashboards and balanced scorecards, enhancing  
extending with custom code. *See* custom code, extending dashboards and balanced scorecards with  
interactive. *See* interactive dashboards and balanced scorecards  
strategy maps and, 196  
visual components. *See* visual dashboards and balanced scorecards

- Balanced Scorecards & Operational Dashboards with Microsoft Excel*, 201
- Balanced Scorecards case study, 219–231
- balanced scorecards, building, 226–231
  - data warehouses, 226
  - defining objectives, 220–221
  - four business perspectives, 219
  - indicators, defining, 225–226
  - metrics, selecting, 222–223
  - targets, selecting appropriate, 223–225
- balanced scorecards, introduction to, 3–19
- audiences, designing for, 10–11
  - basics of, 4–5
  - benefits of, 7–9
  - evaluation of, 16–17
  - example of, 5
  - organization of. *See* balanced scorecards, organization of
  - strategic management systems, origins of, 3–4
  - tools, selecting, 9–10
  - tools of modern management, 4–7
- balanced scorecards, organization of, 11–19
- strategic function, defining. *See* strategic function, defining
  - strategy maps, importance of, 17–18
  - tools, reviewing, 19
- bar charts, 92, 240
- BarStyleLabel property, 250
- Beautiful Evidence*, 107
- Beginning Visual Basic 2012* (Wrox, 2012), 140
- BI (business intelligence)
- Analysis Services Project template, 263
  - mobile BI, audiences and, 125–126
  - mobile BI markets, 125
- BIDS. *See* SQL Server Data Tools (SSDT)
- Bollinger bands, 102, 272
- borders, formatting, 57
- box plots, 233, 235
- box-and-whisker charts
- building, 235–239
  - five statistics, 234–235
- Brainstorming and Intensive Discussion Approach, 201
- browser-based delivery solutions, 126–127
- bullet graphs, 110, 112–115
- business intelligence (BI)
- Analysis Services Project template, 263
  - mobile BI, audiences and, 125–126
  - mobile BI markets, 125
- Business Intelligence Development Studio or BIDS. *See* SQL Server Data Tools (SSDT)
- C**
- C#, embedded code and, 141, 146
- calculated series (charts), 102–103
- callbacks, 128
- candlestick box-and-whisker charts, 233, 235
- CanGrow/CanShrink properties, 66
- cascading parameters, 82–83
- cascading style sheets (CSS), Reporting Services and, 128
- case studies
- Adventure Works. *See* dashboards, building (Adventure Works)
  - Balanced Scorecards. *See* Balanced Scorecards case study
  - Contoso plastics. *See* Contoso Plastics case study
  - health care forecasting case study, 273–276
  - casual links, strategy maps, 200–201
  - category groups (charts), 94
  - cells, formatting, 144–145
  - charts and graphs. *See also* box-and-whisker charts; decision-making charts
  - axes, formatting, 97–99
  - basics of, 92–94
  - category sales vs. gross profit sparkline, 186–187
  - chart area, 94
  - chart data region, 94
  - chart series, 94
  - chart types, 92–93
  - chart visualization, creating, 86
  - color, controlling, 106–107
  - color palette, modifying, 101–102
  - horizontal and vertical axes, 95
  - implementing, 95–97
  - Internet sales vs. gross profit (Adventure Works), 185–186
  - keywords, 100
  - legends, 95
  - reseller sales vs. prior year graph (Adventure Works), 187–188
  - secondary axes, configuring, 105–106
  - titles, 95
  - tooltips, adding, 99–101
  - trendlines, adding to charts, 261
  - trendlines, defining, 102–105

- code. *See also* custom code, extending dashboards and balanced scorecards with
- color  
 alternating in table rows, 62  
 background, changing, 143–145  
 chart color palette, modifying, 101–102  
 controlling in charts, 106–107  
 in data visualizations, 91  
 formatting and, 57  
 ColorFormat project, 147  
 column charts, 92  
 column headers, adding interactive sorting to, 71–72  
 conditional formatting, 60–62  
 configuring  
 email for services application, 167–168  
 gauges, 111, 112  
 native report servers. *See* native report servers, configuring and publishing to Reporting Services Configuration Manager, 158  
 Reporting Services for SharePoint, 166–169  
 Reporting Services, SSL and, 157  
 secondary axes in charts, 105–106  
 content types, adding to libraries, 167–168  
 Contoso Health System. *See* health care forecasting case study  
 Contoso Plastics case study, 210–213  
 controls  
 Report Services Controls Toolbox, 45–46  
 unusable in headers and footers, 53  
 cost, of using SharePoint, 156  
`Count()` function, 249  
`CrossAt` property, 246  
 CSS (cascading style sheets), Reporting Services and, 128  
 custom code, extending dashboards and balanced scorecards with, 139–153  
 assemblies. *See* assemblies  
 embedding, pros and cons of, 140  
 expressions, 139  
 functions, writing, 142–145  
 inputting, 141–142  
 Visual Basic and, 139–140  
 customer satisfaction  
 metrics and, 28, 219  
 strategic objectives and, 26  
 customizing  
 chart color, 101  
 date and time, 64–66  
 labels and titles, 63–64  
 roles, 164
- D**
- dashboard and scorecard projects, planning and initiating, 21–32  
 defining measures, 25–28  
 development and implementation, 29–30  
 executive involvement, 22–25  
 expectations and scope, managing, 30–31  
 metrics and, 28–29  
 Dashboard Designer example, 129  
 dashboards  
 basics of, 7  
 extending with custom code. *See* custom code, extending dashboards and balanced scorecards with  
 illustration of, 7  
 interactive. *See* interactive dashboards and balanced scorecards  
 mobile. *See* mobile dashboards and balanced scorecards  
 PerformancePoint, 128–129, 134  
 projects, planning and initiating.  
*See* dashboard and scorecard projects, planning and initiating  
 publishing. *See* publishing dashboards and balanced scorecards  
 visual components. *See* visual dashboards and balanced scorecards  
 dashboards, building  
 (Adventure Works), 175–194  
 category sales vs. gross profit sparkline, 186–187  
 data source set-up, 178–179  
 datasets, creating, 179–183  
 design, 177  
 finishing touches, 193–194  
 Internet and reseller sales map, 188–192  
 Internet sales vs. gross profit graph, 185–186  
 KPIs table, 193  
 layout, defining, 184  
 objective measurements, 176  
 project set-up, 177–183  
 reports, creating, 183  
 reseller sales vs. prior year graph, 187–188  
 strategic, building dashboards and, 175–176  
 strategic objectives, 175–176

- dashboards and balanced scorecards,  
building, 35–50  
dashboards, building, 46–50  
Reporting Services and. *See* Reporting Services project  
dashboards and balanced scorecards,  
enhancing, 51–66  
conditional formatting, 60–62  
custom labels, 63–64  
dates and times, 64–66  
fonts and textboxes, formatting, 55–56  
headers and footers, 53–54  
placement and layout, 52–53  
reports, subreports and paging, 54–55  
data  
data strategy, 8  
gathering (health care forecasting case study), 273–276  
gathering when building dashboards, 48  
retrieval, parameters and, 76–77  
SMA and, 255  
Data Alerts, 170–171  
data bars, 107, 109  
data marts/data warehouses, 135  
data mining  
algorithm, 263  
Data Mining Add-In for Excel, 262  
Data Mining Expression (DMX)  
language, 270  
mining model, building, 274–275  
Mining Model Viewer, 270  
model, 266–269  
model processing, 269–271  
preparation for, 265  
structure, 265  
data region (charts), 94  
data sources  
basics of, 36  
creating, 37–38  
dashboards, 130  
defining (SSAS), 178–179  
performance and security, 135  
SSAS, 178–179, 264–265  
data warehouses  
Balanced Scorecards case study, 226  
Data Warehouse database, 178  
defined, 226  
databases  
for Adventure Works project, 178  
database repository for Reporting Services, 158  
examples, downloading, 38  
separating from Report Servers, 159  
data-driven parameters, 79–81  
datasets  
basics of, 36–38  
cascading parameters and, 82  
creating (Adventure Works), 179–183  
data-driven parameters and, 79–81  
date and time  
customizing, 64–66  
stamping, 64  
day of the month, calculating, 65–66  
decision-making charts, 233–251  
box-and-whisker charts. *See* box-and-whisker charts  
Gantt charts, 248–250  
Pareto Charts, 240–244  
variance charts, 244–248  
decomposition tree diagram, 133  
delivery solutions, planning  
mobile BI and, 126–127  
deploying  
assemblies, 150–151  
when publishing reports, 155–156, 170  
design  
Contoso Plastics, 212–213  
dashboards, building  
(Adventure Works), 177  
designing dashboards, 49–50  
mobile dashboards and, 137  
Reporting Services project, 41–44  
strategy maps, 201  
design tool, 227  
destination statements (balanced scorecards), 4  
development and implementation, project planning and, 29–30  
DMAIC (Six Sigma), 211  
DMX (Data Mining Expression language), 270  
downloading. *See* websites for downloading  
drillthrough reports  
basics of, 68–69  
step-by-step creation of, 83–88  
dropdown lists, data-driven parameters and, 79–81

**E**

EMA (exponential weighted moving average), 257–258  
 email  
     configuring, services applications and, 168–169  
     E-mail Settings page, configuring, 158  
 embedding  
     custom code, pros and cons of, 140  
     metrics, 49  
 Encryption Keys page, Reporting Services Configuration Manager, 158  
 enhancing dashboards. *See* dashboards  
     and balanced scorecards, enhancing  
 environmental factors, PESTEL  
     framework and, 198  
 ergonomics, mobile design and, 137  
 ETL (Extract-Transform-Load) process,  
     defined, 48  
 Excel  
     *Balanced Scorecards & Operational Dashboards with Microsoft Excel*, 132  
     as basic tool, 9–10  
     building process and strategy  
         maps and, 217–218  
     business diagrams and, 217  
     Data Mining Add-In, 262  
     data strategy and, 8  
     forecasting trends with, 259–263  
     PerformancePoint reports and, 132  
 executives  
     designing for, 10  
     explaining metrics to, 24  
     selecting during project planning, 22–25  
 Expand and Collapse groups, 69–71  
 expectations, project planning and, 30–31  
 exponential growth, forecasting, 261  
 exponential weighted moving average (EMA), 257–258  
 expressions  
     conditional formatting and, 60–62  
     custom code and, 139  
 Extract-Transform-Load (ETL) process,  
     defined, 48

**F**

Few, Stephen, 90, 112, 113  
 filtering  
     drillthrough reports and, 69  
     measures, 27

filters, 130, 131

fonts  
     formatting, 55–56  
     size, changing, 215  
 footers, 53–54  
 FORECAST() function, 260  
 forecasting trends, 253–276  
     basics of, 259  
     exponential growth, 261  
     health care forecasting case  
         study, 273–276  
     linear growth, 259–261  
     with Microsoft Excel, 259–263  
     smoothing erratic data. *See* smoothing erratic data  
     with SQL Server Analysis Services  
         *See* forecasting with SQL Server Analysis Services; SSAS project  
     trendlines, 271–273  
 forecasting with SQL Server Analysis Services, 262–273  
     basics of, 262  
     data mining algorithm introduced, 263  
     SSAS project. *See* SSAS project  
 formatting  
     cells, 144–145  
     cleaning up (charts), 186  
     fonts and textboxes, 55–56  
     FormatCurrency function, 63  
     FormatDateTime() function, 64  
     maps, 191–192  
     Report Formatting toolbar, 56  
     in Reporting Services, 43  
 forms-based authentication, mobile devices and, 136  
 functions, writing, 142–145

**G**

Gantt, Henry, 248  
 Gantt charts  
     basics of, 248–250  
     project planning and, 30  
 gauges, 110–112  
 graphs. *See also* charts and graphs  
     bullet graphs, 112–115  
     Internet sales vs. gross profit (Adventure Works), 185–186  
     reseller sales vs. prior year (Adventure Works), 187–188  
 groups, opting for, security and, 163

growth  
 exponential, 261  
`GROWTH()` function, 261  
 linear, 259–261  
 planning for, Reporting Services  
 and, 158–161

**H**

Harvard Business Journal, 3  
 headers  
 basics of, 53–54  
 column headers, adding interactive  
 sorting to, 71–72  
 health care forecasting case study, 273–276  
 horizontal axis in charts and graphs,  
 95, 97–98

**I**

ICD-9 procedure codes, 273, 274  
 identifying  
 metrics, 48  
 strategic initiatives, 15–16  
**IF-THEN-ELSE** statement  
 (Visual Basic), 60–61  
 indicators  
 basics of, 115–118  
 defining (Balanced Scorecards  
 case study), 225–226  
 PerformancePoint, 130, 131  
*Information Dashboard Design*, 90  
 inputting code, 141–142  
 interactive dashboards and balanced  
 scorecards, 67–88  
 basics of, 67  
 drillthrough reports, 68–69  
 Expand and Collapse groups, 69–71  
 interactive sorting, 71–72  
 parameterization. *See* parameters  
 step-by-step drillthrough report, 83–88  
 interactive sorting, 71–72  
 International Society of Six Sigma  
 Professionals, 202  
 Internet  
 Internet and reseller sales map, 188–192  
 Internet sales vs. gross profit (Adventure  
 Works), 185–186  
 interquartile range, charts and, 234  
 interval of measurements, 176  
 item-level roles, 162, 164

**K**

Kaplan, Robert  
 four perspectives and, 198  
 strategic management systems  
 and, 1, 3–4  
 strategy maps and, 196  
 keywords (charts), 99–100  
 Kimball, Ralph, 135  
 KPIs (key performance indicators)  
 dashboards, building and, 48–49  
 displaying in tables  
 (Adventure Works), 193  
 Kuler by Adobe, 102

**L**

labels, custom, 63–64  
 lags  
 defined, 255  
 EMA and, 257  
 WMA and, 256, 257  
 Layer Wizard, 190  
 layers, maps and, 118–123  
 layout  
 Adventure Works dashboard, 176–177  
 in balanced scorecards, 52–53  
 defining (Adventure Works), 184  
 Lean Manufacturing, 202, 207  
 left-to-right, in placement  
 and layout, 53–54  
 legends, charts and graphs and, 95  
 libraries, adding content types to  
 (Reporting Services), 167–168  
 Line Chart function, 261  
 line charts  
 data visualization and, 90–92, 104, 177  
 Pareto charts and, 240  
 variance charts and, 247  
 line layer (maps), 118  
 linear gauges, 110  
 linear growth, forecasting, 259–261  
 lines, formatting, 56–57  
`LINEST()` function, 260  
`LOGEST();`, 261  
 lower quartile value, charts and, 234

**M**

MACD (Moving Average Convergence  
 Divergence) in charts, 103  
*Manifesto for Agile Software Development*, 29

- maps, 118–123  
     Internet and reseller sales maps, 188–192  
     Map wizard, 188–190  
     perspective maps (balanced scorecards), 4  
     point and zoom centering, 192  
     strategy maps, importance of, 17–18  
 marketing  
     designing for, 10–11  
     mobile BI markets, 125  
 matrix cells, sparklines and  
     data bars and, 109  
 max value, charts and, 234  
 mean (charts), defined, 103  
 measurements, objective  
     (Adventure Works), 176  
 measures  
     defining, 25–28  
     filtering, 27  
     performance measures, defining, 15  
     placement of metrics and measures, 52–53  
     validating, 27–28  
 median (charts), defined, 103  
 median value, charts and, 234  
 menus, Report, 142  
 metadata, to include in dashboards, 53, 54  
 metrics  
     customer satisfaction and, 28, 219  
     embedding, 49  
     explaining to executives, 24  
     implementing when building  
         dashboards, 48–49  
     placement of, 52–53  
     project planning and, 28–29  
     selecting (Balanced Scorecard  
         case study), 222–223  
     when building dashboards, 47–48  
 Microsoft Analysis Services cube, 135  
 Microsoft Excel  
     *Balanced Scorecards & Operational  
         Dashboards with Microsoft Excel*, 132  
     as basic tool, 9–10  
     building process and strategy maps and,  
         217–218, 275–276  
     business diagrams and, 217  
     Data Mining Add-In, 262  
     data strategy and, 8  
     forecasting trends with, 259–263  
     PerformancePoint reports and, 132  
 Microsoft Office, business diagrams  
     and, 217–218  
 Microsoft PerformancePoint  
     building process and strategy  
         maps and, 218  
     as a mobile platform, 128–134  
 Microsoft SQL Server 2012 Reporting  
     Services. *See* Reporting Services  
 Microsoft Time Series mining model,  
     building, 274–275  
 Microsoft Visio  
     strategy and process maps,  
         implementing with, 214–217  
 Visio Services, building process and  
     strategy maps and, 218  
 Microsoft Word, building process and  
     strategy maps and, 217–218  
 min value, charts and, 234  
 mining. *See* data mining  
 MISSING\_VALUE\_SUBSTITUTION  
     parameter, 265  
 mobile dashboards and balanced  
     scorecards, 125–138  
     audiences, planning for, 125–126  
     browser-based or native delivery  
         solutions, 126–127  
     design, 137  
 mobile BI market, 125  
 mobile platform selection. *See* mobile  
     platform selection  
     performance, 134–136  
     security, 134, 136  
 mobile platform selection, 127–134  
     PerformancePoint, 128–134  
     Reporting Services, 127–128  
 MonthIndex function, 274  
 Moving Average Convergence Divergence  
     (MACD) in charts, 103  
 moving averages  
     exponential weighted moving  
         average, 257–258  
     forecasting with, 259  
     simple moving average, 254–255  
     types of (charts), 102  
     weighted moving average, 256–257  
 MSDN Hands-On lab, 129  
 Multidimensional mode  
     (Analysis Services), 263  
 multi-tier scorecards, defining, 16

**N**

names, strong, 149–150  
 native delivery solutions, 126–127  
 native report servers, configuring and publishing to, 156–165  
 planning for growth and scalability, 158–161  
 security, 162–165  
 setup and configuration of Reporting Services, 156–158  
 SQL Server Reporting Services, deploying, 161–162  
 .NET assemblies, building, 147–150  
 network load balancers, 160  
 Newsome, Bryan, 140  
 Norton, David  
 strategic management systems and, 3–4  
 strategy maps and, 196, 198

**O**

object visibility, parameters and, 78–79  
 objectives  
 defining for balanced scorecards, 220–221  
 metrics and, 222  
 organizational, 15  
 strategic, customer satisfaction and, 26  
 strategy maps, 200–201  
 Office (Microsoft), business diagrams and, 217–218  
 OLAP (online analytical processing) data source, 130, 135  
 OLTP (online transactional processing)  
 Adventure Works database, 178  
 performance and, 135  
 operations and production, designing for, 11  
 orientation  
 bar graphs and, 95  
 of reports, selecting, 184

**P**

paging, 54–55  
 parameters  
 cascading, 82–83  
 data retrieval and, 76–77  
 data-driven, 79–81  
 object visibility and, 78–79  
 Pareto, Vilfredo, 240  
 Pareto Charts, 240–244  
 Pareto law probability distribution, 240  
 Pareto Principle, 240

`PERCENTILE_CONT()` function, 236

performance, maximizing, 134–136

PerformancePoint

building process and strategy maps and, 218

as a mobile platform, 128–134

`PERIODICITY_HINT` algorithm

parameter, 275

Person, Ron, 201

perspectives

four business perspectives, 219–224

perspective maps (balanced scorecards), 4

strategic function and, 14

strategy maps and, 197–198, 221

PESTEL (Political, Economic, Social, Technology, Environmental and Legal) framework, 198

pixels

charts and graphs and, 90–91

pixel-perfect layouts, 127

placement of metrics and measures, 52–53

planning. *See also* dashboard and scorecard

projects, planning and initiating

for growth and scalability, 158–161

metrics and projects, 28–29

mobile BI and audiences and, 125–126

mobile BI delivery solutions and, 126–127

scope and, 30–31

point layer (maps), 118, 121–122

pointers, in gauges, 111–112

polar charts, 93

polygon layer (maps), 118–123

Porter’s 5 Forces framework, 199

postbacks, 128

power law probability distribution, 240

PowerPoint, building process and strategy

maps and, 217–218

predefined roles (Reporting Services), 163–164

`PredictionKey` function, 274

preview (Reporting Services project), 44–45

`ProcedureCount`, 274–275

process flow maps, 202–205

process flowcharts. *See* process maps

process improvement project

(Contoso Plastic), 212

process maps

basics of, 202

implementing, 214–218

process flow maps, 202–205