Transforming Data into Insightful Action with Visual Studio 2010 & SQL Server 2008 R2

June 2011

As more and more data become easily consumable, business leaders are beginning to understand the power of having that data when and where they need it. Wide data availability means that the right people can translate those data into meaningful information at the right time, creating business insight. In this whitepaper you will see how SQL Server 2008 R2 and Visual Studio 2010 enable you to create solutions that will transform your data into insights and make those insights available to the right people, regardless of where they are in the organizational structure, while conforming to an organization's data access rules. The right people with the right data at the right time: that's transformed business intelligence – business insight for all.

THE RISE OF DATA

Name a big application on the web, and the chances are that its success isn't based solely on its user interface. A good base UI is important, but it has to be coupled with great data and extended by dozens of wildly different UIs accessing the data via an API. Imagine not being able to check your favorite social platform from your phone, from your laptop, and sometimes even in widgets on unrelated web sites that you visit! For web and social applications it's not a single user interface; it's the data, available on multiple, task specific applications that makes the experience compelling.

This move from the primacy of a particular application to the primacy of data is happening in the enterprise world as well. No longer is it acceptable for data to be constrained to a single application. It's too limiting to provide the insight necessary to run a dynamic and competitive organization. Instead, organizations of today are demanding open, consumable data feeds that can be easily integrated into new tools – tools that deliver a competitive advantage.

As more and more data become easily consumable, business leaders are beginning to understand the power of having that data when and where they need it. Wide data availability means that the right people can translate those data into meaningful information at the right time, creating business insight. In order to deliver on that promise, business leaders need to push BI capabilities ever deeper into their organizations. Contrast this with just a few years ago, when people spoke of "business intelligence" and "data analytics" as an important, yet arcane, technology whose outputs were geared to executives. Important patterns were discovered, and critical decisions made. However, critical decisions are made at all levels of the organization and by keeping access at the highest levels of management, lower level decision makers don't have the same level of real-time visibility required to meet customer needs, drive down costs and move the business forward. With SQL Server 2008 R2 and Visual Studio 2010, insights from data can be made available to the right people, regardless of where they are in the organization structure, while conforming to an organization's data access rules. The right people with the right data at the right time: that's transformed business intelligence – business insight for all.

DATA → APPLICATIONS → INFORMATION → ACTION

Information workers make decisions every day that impact the success of their organizations. With fast access to data, better decisions can be made more quickly, helping the entire organization react more quickly to changes in the marketplace.

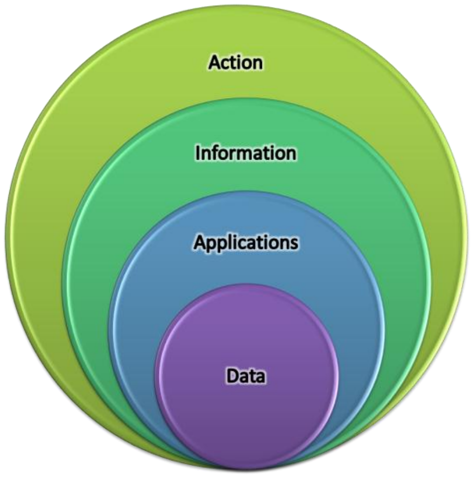


Figure 1 - Moving from Data to Action

But access to data isn't enough. Information workers need to be able to see the data through a variety of lenses. They need the flexibility to see the big picture, drill into specific facets of the data, identify patterns, and discover cause-and-effect relationships. In practice, this can mean viewing processed data in one or more applications, and then drilling into the raw data for specific insight. Whether through a web interface, an Office product, or a custom client application, information workers should have fast access, in familiar interfaces, to the data they need.

The process begins with data. The data should be exposed in easily consumable formats, be accessible quickly, and be protected against corruption or loss. This requires a powerful data store, such as SQL Server 2008 R2.

"We have very knowledgeable, experienced subject matter experts, and we no longer wanted them to totally depend on IT for all aspects of report generation and management. We needed to give them the ability to do self-service analysis and reporting."

Stephen Walker, Enterprise Database Architect, Chevron

Applications should allow for easy access to common views of the data, providing a foundation for gaining insight. Most are created by professional developers working with database professionals. Building high quality applications quickly requires powerful development tools, like Visual Studio 2010 working in concert with SQL Server 2008 R2.

Although prebuilt applications and reports can provide a great deal of information, when knowledge workers pivot on raw data they often come up with impactful insights. Organizations are exposing data to far more people than ever before, giving their subject matter experts the tools they need to do self-service analysis. Enabling knowledge workers to easily access needed data entails two prerequisites: easily accessible data in consumable formats, and powerful, user-friendly analytic tools.

Once people in the organization have the right information, based on timely and actionable data, they can confidently take action. And this action can lead to dramatic results.

THE DATA FOUNDATION

Building data-centric solutions is hard, and one of the primary reasons is that data are often locked away in databases, inaccessible to all but a small core of database experts and professional developers. Let's first take a look at the database management experience for the database experts.

Enhanced Database Management

For data-centric applications, management of the data infrastructure can be difficult. These applications not only can contain substantial amounts of data, they are often composed of data residing in different databases, and possibly even different machines. Yet all these resources need to be managed in a coordinated way.

To solve this problem, SQL Server 2008 R2 introduced the SQL Server Utility Explorer, a tool for centrally visualizing and managing multiple database servers, environments, and resources as a managed group. This tool also supports collaboration between development and IT departments through dashboards and views depicting database and application health. Regular use of the Utility Explorer helps identify resource utilization bottlenecks and consolidation opportunities, resulting in increased performance, easier management, and better uptime.

Better Data Integrity: Master Data Services

One of the most difficult aspects of providing an authoritative source of data is keeping the data up-to-date and consistent. Over time data become outdated, invalid duplicate data are introduced, data are corrupted through human error, and inconsistency can spring up as data formats change. In many cases what is needed is an authoritative data stream that can be used across the organization.

Master Data Services (MDS), a feature introduced in SQL Server 2008 R2, ensures that critical data are managed from inception to deletion to ensure integrity. Using MDS, administrators can proactively enforce data quality rules, define workflows around data changes, and create a single, authoritative source for multiple consumers. Coupled with other features that expose data in standard formats, MDS can ensure standardized data are widely available to the right consumers.

Better Data Integrity: Version Controlled Database Schema

When database schemas are maintained outside of version control, they rapidly become disconnected from changes in application code. This is especially true when application code is branched to support multiple versions of the same code. Changes made directly to a database, whether in test or production, are then disassociated from the appropriate version of the application code that is using the databases. This increases the complexity of rolling back an application, debugging an older version or understanding why changes were made to a schema.

With Visual Studio 2010, you can reverse engineer your existing databases, pulling the schema into a version controlled database project. From there you can safely do parallel development alongside the other developers on the team. This allows both database and application code versions to be architected, coded, tested and deployed as a versioned unit, keeping your application and database in sync. It also means that you can modify your schema "offline" and deploy all your related changes in a single, atomic transaction.

"In the past, I would have hundreds of script files stored on a file system, generally not under source control. Now, we have a secure, managed way to store the source code for our entire reporting data warehouse, including all database schemas, integration packages, report definitions, and deployment scripts and utilities… Team Foundation Server gives us the security and confidence of knowing exactly what we deployed during this upgrade and allows for controlled, managed changes."

Jeff Lynch, IT Manager, Gulf Coast Seal

Better Data Integrity: Management of Database Schema Changes

Another key component of data integrity is managing database schema changes in a controlled manner. One common challenge encountered by development and database professionals is comparing two databases – not only in terms of schema, but also in the lookup data. For instance, as version 2 of an application moves to production the database schema will likely will have changed, but so will the lookup data which provides the sales geographies, or the product categories. Visual Studio 2010 provides the tools to compare both schema and data between two databases, and to generate the scripts necessary to modify the schema or data in the target database to match that in the source database.

These scripts can even be included in scheduled or on-demand automated builds to automatically upgrade the database schemas in the development or test environments. Deploying database changes nightly can provide a high level of confidence to the database team that the schema is solid and the deployment of that schema is working as planned.

Expanded Deployment Options

Organizations today are faced with numerous deployment choices: on-premises, infrastructure as a service (IaaS) and platform as a service (PaaS). This means organizations can deploy their databases internally, on a machine hosted as a VHD in Azure, or on top of SQL Azure. Where on-premises and IaaS allow full control over the SQL Server 2008 R2 administration, SQL Azure allows an organization to let go of the IT administration tasks and focus on database creation.

For on-premises hosting SQL Server 2008 R2 provides many different editions, including Datacenter and Parallel Data Warehouse editions for massive scalability and high performance. For Platform as a Service hosting, you can choose to deploy your application directly to SQL Azure. Practically, this means deployment can be either on-premises or in the cloud. In fact, with just a few simple configuration changes, an on-premises database solution can be deployed or migrated to SQL Azure, and vice-versa.

A key component in enabling the streamlined deployment, management, and upgrade of database applications, both on-premises and in SQL Azure, is the introduction in SQL Server 2008 R2 of the Data-Tier Application, or DAC. A DAC defines the entire Database Engine schema and instance objects (such as tables, views, and logins) required to support the application. This new mechanism supports a single, versioned file to hand-off to the DBA for simple deployment to either on-premises data centers or to the cloud.

Visual Studio 2010 and SQL Server 2008 R2 provide a great deal of flexibility for working with DACs. For example, DACs can be created from existing database applications. In addition, it is possible to do the maintenance, release, and deployment of new and subsequent versions. In fact, when a subsequent version of a data-tier application is built into a new DAC package, the same package can be used both for deploying a new instance of the DAC or upgrading an existing instance to the new version. The Upgrade Data-tier Application wizard compares the schema of the existing DAC instance and the new DAC package, and dynamically performs the actions needed to transform the existing DAC instance to the new version of the DAC. You can even use Visual Studio 2010 to create a single deployment package that can deploy the same schema to either SQL Server 2008 R2 or SQL Azure.

THE INTERSECTION OF DATA & APPLICATIONS: ENABLING INSIGHTFUL ACTION

Getting to insightful action requires a comprehensive platform that integrates the database management and tooling experience to enable developers to build applications that provide end users with information when and where they need it.

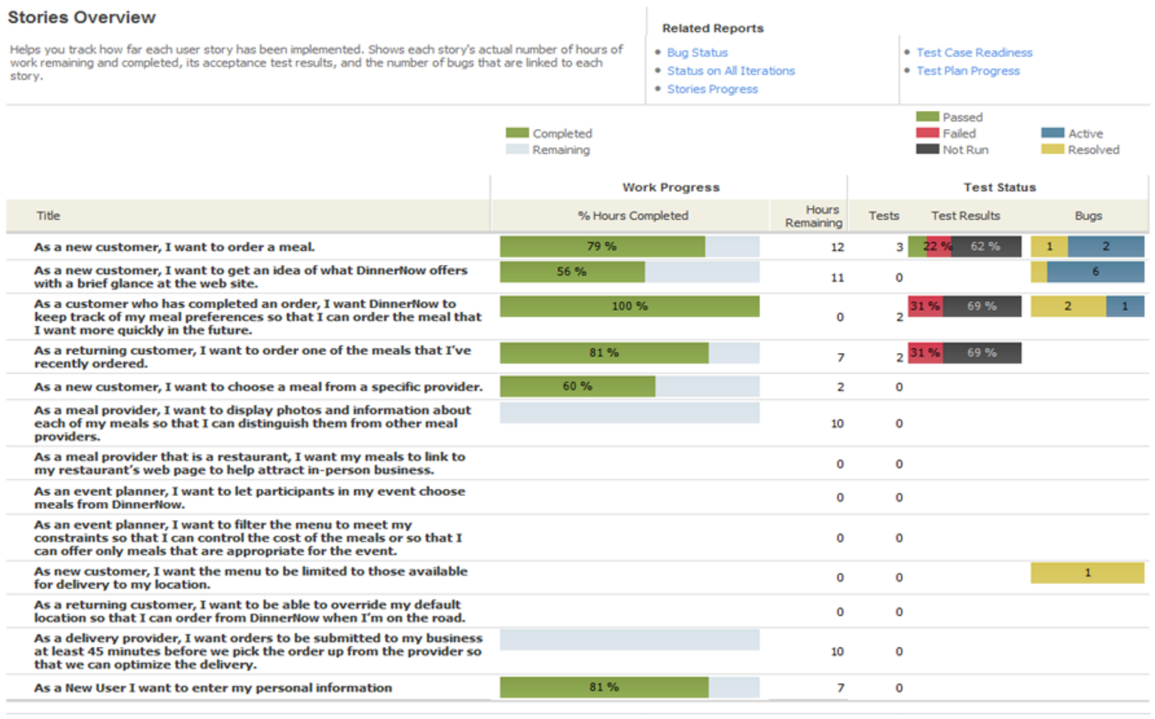
INTEGRATING DATABASE DEVELOPMENT INTO APPLICATION LIFECYCLE MANAGEMENT

One of the major features of Visual Studio 2010 is the Application Lifecycle Management (ALM) tooling that helps teams integrate and track their work. This customizable ALM support is provided by Visual Studio 2010 Team Foundation Server, which supports source control, work assignment and tracking, build automation, requirements traceability, defect tracking, systems verification, and reporting.

Visual Studio 2010 can provide some of the same benefits application developers rely on daily to the database administrator. For instance, an automated build that provides for regular code compilation, integration, deployment and testing, can also include automated database deployment. With the advanced database build and deploy features in Team Foundation Server 2010, database schema and data changes can be built, deployed and tested nightly with the application code, providing a deeper confidence level across the entire application.

Visual Studio 2010 also provides specific tools for the database developer, helping integrate them into the overall development effort, and aligning their work with the rest of the team. DBAs and other IT practitioners have often worked largely “without a net,” making changes to databases or systems directly. Although this is an easy and common approach, it dramatically increases risk. The impact of an error can be far reaching. Visual Studio 2010 brings order to this risky approach by integrating the DBAs and IT professionals with the rest of the development team, as well as the business. By tracking database changes against specific requirements and business needs, the database team is able to understand why and when specific changes were made to the system. This also allows them to correlate their changes with changes made by the application developers, further supporting traceability. This team integration is key for reducing production surprises, customer dissatisfaction, and unplanned IT efforts.

The traceability enabled by tracking changes to requirements has far reaching implications for the entire development team. Not only do application and database developers use TFS, but also testers and business analysts. This means we can have an integrated view of the quality of the entire application. Dozens of out of the box reports track progress, quality and risk. The figure below shows one such report that tracks work effort, test progress and bugs against each individual requirement.



Stories Overview

Helps you track how far each user story has been implemented. Shows each story's actual number of hours of work remaining and completed, its acceptance test results, and the number of bugs that are linked to each story.

Related Reports

- Bug Status
- Status on All Iterations
- Stories Progress
- Test Case Readiness
- Test Plan Progress

🟩 Completed
⬜ Remaining

🟩 Passed
🟥 Failed
⬛ Not Run

🟦 Active
🟨 Resolved

	Work Progress		Test Status		
Title	% Hours Completed	Hours Remaining	Tests	Test Results	Bugs
As a new customer, I want to order a meal.	79 %	12	3	22 % 62 %	1 2
As a new customer, I want to get an idea of what DinnerNow offers with a brief glance at the web site.	56 %	11	0		6
As a customer who has completed an order, I want DinnerNow to keep track of my meal preferences so that I can order the meal that I want more quickly in the future.	100 %	0	2	31 % 69 %	2 1
As a returning customer, I want to order one of the meals that I've recently ordered.	81 %	7	2	31 % 69 %	
As a new customer, I want to choose a meal from a specific provider.	60 %	2	0		
As a meal provider, I want to display photos and information about each of my meals so that I can distinguish them from other meal providers.		10	0		
As a meal provider that is a restaurant, I want my meals to link to my restaurant's web page to help attract in-person business.		0	0		
As an event planner, I want to let participants in my event choose meals from DinnerNow.		0	0		
As an event planner, I want to filter the menu to meet my constraints so that I can control the cost of the meals or so that I can offer only meals that are appropriate for the event.		0	0		
As new customer, I want the menu to be limited to those available for delivery to my location.		0	0		1
As a returning customer, I want to be able to override my default location so that I can order from DinnerNow when I'm on the road.		0	0		
As a delivery provider, I want orders to be submitted to my business at least 45 minutes before we pick the order up from the provider so that we can optimize the delivery.		10	0		
As a New User I want to enter my personal information	81 %	7	0		

Figure 2 - End-to-End Traceability

Visual Studio 2010 ALM tooling makes your development safer, more consistent, and faster, while still allowing you to adapt quickly to change. You can use the tools with traditional or agile processes, or anything in between. The collaboration tooling and practice support in team projects built on Team Foundation Server 2010 can help your team execute on complex change, delivering broad IT systems and complexity with greater awareness, control, and predictability.

Powerful Development Tools

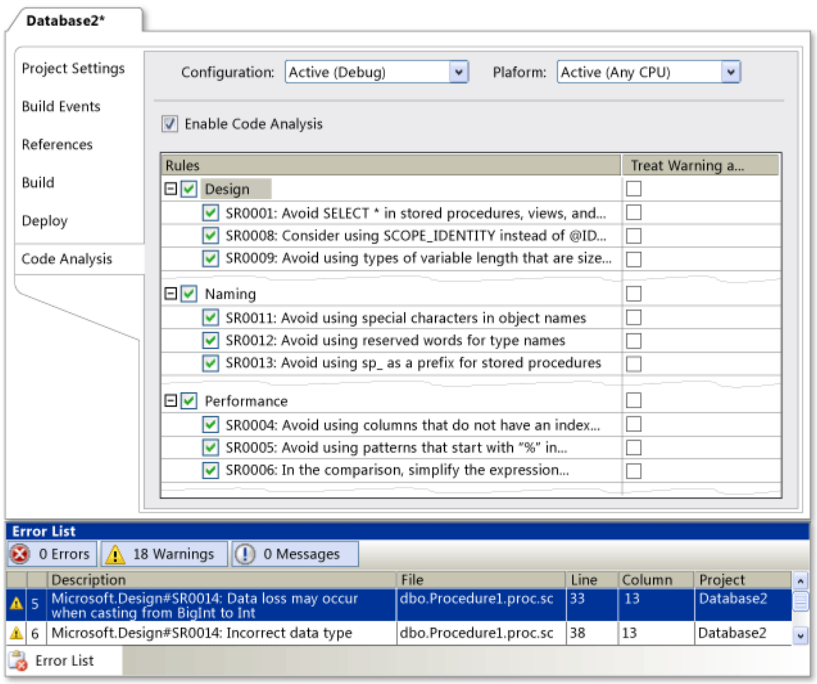
Once your database schema is safely protected in version control, you can begin to use some of the more advanced database development features, such as automated refactoring, database unit testing, static analysis, and tier interaction profiling. These tools enable database developers to rapidly modify, test and validate their existing and newly created databases. In the past, this type of work was often error prone and time intensive. Without effective tools, refactoring a database object was fraught with dangers and possible unintended consequences. With the tools available in Visual Studio 2010, databases can now be edited and tested with confidence.

Refactoring allows you to rename a database object or move it to a different schema, automatically fully qualify all database objects, rename all database or server references, and even automatically expand all wildcards in SELECT statements.

Data generation provides a powerful tool for generating data for use in testing. This is critical in cases where personally identifiable information (PII) is maintained in the production database. The data generation tools allow you to base your test data on the production data, while scrubbing PII into usable test data. Thus, developers and testers can test against realistic data, while still maintaining the privacy of the user data in the production database.

Database unit testing provides tools to create automated unit tests to test the correctness of stored procedures and user defined functions. These unit tests can then be executed nightly or on-demand to validate that any schema, data or business logic changes didn't negatively impact the database.

Static analysis can examine Transact-SQL for design, naming and performance problems. This helps identify problems proactively, keeping your database more maintainable and performant.


Database2*

Project Settings
Build Events
References
Build
Deploy
Code Analysis

Configuration: Active (Debug) Plaform: Active (Any CPU)

[✓] Enable Code Analysis

Rules	Treat Warning a...
☑ Design	☐
☑ SR0001: Avoid SELECT * in stored procedures, views, and...	☐
☑ SR0008: Consider using SCOPE_IDENTITY instead of @ID...	☐
☑ SR0009: Avoid using types of variable length that are size...	☐
☑ Naming	☐
☑ SR0011: Avoid using special characters in object names	☐
☑ SR0012: Avoid using reserved words for type names	☐
☑ SR0013: Avoid using sp_ as a prefix for stored procedures	☐
☑ Performance	☐
☑ SR0004: Avoid using columns that do not have an index...	☐
☑ SR0005: Avoid using patterns that start with "%" in...	☐
☑ SR0006: In the comparison, simplify the expression...	☐

Error List

0 Errors | 18 Warnings | 0 Messages

		Description	File	Line	Column	Project
⚠	5	Microsoft.Design#SR0014: Data loss may occur when casting from BigInt to Int	dbo.Procedure1.proc.sc	33	13	Database2
⚠	6	Microsoft.Design#SR0014: Incorrect data type	dbo.Procedure1.proc.sc	38	13	Database2

Error List

Figure 3 - Static Analysis for Databases

Tier Interaction Profiling shows the number of times a database query or stored procedure was executed by your application, and allows you to analyze the amount of time you spend inside of ADO.NET for that query. If you are testing a web application, the data is broken down further, and you can see the number of times a page was called, the processing time for that page, and the queries called by that particular page. You can even drill into the text of the queries. This can help you refactor your code to limit the number of database calls your application is making, or identify areas where performance improvements could make a positive difference.

"SQL Server 2008 R2 is a game-changer for us .... We can 'wow' our customers by giving them location-based functionality that brings the 'where' to the 'what' for their applications."

Peter Hammond, President, CyberSavvy

Improved Transact-SQL

In many cases, application or validation logic is coded directly in stored procedures. This has the power to tie business logic very close to the data store, providing very high performance. However, working directly with T-SQL can be intimidating for people not well versed in its syntax, and frustrating due to the lack of complex data types.

SQL Server 2008 R2 introduces enhancements to make it easier to work with T-SQL. There is the inclusion of fully integrated full-text search, sparse columns and filtered indexes to optimize speed when searching through data. In addition, Transact-SQL has been enhanced with rich data types to manage virtually any type of data, including precise date and time, XML data, and even external documents and files.

But perhaps the most dramatic of the new rich data types is spatial information, allowing geographic data to be precisely contained in a single field, and queried using improvements in T-SQL. With spatial information, you can represent locations directly and write queries that return 'nearby' records – Return all customers within 100 miles of Customer A.

In addition, IntelliSense is available to assist in writing T-SQL logic within Visual Studio 2010. This can dramatically speed development of new stored procedures, user defined functions and queries, especially for less experienced database developers.

Better Object-Relational Mapping Support

Bridging the gap between relational database schemas and an applications object oriented design has traditionally been difficult. Direct access to the database tends to be brittle, and require recompilation for even small changes in the database structure; building a data access layer, or even reusing an existing one, is still labor intensive; and many solutions are very difficult to test effectively with automated testing tools.

Visual Studio 2010 and .NET 4.0 provide the Entity Framework, an advanced object-relational mapping (ORM) tool dramatically simplifying interaction between code and database. The beauty of Entity Framework is that it supports the ease of drag and drop data modeling, while still including a number of very powerful features, including lazy loading of data, n-tier support, testable interfaces, and full support for language integrated query (LINQ).

Entity Framework also supports different styles of coding. For developers who prefer to create their data models first, they can take advantage of the graphical tools to generate .NET classes from existing database schemas. These classes already include loading, support for LINQ and other benefits, and can also be extended directly for even more granular control. For developers who prefer to first design an object oriented structure, Entity Framework allows a database schema to be generated from .NET classes by relying on convention over configuration, allowing software developers to create and deploy database schemas without the need to write T-SQL.

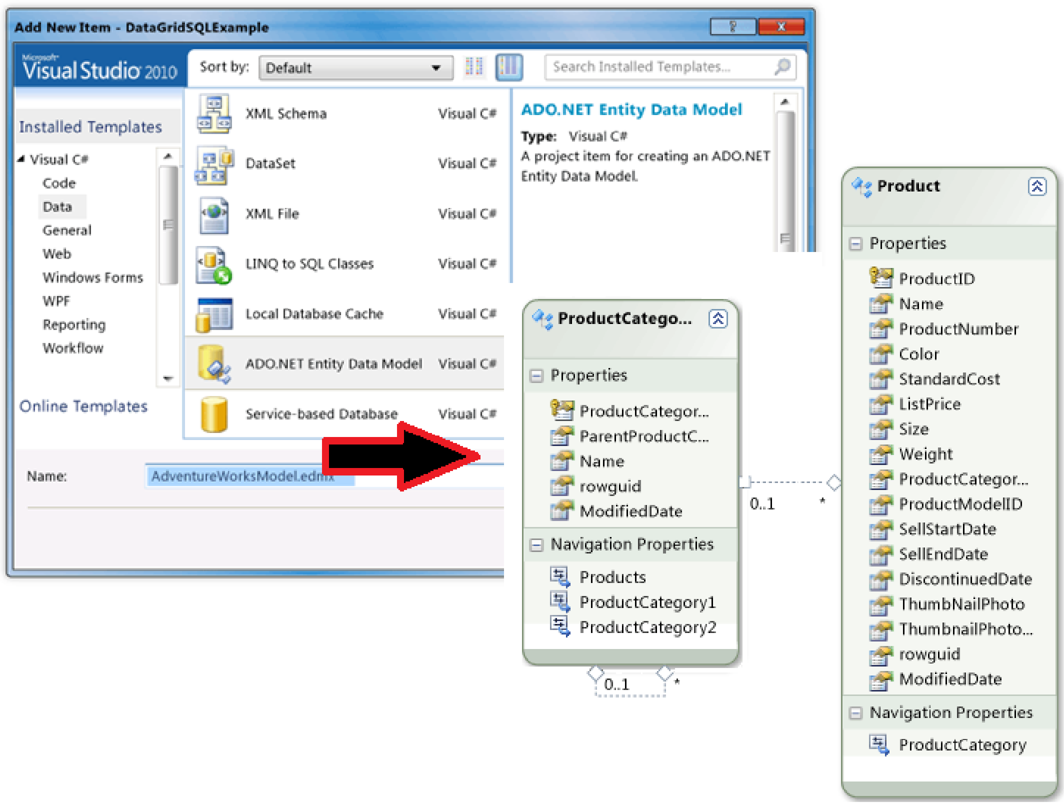


Figure 4 - Entity Framework with Drag and Drop

Automatic User Interface Generation that's Customizable and Extensible

One of the simplest ways to link a UI to a database is to drag-and-drop directly from a database query or stored procedure to the design surface. This is supported across all the major user interfaces, including Web, Silverlight, WebForms and WPF. However, for those building full-fledged, data-centric web sites, a core fixture of nearly all organizations today, there is an even simpler solution – ASP.NET Dynamic Data.

For highly data-centric applications that need to expose database data quickly and easily, Visual Studio 2010 and ASP.NET Dynamic Data can be the solution. Very rapid application development is made possible by a rich scaffolding framework and reuse of existing ASP.NET data controls. By simply registering a LINQ to Entity Framework data model to a project (created in minutes), you can create a fully functional web site that supports all CRUD (create, read, update, and delete) operations with a click of a button. The resulting site and rules for transforming data into visual representations are fully editable, allowing you complete flexibility. For instance, you could always transform a US Social Security Number from a 9 digit field, into a format that includes dashes (*xxx-xx-xxxx*).

Past experiences with other web site generation tools leave many people critical of its automated nature, believing that the tight coupling between user interface and data makes maintenance difficult. But ASP.NET Dynamic Data doesn't necessarily tie UI directly to a database. Instead, it can rely on a well-constructed Entity Framework model, thus providing a layer of indirection between the application and data, providing a surprising amount of power and customizability for something requiring so little work. Indeed, ASP.NET Dynamic Data is very flexible, supporting a customizable UI, customizable data field validation, and rich, customizable data table display support.

STANDARD DATA FORMATS TO MORE EASILY EXPOSE DATA: ENABLING INSIGHTFUL ACTION

Existing reports are an excellent means of providing information in a graphical format, but business users sometimes have a need to access raw data to manipulate as they see fit. Enabling 'self-service' scenarios is critical to create a more agile organization.

There are numerous techniques for exposing SQL Server 2008 R2 data in easily consumed formats, but the most popular is OData. OData stands for "Open Data" and is an open standard for sharing data through HTTP. It allows users and applications to access data published through the OData endpoint, and allow querying of data using REST (a format for interacting with data using URLs). One of the easiest ways of exposing data is through Reporting Services. Reports created with SQL Server Reporting Services optionally expose their underlying data as an OData stream. This allows developers and information workers to use the data "underneath" a report in their own applications, reports and ad-hoc analysis.

Many applications and development frameworks already support consuming OData. For instance, PowerPivot for Excel 2010 can consume OData, merge it with other data sources, and allow end users to manipulate the data in an ad-hoc fashion. In addition, most web browsers already access OData directly, and developers have libraries for Silverlight 4, PHP, Windows Phone 7, and many more platforms.

As you'll see in the next section, this capability of exposing data as OData has powerful implications for organizations that are moving critical decision data to ever lower levels.

BETTER INSIGHT – REPORT BUILDER 3.0, EXCEL, POWERPIVOT AND SHAREPOINT

The exposure of data in reports is a powerful mechanism for understanding data, but it is no match for the dynamic nature of experts parsing data on the fly. An expert with the right data, coupled with the right tool, can discover insights that would be impossible even with a well-designed set of pre-planned reports.

In the past, allowing end users access to raw data meant giving them permissions to access the data store directly. It also meant providing them with the appropriate IT resources to assist in manipulating, transforming and displaying the data.

Today, the deliberate manipulation of data by end users is a reality. Stephen Walker, Enterprise Database Architect for Chevron Data Management Solutions, puts the need clearly: "We have very knowledgeable, experienced subject matter experts, and we no longer wanted them to totally depend on IT for all aspects of report generation and management. We needed to give them the ability to do self-service analysis and reporting."[1]

Microsoft Excel, PowerPivot, Report Server 3.0, and SharePoint all work together to provide end users with powerful ways to consume, analyze and display data.

POWERFUL AD-HOC DATA ANALYSIS

Microsoft Excel has always been a powerful tool for data analysis. Most data savvy end-users are familiar with using pivot tables to analyze their data quickly, and pivot charts to display those results, as well as the vast array of data transformations available. With their latest release, Microsoft has added considerable power to the venerable tool.

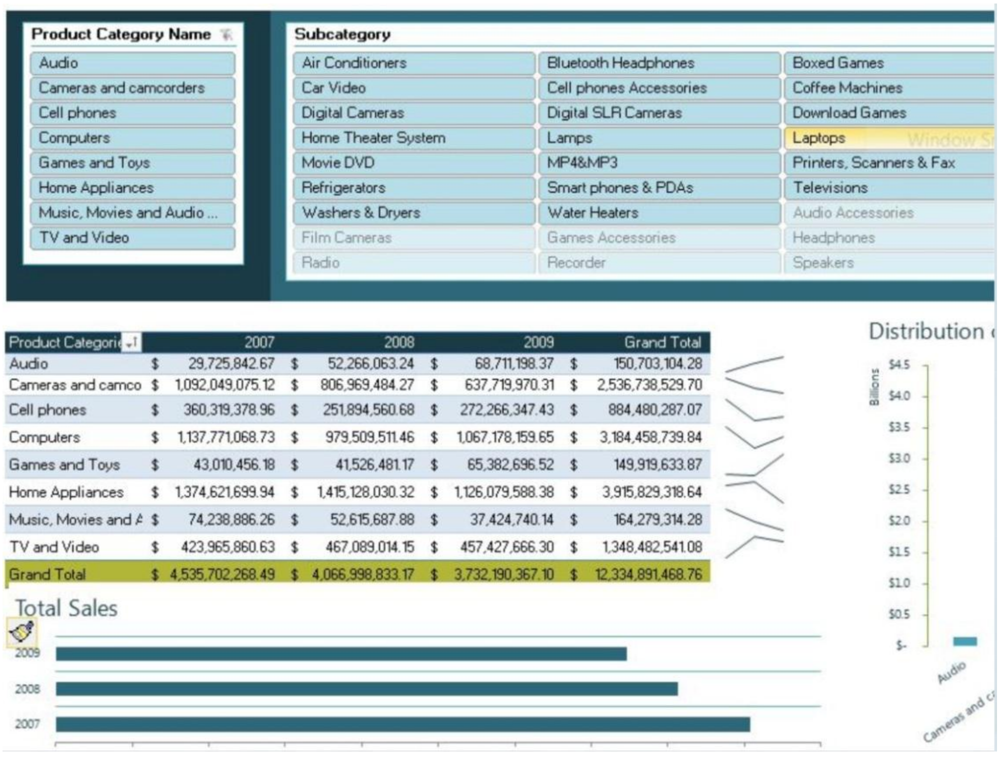
One of the most effective additions to Excel is PowerPivot. PowerPivot extends Excel and allows users to load even the largest data sets from virtually any source, process massive amounts of data in seconds (while leveraging multi-core processors), and use a variety of new analytical capabilities such as Data Analysis Expressions. And users do this all within the familiar Excel interface.

As one of its data sources, PowerPivot can consume data provided in the OData format. This means that end users can find an existing report hosted in Reporting Services and pull the underlying data into

[1] http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?casestudyid=4000007043

Excel. This data can then be merged from data pulled from another source, such as an online customer relationship management tool, and analyzed together. By merging multiple data streams, end users can build up powerful solutions that look at data in new and unique ways.

Finally, Excel and PowerPivot both provide excellent data visualization tools, enabling the creation of dashboards, charts and other data representations. This allows any new discovery or insight to be more easily understood by others.



Product Categori	2007	2008	2009	Grand Total
Audio	$ 29,725,842.67	$ 52,266,063.24	$ 68,711,198.37	$ 150,703,104.28
Cameras and camco	$ 1,092,049,075.12	$ 806,969,484.27	$ 637,719,970.31	$ 2,536,738,529.70
Cell phones	$ 360,319,378.96	$ 251,894,560.68	$ 272,266,347.43	$ 884,480,287.07
Computers	$ 1,137,771,068.73	$ 979,509,511.46	$ 1,067,178,159.65	$ 3,184,458,739.84
Games and Toys	$ 43,010,456.18	$ 41,526,481.17	$ 65,382,696.52	$ 149,919,633.87
Home Appliances	$ 1,374,621,699.94	$ 1,415,128,030.32	$ 1,126,079,588.38	$ 3,915,829,318.64
Music, Movies and A	$ 74,238,886.26	$ 52,615,687.88	$ 37,424,740.14	$ 164,279,314.28
TV and Video	$ 423,965,860.63	$ 467,089,014.15	$ 457,427,666.30	$ 1,348,482,541.08
Grand Total	$ 4,535,702,268.49	$ 4,066,998,833.17	$ 3,732,190,367.10	$ 12,334,891,468.76

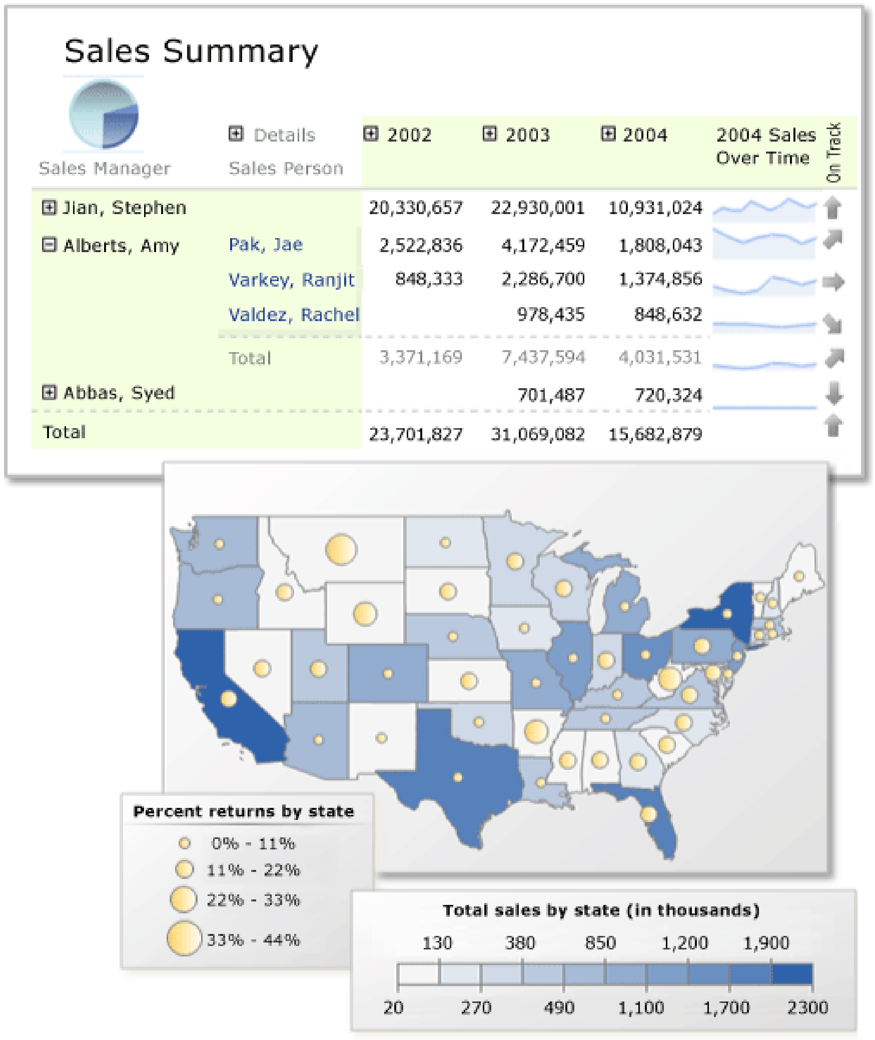
Figure 5 - PowerPivot in Excel

Easy End-user Report Creation

Reporting data in visual formats is one of the most common ways for applications to enable insightful action. People are generally able to grasp the meaning of a graph or image, far faster and with deeper impact, than they are a tabular layout of data. However, creating compelling visuals can be difficult.

Reporting has taken a big step forward in SQL Server 2008 R2 and Visual Studio 2010. Beyond the existing report wizards, OLAP cube capabilities, and other features, SQL Server Reporting Services (SSRS) now supports Report Builder 3.0.

Report Builder 3.0 provides an Office-like experience for business users to create new reports themselves, without the need to rely on IT support. Users specify where to get the data, which data to get, and how to display the data. For retrieving data, it provides a vast array of possible data sources, including data directly from SharePoint lists and SQL Azure. For selecting the data to show, it provides interactive tools such as query builders. For displaying data in a compelling format, it brings a suite of rich visualization tools, including geospatial relationships and visuals, sparklines, and shared report parts. Plus, you can easily enable drill-through to provide deeper information to report consumers.



Sales Summary

Sales Manager	⊞ Details Sales Person	⊞ 2002	⊞ 2003	⊞ 2004	2004 Sales Over Time	On Track
⊞ Jian, Stephen		20,330,657	22,930,001	10,931,024		
⊟ Alberts, Amy	Pak, Jae	2,522,836	4,172,459	1,808,043		
	Varkey, Ranjit	848,333	2,286,700	1,374,856		
	Valdez, Rachel		978,435	848,632		
	Total	3,371,169	7,437,594	4,031,531		
⊞ Abbas, Syed			701,487	720,324		
Total		23,701,827	31,069,082	15,682,879		

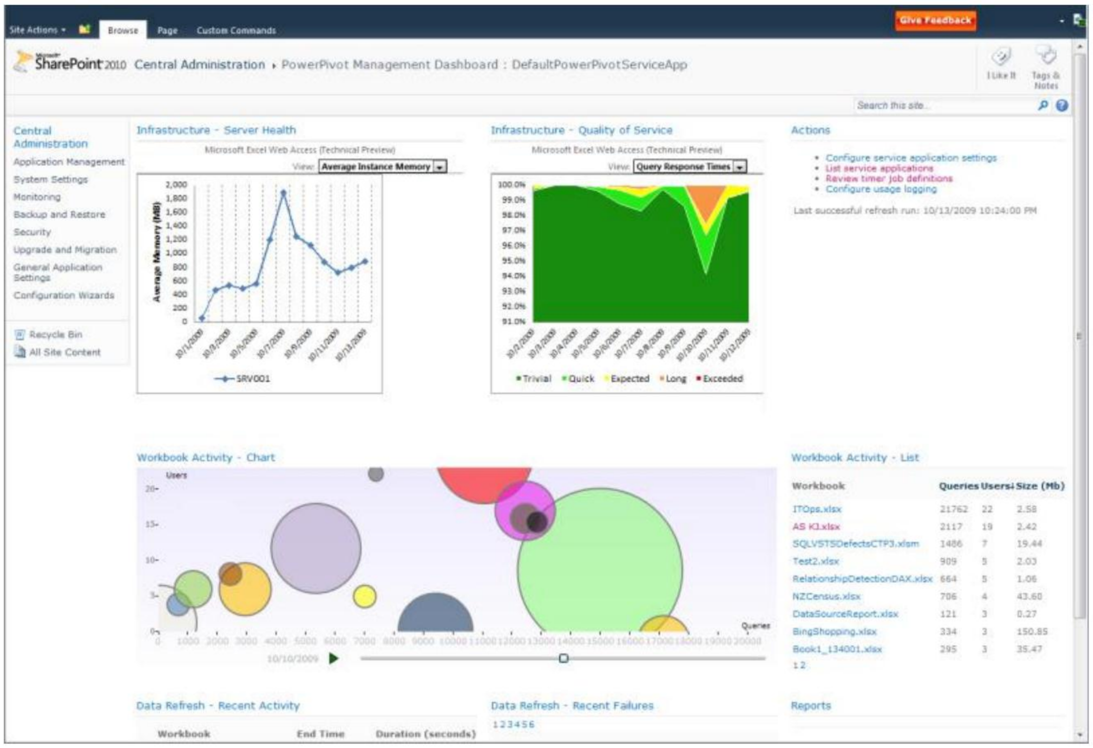
Figure 6 - Report Builder 3.0 in Action

Simplified Insight Sharing

The ability to quickly, efficiently and effectively analyze sets of data into business insight is powerful in its own right. But when that insight can be easily shared across the organization and remixed in different ways by others, that insight can be magnified and turned into concrete action. It's then, at the moment of action, that the business begins to benefit from this insight.

Historically, business insights spread slowly through an organization. To get the information into the right hands, people resorted to emailing Excel spreadsheets, asking IT to build a web site for broad distribution, or hosting the spreadsheet on a portal. With Visual Studio 2010 and SharePoint, however, the difficulty in sharing insights is a thing of the past.

For the end-user working with Excel or PowerPivot, using SharePoint 2010 is both easy and effective. With SharePoint 2010 and Excel Data Services, users can host any graphs or other Excel visualizations directly on a SharePoint page. This means that end-users can create portals, exposing many different views of their data, in a SharePoint page of their own creation, all without needing IT support. PowerPivot provides additional support for SharePoint, and provides an easy mechanism to not only host visualizations of the data, but also allow data manipulation directly on a SharePoint page. In addition, PowerPivot for SharePoint ships with a dashboard that lets IT managers monitor and track usage over time. This capability is critical, since it allows an organization to freely expose end-user PowerPivot solutions to the broader community, and then discover which of the solutions is being used more widely over time. In this way, new solutions can become part of the standard set of reports and solutions, and may even become mission critical in their own right.


Site Actions
Browse
Page
Custom Commands
Give Feedback
SharePoint 2010 Central Administration › PowerPivot Management Dashboard : DefaultPowerPivotServiceApp
I Like It
Tags & Notes
Search this site...
Central Administration
Application Management
System Settings
Monitoring
Backup and Restore
Security
Upgrade and Migration
General Application Settings
Configuration Wizards
Recycle Bin
All Site Content
Infrastructure - Server Health
Microsoft Excel Web Access (Technical Preview)
View: Average Instance Memory
Average Memory (MB)
SRV001
Infrastructure - Quality of Service
Microsoft Excel Web Access (Technical Preview)
View: Query Response Times
Trivial
Quick
Expected
Long
Exceeded
Actions
Configure service application settings
List service applications
Review timer job definitions
Configure usage logging
Workbook Activity - Chart
Users
Queries
10/10/2009
Workbook Activity - List
Reports
Data Refresh - Recent Activity
Data Refresh - Recent Failures
1 2 3 4 5 6

Last successful refresh run: 10/13/2009 10:24:00 PM

Workbook	Queries	Users↓	Size (Mb)
ITOps.xlsx	21762	22	2.58
AS K1.xlsx	2117	19	2.42
SQLVSTSDefectsCTP3.xlsm	1486	7	19.44
Test2.xlsx	909	5	2.03
RelationshipDetectionDAX.xlsx	664	5	1.06
NZCensus.xlsx	706	4	43.60
DataSourceReport.xlsx	121	3	0.27
BingShopping.xlsx	334	3	150.85
Book1_134001.xlsx	295	3	35.47

1 2

Workbook	End Time	Duration (seconds)

Figure 7 - PowerPivot Management Dashboard

Users of Report Builder 3.0 can also upload their solutions to SharePoint, and have them displayed throughout the portal, thus gaining much of the same benefits as Excel and PowerPivot solutions. They also have an additional option. Visual Studio 2010 includes several Report Viewer components that can display Reporting Services reports in any type of user interface, including web, windows, and WPF. The Report Viewer component for the web provides a number of powerful extensibility points, including the ability to script with JavaScript, use AJAX for asynchronous data updates, and many options for extended control into the look-and-feel and behavior of the report. In addition, it is supported across a large number of browsers including IE 6 and above, Firefox 3.5 and above, and Safari 4. Although this generally requires development support, it allows an end-user created report to be wrapped into a web page or smart client application that can be deployed widely.

Whether through Excel, PowerPivot or Report Builder, creating and sharing new data analysis solutions has never been easier. Both SharePoint 2010 and Visual Studio 2010 provide the capabilities end-users and developers need to widely expose these end-user solutions.

BEGIN TODAY

The tools you need to deploy "better insight" through your organization exist today. By integrating the features and tools of both SQL Server 2008 R2 and Visual Studio 2010, you can begin creating today the solutions that will transform your data into insight and action. In this whitepaper you have seen how intersection of data and applications on the Microsoft platform can provide you with:

- Higher quality for your data and applications
- Robust development and reporting tools spanning the entire organization
- Faster time to market for applications
- Self-service reporting

You've also seen how data, exposed widely throughout your organization, can revolutionize the way you do business. Start turning data into action today.

Learn more at: http://www.microsoft.com/visualstudio and http://www.microsoft.com/sqlserver