

msdn magazine



XAML Data Binding
in Xamarin.....22

Introducing the FREE DevExpress Grid Built and Optimized for Xamarin.Forms



Get your free copy of the
DevExpress Grid for Xamarin.Forms
devexpress.com/xamarin

The FREE

DevExpress Grid for Xamarin.Forms

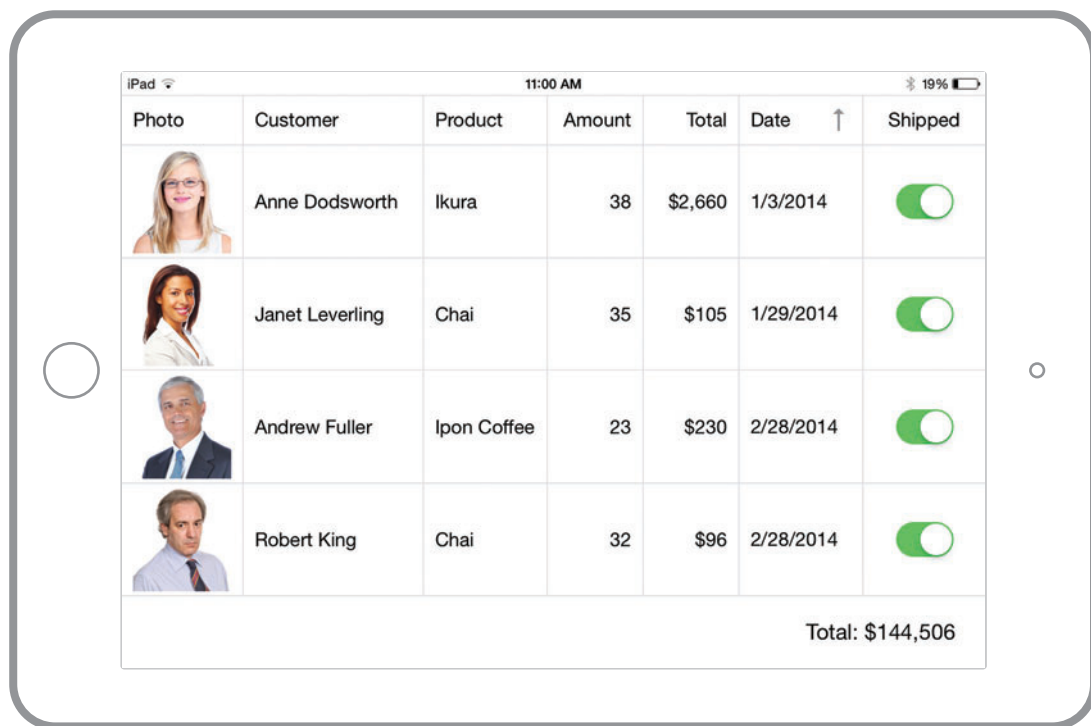






Photo	Customer	Product	Amount	Total	Date ↑	Shipped
	Anne Dodsworth	Ikura	38	\$2,660	1/3/2014	<input checked="" type="checkbox"/>
	Janet Leverling	Chai	35	\$105	1/29/2014	<input checked="" type="checkbox"/>
	Andrew Fuller	Ipon Coffee	23	\$230	2/28/2014	<input checked="" type="checkbox"/>
	Robert King	Chai	32	\$96	2/28/2014	<input checked="" type="checkbox"/>
Total: \$144,506						

devexpress.com/xamarin

msdn magazine



XAML Data Binding
in Xamarin.....22

Implementing and Using Data Binding in Xamarin Laurent Bugnion	22
MapReduce Without Hadoop Using the ASP.NET Pipeline Doug Duerner and Yeon-Chang Wang	28
Building Responsive Web Sites with Bootstrap Keith Pijanowski	40
Create Your Own Visual Studio Templates with SideWaffle Tyler Hughes and Sayed Hashimi	52
Event Hubs for Analytics and Visualization, Part 2 Bruno Terkaly	66

COLUMNS

UPSTART

Job Hunting
Ryder Donahue, page 6

CUTTING EDGE

CQRS for the
Common Application
Dino Esposito, page 10

DATA POINTS

An Overview of Microsoft
Azure DocumentDB
Julie Lerman 16

TEST RUN

Firefly Algorithm Optimization
James McCaffrey, page 72

DON'T GET ME STARTED

Natives, Immigrants
and Symbionts
David Platt, page 80

Axosoft helps you see software development in a totally new way.

Reimagine project management with Axosoft: a super-powered agile tool, that helps your dev team ship awesome software faster than a speeding bullet.

Plan your releases effectively, adapt to your process, and keep everyone in the loop. With Axosoft, empower your dev team to perform heroic software feats!



Check out item status and progress at a glance with our Kanban board view

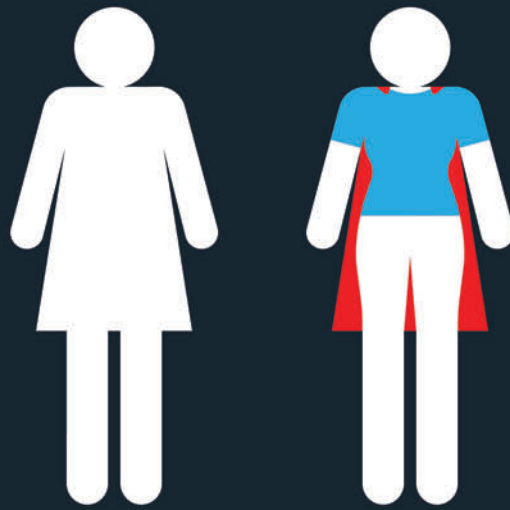


Automated burndown charts, velocity, projected ship dates and more

See Axosoft in action—watch our video adventure at axosoft.com/MSDN!



take back
your development time



#ItWasNeverADress

It Was Never a Dress is an invitation to shift perceptions and assumptions about women and the audacious, sensitive, and powerful gestures they make every single day. In science, technology, arts, mathematics, politics, houses of worship, on the streets, and in our homes, insightful women are often uninvited, overlooked, or just plain dismissed. Through storytelling, community building, innovation and creative disruptions, **It Was Never a Dress** will foster necessary conversations, vital voices, and images from around the world that honor ALL women. When we see women differently... we see the world differently!

Visit [**ItWasNeverADress.com**](http://ItWasNeverADress.com) to read and submit stories, images, and ideas about perceptions and realities, and to buy merchandise, where a portion of proceeds will go towards scholarships for those underrepresented in STEAM fields: science, technology, engineering, arts, and mathematics.

msdn magazine

JUNE 2015 VOLUME 30 NUMBER 6

Director Keith Boyd

Editorial Director Mohammad Al-Sabt mmeditor@microsoft.com

Site Manager Kent Sharkey

Editorial Director, Enterprise Computing Group Scott Bekker

Editor in Chief Michael Desmond

Features Editor Lafe Low

Features Editor Sharon Terdeman

Group Managing Editor Wendy Hernandez

Senior Contributing Editor Dr. James McCaffrey

Contributing Editors Rachel Appel, Dino Esposito, Kenny Kerr, Julie Lerman, Ted Neward, David S. Platt, Bruno Terkaly, Ricardo Villalobos

Vice President, Art and Brand Design Scott Shultz

Art Director Joshua Gould



President
Henry Allain

Chief Revenue Officer
Dan LaBianca

Chief Marketing Officer
Carmel McDonagh

ART STAFF

Creative Director Jeffrey Langkau
Associate Creative Director Scott Rovin
Senior Art Director Deirdre Hoffman
Art Director Michele Singh
Assistant Art Director Dragutin Cvijanovic
Graphic Designer Erin Horlacher
Senior Graphic Designer Alan Tao
Senior Web Designer Martin Peace

PRODUCTION STAFF

Director, Print Production David Seymour
Print Production Coordinator Anna Lyn Bayaua

ADVERTISING AND SALES

Chief Revenue Officer Dan LaBianca
Regional Sales Manager Christopher Kourtoglou
Account Executive Caroline Stover
Advertising Sales Associate Tanya Egenolf

ONLINE/DIGITAL MEDIA

Vice President, Digital Strategy Becky Nagel
Senior Site Producer, News Kurt Mackie
Senior Site Producer Gladys Rama
Site Producer Chris Paoli
Site Producer, News David Ramel
Senior Site Administrator Shane Lee
Site Administrator Biswarup Bhattacharjee
Senior Front-End Developer Rodrigo Munoz
Junior Front-End Developer Anya Smolinski
Executive Producer, New Media Michael Domingo
Office Manager & Site Assoc. James Bowling

LEAD SERVICES

Vice President, Lead Services Michele Imgrund
Senior Director, Audience Development & Data Procurement Annette Levee
Director, Audience Development & Lead Generation Marketing Irene Fincher
Director, Client Services & Webinar Production Tracy Cook
Director, Lead Generation Marketing Eric Yoshizuru
Director, Custom Assets & Client Services Mallory Bundy
Editorial Director, Custom Content Lee Pender
Senior Program Manager, Client Services & Webinar Production Chris Flack
Project Manager, Lead Generation Marketing Mahal Ramos
Coordinator, Lead Generation Marketing Obum Ukabam

MARKETING

Chief Marketing Officer Carmel McDonagh
Vice President, Marketing Emily Jacobs
Senior Manager, Marketing Christopher Morales

ENTERPRISE COMPUTING GROUP EVENTS

Senior Director, Events Brent Sutton
Senior Director, Operations Sara Ross
Director, Event Marketing Merikay Marzoni
Events Sponsorship Sales Danna Vedder
Senior Manager, Events Danielle Potts
Coordinator, Event Marketing Michelle Cheng
Coordinator, Event Marketing Chantelle Wallace



Chief Executive Officer
Rajeev Kapur

Chief Operating Officer
Henry Allain

Senior Vice President & Chief Financial Officer
Richard Vitale

Executive Vice President
Michael J. Valenti

Vice President, Information Technology & Application Development
Erik A. Lindgren

Chairman of the Board
Jeffrey S. Klein

ID STATEMENT MSDN Magazine (ISSN 1528-4859) is published monthly by 1105 Media, Inc., 9201 Oakdale Avenue, Ste. 101, Chatsworth, CA 91311. Periodicals postage paid at Chatsworth, CA 91311-9998, and at additional mailing offices. Annual subscription rates payable in US funds are: U.S. \$35.00, International \$60.00. Annual digital subscription rates payable in U.S. funds are: U.S. \$25.00, International \$25.00. Single copies/back issues: U.S. \$10, all others \$12. Send orders with payment to: MSDN Magazine, P.O. Box 3167, Carol Stream, IL 60132, email MSDNmag@1105service.com or call (847) 763-9560. POSTMASTER: Send address changes to MSDN Magazine, P.O. Box 2166, Skokie, IL 60076. Canada Publications Mail Agreement No: 40612608. Return Undeliverable Canadian Addresses to Circulation Dept. or XPO Returns: P.O. Box 201, Richmond Hill, ON L4B 4R5, Canada.

Printed in the U.S.A. Reproductions in whole or part prohibited except by written permission. Mail requests to "Permissions Editor," c/o MSDN Magazine, 4 Venture, Suite 150, Irvine, CA 92618.

LEGAL DISCLAIMER The information in this magazine has not undergone any formal testing by 1105 Media, Inc. and is distributed without any warranty expressed or implied. Implementation or use of any information contained herein is the reader's sole responsibility. While the information has been reviewed for accuracy, there is no guarantee that the same or similar results may be achieved in all environments. Technical inaccuracies may result from printing errors and/or new developments in the industry.

CORPORATE ADDRESS 1105 Media, 9201 Oakdale Ave. Ste 101, Chatsworth, CA 91311 www.1105media.com

MEDIA KITS Direct your Media Kit requests to Chief Revenue Officer Dan LaBianca, 972-687-6702 (phone), 972-687-6799 (fax), dlabianca@1105media.com

REPRINTS For single article reprints (in minimum quantities of 250-500), e-prints, plaques and posters contact: PARS International Phone: 212-221-9595. E-mail: 1105reprints@parsintl.com. www.magreprints.com/QuickQuote.asp

LIST RENTAL This publication's subscriber list, as well as other lists from 1105 Media, Inc., is available for rental. For more information, please contact our list manager, Jane Long, Merit Direct. Phone: 913-685-1301; E-mail: jloug@meritdirect.com; Web: www.meritdirect.com/1105

Reaching the Staff

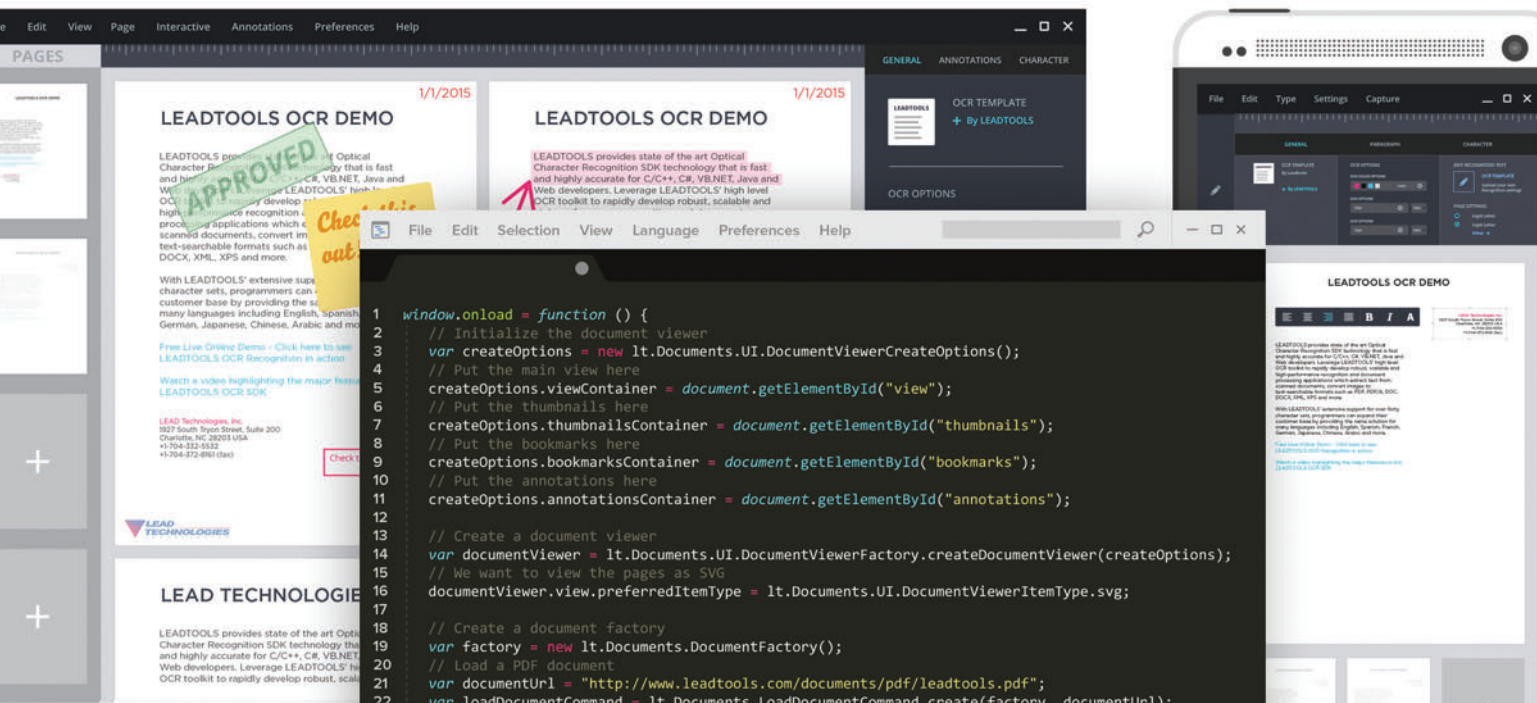
Staff may be reached via e-mail, telephone, fax, or mail. A list of editors and contact information is also available online at Redmondmag.com. E-mail: To e-mail any member of the staff, please use the following form: FirstInitial.LastName@1105media.com Irvine Office (weekdays, 9:00 a.m. – 5:00 p.m. PT) Telephone 949-265-1520; Fax 949-265-1528 4 Venture, Suite 150, Irvine, CA 92618 Corporate Office (weekdays, 8:30 a.m. – 5:30 p.m. PT) Telephone 818-814-5200; Fax 818-734-1522 9201 Oakdale Avenue, Suite 101, Chatsworth, CA 91311 The opinions expressed within the articles and other contents herein do not necessarily express those of the publisher.



DOCUMENT VIEWER & CONVERTER

Create robust, fully-featured document viewing and editing solutions for any document, vector and raster file format
with only a few lines of code!

- ◇ View document, vector and raster image formats in one viewer
- ◇ Document-specific interactive tools including Select Text, Search Text, Pan and Zoom
- ◇ Scalable Vector Graphics (SVG) display enables infinite zoom without loss of display quality or aliasing
- ◇ Easy to use and customizable programming interface available in .NET and HTML5/JavaScript
- ◇ Viewer seamlessly integrates with LEADTOOLS Document Converter to convert any file format into document formats including, PDF, PDF/A, DOX/DOCX, XLS/XLSX, PPT/PPTX, XPS, Text, RTF, HTML, MOBI, ePub and more
- ◇ Perform document-to-document conversion at 100% accuracy without the need for OCR





Defragmenting Windows

Just about every year I travel to the Microsoft Build conference to take in the Microsoft message to developers, and just about every year I leave impressed with the event, the audience and the speakers. Build is one of those all-in moments for Microsoft. The pace of work around Redmond heightens ahead of the conference, as product teams finalize their plans and presentations. There's an urgency around Build, one that comes from understanding how important the conference is to developer decision making over the year to come, and beyond.

And make no mistake, this Build conference was incredibly important. The universal apps strategy that Microsoft articulated around Windows 10 during the three days in San Francisco is both elegant and focused. It offers developers consistent, durable targets in the form of the Universal Windows Platform (UWP), device-specific API extensions and robust XAML-based UIs. The new strategy ably integrates both native Windows Runtime apps and—I hesitate to say the word—legacy .NET and Win32 applications as co-first class citizens. Gone are the days of Silverlight and XNA on Windows Phone, and the odd Berlin Wall that stood between Windows Runtime and traditional Windows apps under Windows 8 and 8.1 on the desktop.

Fixing the divide that Microsoft created within itself will take time, but the payoff is potentially immense. Developers will soon be able to write Windows apps that can run, unmodified, on all flavors of Windows, from Windows PCs and tablets to Windows Phones, to Xbox game controllers and a variety of other Windows-supported device targets. At the same time, Microsoft is working to make Windows a convenient target for iOS and Android mobile developers.

Al Hilwa, program director for Software Development Research at research firm IDC, describes the new Universal Windows apps model as “a triumph.”

“I think if you look at the totality of what they have done in terms of increasing the attractiveness of the platform to developers, it is one of the most massive and widespread outreach efforts I have ever seen in the industry,” Hilwa says. “They have constructed pathways to bring code from various platforms into Windows, including classic Win32 and .NET apps. They have brought support to a broader

set of programming languages ranging from Web and C++ through to Objective C with Microsoft's tools. They have reached out to the Android ecosystem to support working with other IDEs like Eclipse and IntelliJ, which is the basis for Android Studio today. And, finally, they have partnered deeply with Xamarin to make sure that C# (and F#) developers have a strong toolchain to bring their apps to other platforms.”

Gone are the days of Silverlight
and XNA on Windows Phone,
and the odd Berlin Wall that
stood between Windows
Runtime and traditional
Windows apps under Windows 8
and 8.1 on the desktop.

In short, Microsoft is defragmenting the Windows platform, and in the process offering developers more reasons than ever to support it. The scope and scale of the plan is impressive, says Billy Hollis, a partner at Next Version Systems and an expert on UI development.

“Microsoft's keynote at Build had more innovation than anything I've seen from them since PDC 2000,” Hollis says, referring to the Professional Developers Conference where the company first announced the Microsoft .NET Framework.

I wonder if, 10 years from now, we'll look back on Build 2015 the same way we do at PDC 2000 today, as the moment when Microsoft rolled out a game-changing dev strategy. What do you think of the Microsoft Universal Windows apps vision? E-mail me at mmeditor@microsoft.com.

Visit us at msdn.microsoft.com/magazine. Questions, comments or suggestions for *MSDN Magazine*? Send them to the editor: mmeditor@microsoft.com.

© 2015 Microsoft Corporation. All rights reserved.

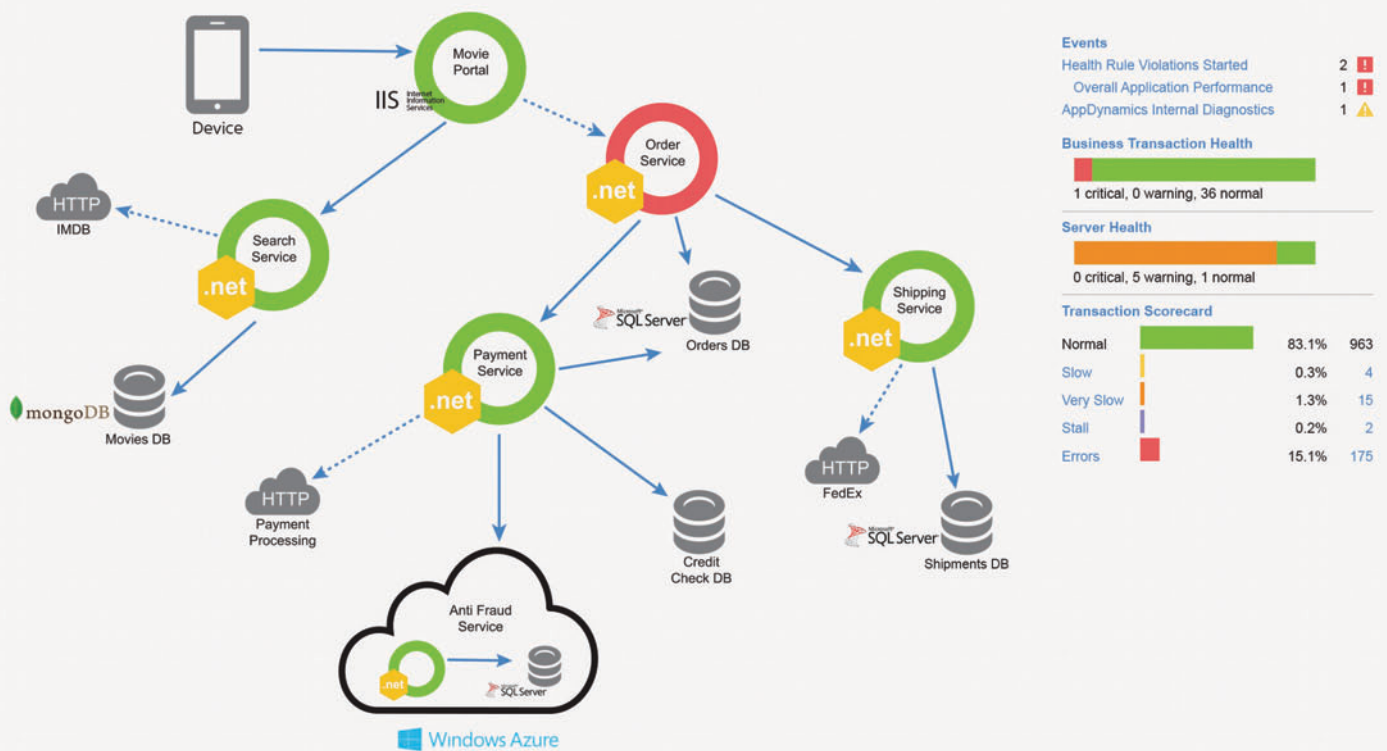
Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, you are not permitted to reproduce, store, or introduce into a retrieval system *MSDN Magazine* or any part of *MSDN Magazine*. If you have purchased or have otherwise properly acquired a copy of *MSDN Magazine* in paper format, you are permitted to physically transfer this paper copy in unmodified form. Otherwise, you are not permitted to transmit copies of *MSDN Magazine* (or any part of *MSDN Magazine*) in any form or by any means without the express written permission of Microsoft Corporation.

A listing of Microsoft Corporation trademarks can be found at microsoft.com/library/toolbar/3.0/trademarks/en-us.mspx. Other trademarks or trade names mentioned herein are the property of their respective owners.

MSDN Magazine is published by 1105 Media, Inc. 1105 Media, Inc. is an independent company not affiliated with Microsoft Corporation. Microsoft Corporation is solely responsible for the editorial contents of this magazine. The recommendations and technical guidelines in *MSDN Magazine* are based on specific environments and configurations. These recommendations or guidelines may not apply to dissimilar configurations. Microsoft Corporation does not make any representation or warranty, express or implied, with respect to any code or other information herein and disclaims any liability whatsoever for any use of such code or other information. *MSDN Magazine*, MSDN and Microsoft logos are used by 1105 Media, Inc. under license from owner.

There's nothing about .NET that AppDynamics doesn't see.

AppDynamics gives you the visibility to take command of your .NET application's performance, no matter how complicated or distributed your environment is.



When your business runs on .NET and SQL Server, count on AppDynamics to give you the complete visibility you need to be sure they are delivering the performance and business results you need — no matter how complex, distributed or asynchronous your environment, live 'in production' or during development.

See every line of code. Get a complete view of your environment with deep code diagnostics and auto-discovery. Understand performance trends with dynamic baselining. And drastically reduce time to root cause and remediation.

See why the world's largest .NET deployments rely on the AppDynamics Application Intelligence Platform. Sign up for a FREE trial today at www.appdynamics.com/Microsoft.

Job Hunting

In my last column, I wrote about what it was like for me to adjust to a new career at Microsoft, and gave some advice that I hope will benefit new employees in the future. But to take advantage of that advice, you need to land a job first.

For college students entering the workforce, I can't emphasize enough how important it is to start your job search early. I began job hunting in the first semester of my senior year, and had secured a full-time job before the start of my last semester. By contrast, I saw how difficult it was for some of my colleagues who didn't begin their search until after they graduated.

I also urge developers—whether new to the industry or simply changing jobs—to carefully research the companies they hope to join. I spent a fair amount of time studying various tech companies before applying for openings, and that helped me determine that Microsoft would be a good fit for me. You'll want to find a workplace that best fits your needs and personality, and provides a compatible corporate culture. Maybe you want to work late nights on cutting-edge technology, while living on campus with access to great free facilities, or maybe you want to be able to go home after work and spend time with your family. Whatever your preference, realize that there's a tech company for everyone, and they all seem to be hiring.

Whatever job you're applying for will almost certainly have you doing something you may not have done a lot in college, which is working on a team.

And keep in mind, when you walk into a job interview, be sure to judge them just as hard as they are judging you. Landing a job won't make you happy if you don't like the company you're working for, which brings me to the next topic: preparing for the job interview.

There are countless resources out there on things like, "How to nail the tech interview," or "101 common coding interview questions," which you may or may not find helpful. The key is to prepare for an interview so you can confidently present yourself. The most important takeaway I had from my interviews was not how tough

they were technically, but how they tested other aspects of my skillset and character beyond coding, problem solving, and basic knowledge of fundamental algorithms. In short, interviewers are trying to decide if you are someone with whom they want to work.

You may be the best programmer at your university, but if you're a pain to work with, no one will hire you. Whatever job you're applying for will almost certainly have you doing something you may not have done a lot in college, which is working on a team. Tech companies don't create the most lucrative or innovative companies in the world by having thousands of programmers coding all day in isolation. It's an entirely collaborative process. In my job, I rarely spend longer than 30 minutes at work before I have to ask someone a question, run something by a product manager or do a code review.

Of course, the technical/coding aspects of the interview remain important, but it really isn't something about which you should be too concerned. Every interviewer handles the process differently, but most are just trying to determine if you have strong problem-solving skills. Most likely you'll be solving a problem on a whiteboard, or on paper, which for me was somewhat daunting, given that my whiteboard handwriting is terrible. Luckily, as one of my interviewers put it, "Don't worry about your handwriting, we actually normally write code on these crazy things called *computers* most of the time here at Microsoft anyway."

One way to prepare is to find some sample technical interview questions and practice solving them on a whiteboard while timing yourself. Don't expect to anticipate the specific topics on which you'll be challenged. Rather, the practice reps can help you feel confident about the process during the interview.

In the end, I had a great time during my interviews, even though I arrived too early and almost froze to death in the below-freezing November weather (I'm from Hawaii, and my wardrobe was utterly insufficient). My interview team was very friendly and tried to make me feel comfortable. I was actually surprised at how quickly it was all over because I was having such a good time. And while I was fortunate to actually land the first job I interviewed for, if I hadn't got it, I would have had plenty of time to explore other options.

Just like anything in life, the harder you work the bigger the payoff, and when it comes to something as important as starting a career, you want everything to go as smoothly as possible. ■

RYDER DONAHUE is a software developer engineer at Microsoft. Originally from the Hawaiian Islands, he now resides in Redmond, Wash., with his fiancée and their cat, Marbles.

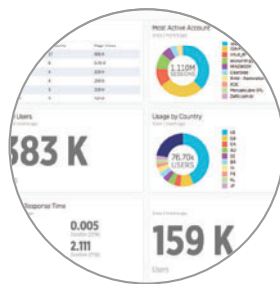
One source of truth

See all your data. Boost performance. Drive accountability for everyone.



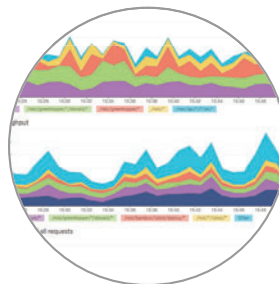
Mobile Developers

End-to-end visibility,
24/7 alerting, and
crash analysis.



Front-end Developers

Deep insights into
your browser-side
app's engine.



IT Operations

Faster delivery.
Fewer bottlenecks.
More stability.



App Owners

Track engagement.
Pinpoint issues.
Optimize usability.

Move from finger-pointing blame to data-driven accountability.
Find the truth with a single source of data from multiple views.

newrelic.com/truth



Open, Create, Convert, Print & Save Files

from within your *own* applications.

> ASPOSE.TOTAL

allows you to process these file formats:

- Word documents
- Excel spreadsheets
- PowerPoint presentations
- PDF documents
- Project documents
- Visio documents
- Outlook emails
- OneNote documents



DOC XLS PPT PDF EML
PNG XML RTF HTML VSD
BMP & barcode images.

SCAN FOR
20% SAVINGS!



ASPOSE
Your File Format APIs

Contact Us:

US: +1 888 277 6734

EU: +44 141 416 1112

AU: +61 2 8003 5926

sales@aspose.com

Helped over 11,000 companies and over 250,000 users work with documents in their applications.



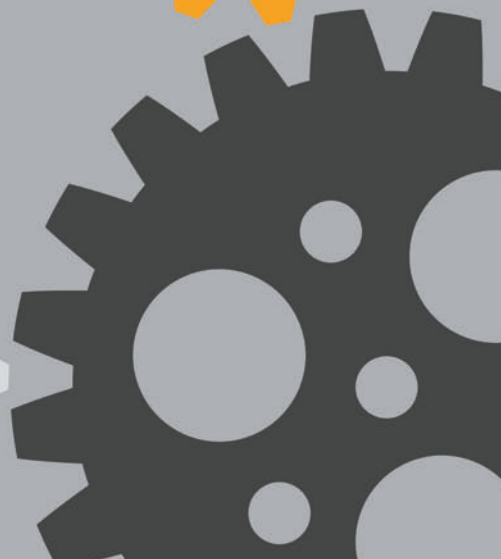
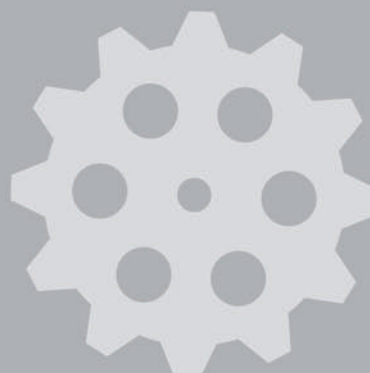
.NET, Java & Cloud

Your File Format APIs



GET STARTED NOW

- Free Trial
- 30 Day Temp License
- Free Support
- Community Forums
- Live Chat
- Blogs
- Examples
- Video Demos





CQRS for the Common Application

Domain-driven design (DDD) surfaced about a decade ago, inspiring software developers and architects. Beyond its concrete merits and faults, DDD incarnates an old dream to which anyone from the early days of the object-oriented paradigm can relate: the dream of building applications around a comprehensive object model to address all stakeholder requirements and concerns.

CQRS is an approach to software development that pays off regardless of the project's perceived complexity.

Over the past decade, many developers embarked in projects following DDD guidelines. Some projects have been successful, some not. The truth is an all-encompassing object model to cover any functional or nonfunctional aspect of a software system is a wonderful utopia. Especially in these crazy days of advanced UX, frequently changing business models and swinging requirements, a solid and stable object model is practically an illusion.

Recently, another method with another acronym started gaining momentum—Command and Query Responsibility Segregation (CQRS). CQRS isn't the latest cool toy of software pundits. It's not even as complex as most available examples imply. Simply stated, CQRS is a concrete implementation pattern that's probably most appropriate for nearly any type of software application, regardless of the expected lifespan and complexity.

There's not just one way of doing CQRS—there at least three different flavors. You could even nickname them after common marketing rules of hotel rooms and drinks: regular, premium and deluxe. Do a quick search for CQRS examples and articles and most of what you'll find falls in the deluxe category. This is effectively too complex and overkill for most common applications.

CQRS is an approach to software development that pays off regardless of the project's perceived complexity. At the end of the day, CQRS for the common application is

simply a rework of the classic multi-layered architecture that leaves the door open for more impactful changes and evolution.

Commands and Queries

While developing the Eiffel programming language in the late 1980s, Bertrand Meyer came to the conclusion that software has commands that alter system state and queries that read system state. Any software statement should either be a command or a query—not a combination of the two. There's a nicer way to express the same concept: Asking a question shouldn't change the answer. CQRS is a more recent re-formulation of the same core principle: Commands and queries are distinct things and should be implemented separately.

The logical separation between commands and queries doesn't show up very well if the two groups of actions are forced to use the same programming stack and the same model. This is especially true in complex business scenarios. A single model—whether an object model, functional model or anything else—can quickly become unmanageable. The model grows exponentially large and complex and absorbs time and budget, but it never works the way it should.

There's not just one way of doing CQRS.

The separation pursued by CQRS is achieved by grouping query operations in one layer and commands in another layer. Each layer has its own model of data, its own set of services and is built using

its own combination of patterns and technologies. More important, the two layers may even be on two distinct tiers and be optimized separately without affecting each other. **Figure 1** lays the groundwork for the CQRS architecture.

Simply recognizing that commands and queries are two different things has a deep impact on software architecture. For example, it suddenly becomes simpler to envision and code each domain layer. The domain layer in the command stack is only concerned with the data and business and security rules needed to conduct tasks. The domain layer in the query stack, on the other hand, may

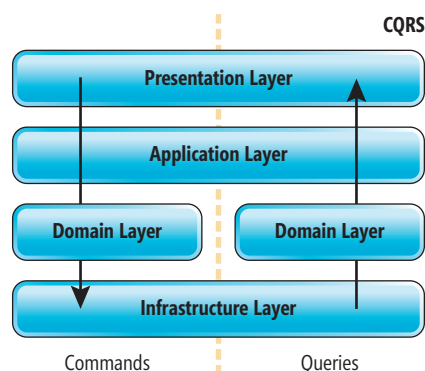


Figure 1 A Canonical and Multi-Layered CQRS Architecture

We didn't invent the Internet...

...but our components help you power the apps that bring it to business.



TOOLS • COMPONENTS • ENTERPRISE ADAPTERS

- **E-Business**
AS2, EDI/X12, NAESB, OFTP ...
- **Credit Card Processing**
Authorize.Net, TSYS, FDMS ...
- **Shipping & Tracking**
FedEx, UPS, USPS ...
- **Accounting & Banking**
QuickBooks, OFX ...
- **Internet Business**
Amazon, eBay, PayPal ...
- **Internet Protocols**
FTP, SMTP, IMAP, POP, WebDav ...
- **Secure Connectivity**
SSH, SFTP, SSL, Certificates ...
- **Secure Email**
S/MIME, OpenPGP ...
- **Network Management**
SNMP, MIB, LDAP, Monitoring ...
- **Compression & Encryption**
Zip, Gzip, Jar, AES ...



The Market Leader in Internet Communications, Security, & E-Business Components

Each day, as you click around the Web or use any connected application, chances are that directly or indirectly some bits are flowing through applications that use our components, on a server, on a device, or right on your desktop. It's your code and our code working together to move data, information, and business. We give you the most robust suite of components for adding Internet Communications, Security, and E-Business Connectivity to

any application, on any platform, anywhere, and you do the rest. Since 1994, we have had one goal: to provide the very best connectivity solutions for our professional developer customers. With more than 100,000 developers worldwide using our software and millions of installations in almost every Fortune 500 and Global 2000 company, our business is to connect business, one application at a time.

connectivity
powered by 

To learn more please visit our website →

www.nsoftware.com

be as simple as a direct SQL query over an ADO.NET connection.

When you place security checks at the presentation gate, the query stack becomes a thin wrapper around Entity Framework or whatever you use to query data. Within each domain layer, you're also free to shape data in close resemblance with the domain needs without duplicating or replicating data to accommodate variegated presentation and business needs.

When DDD first appeared, it was about tackling complexity in the heart of software development. Along the way, though, practitioners faced loads of complexity. Many thought it was simply part of the business domain. Instead, most of that complexity resulted from the Cartesian product of queries and commands. Separating commands from queries reduces complexity by an order of magnitude. In rough math terms, you can compare CQRS and a comprehensive domain model approach in terms of $N+N$ versus $N \times N$.

How Do You Start Doing CQRS?

You can turn a basic create, read, update, delete (CRUD) system into a CQRS-inspired system. Suppose you have a canonical ASP.NET MVC Web application that collects user data to display in various forms. This is what most applications do, and what any architect knows how to build quickly and effectively. You can rewrite this with CQRS in mind. You'll be surprised to see the changes required are minimal, but the benefits you can get are potentially unlimited.

Simply recognizing that commands and queries are two different things has a deep impact on software architecture.

Your canonical system is organized in layers. You have application services invoked directly from controllers that orchestrate use cases. Application services (often referred to as worker services) live in the Web server side-by-side with controllers. With respect to **Figure 1**, application services form the application layer. The application layer is the platform from which you run commands and queries against the rest of the system. Applying CQRS means you'll use two distinct middle tiers. One tier takes care of commands that alter the system state. The other retrieves the data. **Figure 2** shows the architecture diagram on a sample ASP.NET MVC project.

You create a couple of class library projects—query stack and command stack—and reference both from the main Web server project.

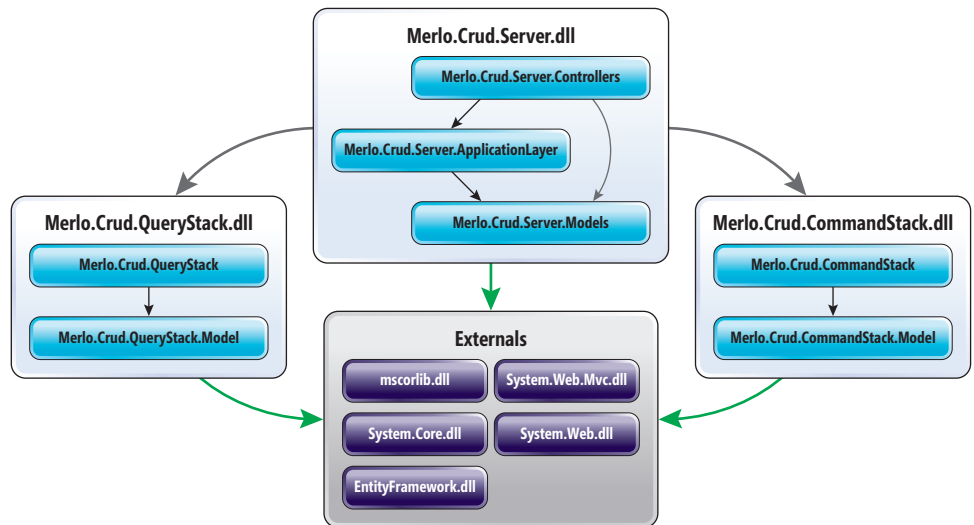


Figure 2 The CQRS Architecture for an ASP.NET MVC Project

The Query Stack

The query stack class library is only concerned with data retrieval. It uses a data model that matches the data used in the presentation layer as closely as possible. You hardly need any business rules here as they typically apply to commands that alter the state.

The domain model pattern made popular by DDD is essentially a way to organize the domain logic. When you make queries from the front end, you're only dealing with part of the application logic and use cases. The term business logic usually results from the union of application-specific logic with invariant domain logic. Once you know the persistence format and the presentation format, all you do is map data as you would in a plain-old ADO.NET/SQL query.

It's useful to recall that any code you can invoke from the application layer represents the system's business domain. Therefore, it's the invariant API that expresses the system's core logic. Ideally, you should ensure no inconsistent and incongruent operations are even possible through the exposed API. So to enforce the read-only nature of the query stack, add a wrapper class around the default Entity Framework context object to connect to the database, as shown in the following code:

```
public class Database : IDisposable
{
    private readonly QueryDbContext _context = new QueryDbContext();

    public IQueryable<Customer> Customers
    {
        get { return _context.Customers; }
    }

    public void Dispose()
    {
        _context.Dispose();
    }
}
```

The Matches class is implemented as a `DbSet<T>` collection in the base `DbContext` class. As such, it provides full access to the underlying database and you can use it to set up queries and update operations via LINQ to Entities.

The fundamental step toward setting up a query pipeline is permitting access to the database for queries only. This is the role of the wrapper class, where Matches is exposed as an `IQueryable<T>`. The

NEW RELEASE



GrapeCity ComponentOne Studio 2015 V1

from \$1,315.60



.NET Controls for Serious Application Development

- Solutions for Data Visualization, Data Management, Reporting and Business Intelligence
- Hundreds of .NET UI controls for all Visual Studio platforms
- Built-in themes and an array of designers for custom styling
- Adaptive, mobile-friendly and touch-enabled controls

BEST SELLER



Help & Manual Professional

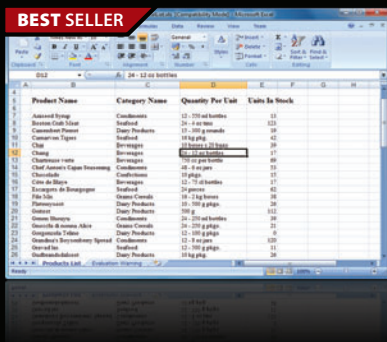
from \$583.10



Easily create documentation for Windows, the Web and iPad.

- Powerful features in an easy accessible and intuitive user interface
- As easy to use as a word processor, but with all the power of a true WYSIWYG XML editor
- Single source, multi-channel publishing with conditional and customized output features
- Output to HTML, WebHelp, CHM, PDF, ePUB, RTF, e-book or print
- Styles and Templates give you full design control

BEST SELLER



Aspose.Total for .NET

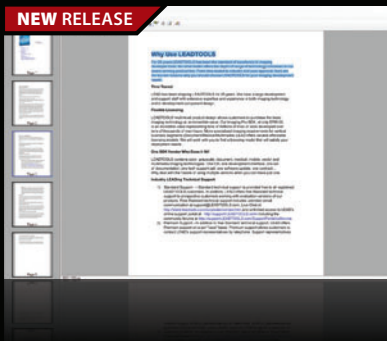
from \$2,449.02



Every Aspose .NET component in one package.

- Programmatically manage popular file formats including Word, Excel, PowerPoint and PDF
- Work with charts, diagrams, images, project plans, emails, barcodes, OCR and OneNote files alongside many more document management features in .NET applications
- Common uses also include mail merging, adding barcodes to documents, building dynamic reports on the fly and extracting text from most document types

NEW RELEASE



LEADTOOLS PDF Pro SDK V19

from \$1,495.00 SRP



Add powerful PDF read, write, view & editing functionality to desktop, tablet & mobile applications.

- View PDF files as raster in .NET, WinRT and C/C++
- Merge, Split, Convert, Linearize and Distill PDF files
- Read and Extract text, hyperlinks, metadata and more from PDF
- Create the smallest file possible utilizing advanced PDF Optimizer
- Convert PDF to an image format such as JPEG or TIFF

application layer will use the Database wrapper class to implement queries aimed at bringing data up to the presentation:

```
var model = new RegisterViewModel();
using (var db = new Database())
{
    var list = (from m in db.Customers select m).ToList();
    model.ExistingCustomers = list;
}
```

There's now a direct connection between the data source and presentation. You're really just reading and formatting data for display purposes. You expect authorization to be enforced at the gate via logins and UI constraints. If not, though, you can add more layers along the way and enable data exchange via IQueryable collections of data. The data model is the same as the database and is 1-to-1 with persistence. This model is sometimes referred to as layered expression trees (LET).

The power of CQRS
lies in the fact that you can
optimize command and query
pipelines at will, without the
risk that optimizing one might
break the other.

There are a couple of things you should note at this point. First, you're in the read pipeline where business rules typically don't exist. All you may have here are authorization rules and filters. These are well-known at the application-layer level. You have no need to deal with data transfer objects along the way. You have one persistence model and actual data containers for the view. In the application service, you end up with the following pattern:

```
var model = SpecificUseCaseViewModel();
model.SomeCollection = new Database()
    .SomeQueryableCollection
    .Where(m => SomeCondition1)
    .Where(m => SomeCondition2)
    .Where(m => SomeCondition3)
    .Select(m => new SpecificUseCaseDto
    {
        // Fill up
    })
    .ToList();
return model;
```

All data transfer objects you spot in the code snippet are specific to the presentation use case you're implementing. They're just what the user wants to see in the Razor view you're building and the classes are unavoidable. Also, you can replace all of the Where clauses with ad hoc IQueryable extension methods and turn the whole code into dialog written in domain-specific language.

The second thing to note about the query stack is related to persistence. In the simplest form of CQRS, the command and query stacks share the same database. This architecture makes CQRS similar to classic CRUD systems. This makes it easier to adopt when people are resistant to change. However, you can design the back end so the command and query stacks have their own databases

optimized for their specific purposes. Synchronizing the two databases then becomes another issue.

The Command Stack

In CQRS, the command stack is only concerned with performing tasks that modify the application state. The application layer receives requests from the presentation and orchestrates execution by pushing commands to the pipeline. The expression "pushing commands to the pipeline" is the origin of the various flavors of CQRS.

In the simplest case, pushing a command consists of simply invoking a transaction script. That triggers a plain workflow that accomplishes all the steps required by the task. Pushing a command from the application layer can be as simple as the following code:

```
public void Register(RegisterInputModel input)
{
    // Push a command through the stack
    using (var db = new CommandDbContext())
    {
        var c = new Customer {
            FirstName = input.FirstName,
            LastName = input.LastName };
        db.Customers.Add(c);
        db.SaveChanges();
    }
}
```

If needed, you can yield control to the true domain layer with services and domain model where you implement full business logic. However, using CQRS doesn't necessarily bind you to DDD and things such as aggregates, factories and value objects. You can have the benefits of command/query separation without the extra complexity of a domain model.

Beyond Regular CQRS

The power of CQRS lies in the fact that you can optimize command and query pipelines at will, without the risk that optimizing one might break the other. The most basic form of CQRS uses one, shared database and calls distinct libraries for reads and writes from the application layer.

More sophisticated forms may have multiple databases, polyglot persistence, data denormalization for query purposes, event sourcing and, more important, a more flexible way to hand commands to the back end. It's more flexible because using a bus to send commands and publish events lets you define and modify any task as it happens, in a similar fashion to managing a flowchart. At the same time, you can scale virtually up where you need to be by adding power and features to the bus component.

Many developers praise CQRS, but tend to limit applicability to collaborative and high-scale applications. CQRS isn't top-level architecture and is technology-agnostic. To a certain degree, CQRS is even agnostic of design patterns, but it's a pattern itself. It's simple and powerful and just right for common applications. ■

DINO ESPOSITO is the co-author of "Microsoft .NET: Architecting Applications for the Enterprise" (Microsoft Press, 2014) and "Programming ASP.NET MVC 5" (Microsoft Press, 2014). A technical evangelist for the Microsoft .NET Framework and Android platforms at JetBrains and a frequent speaker at industry events worldwide, Esposito shares his vision of software at software2cents.wordpress.com and on Twitter at twitter.com/despos.

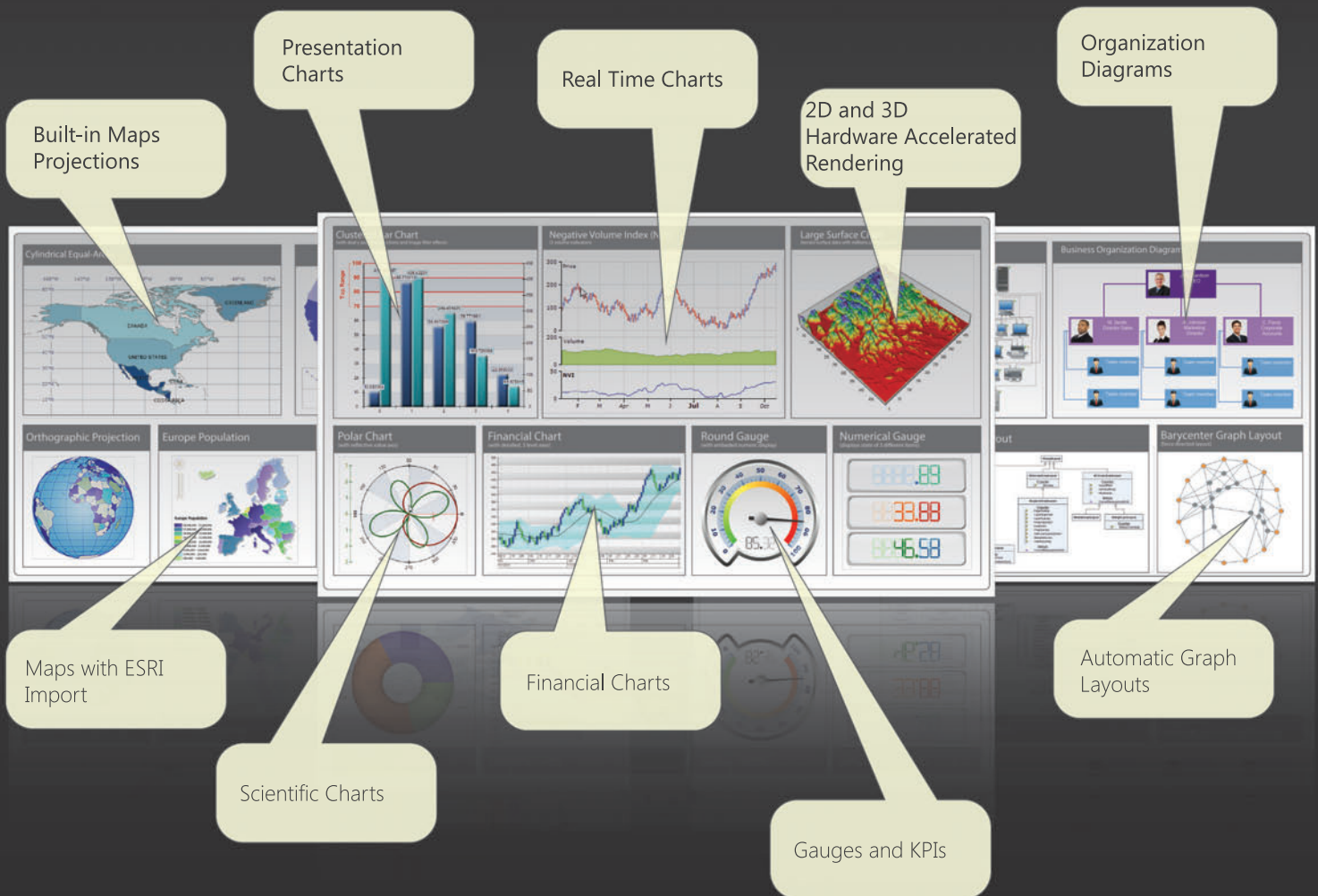
THANKS to the following technical expert for reviewing this article:

Jon Arne Saeteras

Nevron Data Visualization

The leading data visualization components for desktop and web development in a single package.

NEW RELEASE 2015.1



solutions available for








Learn more at www.nevron.com today

www.nevron.com | email@nevron.com | +1 888-201-6088 (Toll free, USA and Canada)

Microsoft, .NET, ASP.NET, SharePoint, SQL Server and Visual Studio are registered trademarks of Microsoft Corporation in the United States and/or other countries. Some Nevron components are only available for certain platforms. For details visit www.nevron.com or send an e-mail to support@nevron.com.



An Overview of Microsoft Azure DocumentDB

In November 2011, I wrote a column called “What the Heck Are Document Databases?” (msdn.microsoft.com/magazine/hh547103) in which I discussed some of the most well-known document databases: MongoDB, CouchDB and RavenDB. All three of these NoSQL databases are still going strong. At that time, Microsoft didn’t have a document database on the market, though it did have Microsoft Azure Table Storage, a NoSQL database based on key-value pairs. However, in August 2014, Microsoft Azure DocumentDB was announced, which, as its name implies, is a NoSQL document database service available on Azure.

In this column, I’ll provide an overview of Azure DocumentDB that I hope will intrigue you enough to investigate it further on your own. The service is available on Azure and was already in use even before its April 8 elevation from a preview to a generally available resource. For example, there’s a great customer story about a company called SGS that implemented DocumentDB as part of a solution at bit.ly/1GMnBd9. One of the developers on that project sent me a tweet about this and said the customer is really happy with it so far.

What Is a Document Database?

My earlier column focused on answering this question, but I’ll discuss it briefly here and recommend you read that other column. A document database stores data as documents, mostly as individual JSON documents. (MongoDB has a small twist because it squishes its JSON documents into a binary format called BSON.) This storage provides much faster performance when working with huge amounts of data because it doesn’t have to jump all over the database to pull together related data. Related data can be combined in a single JSON document. Another critical feature of a document database and other NoSQL databases is that they’re schema-less. Unlike a relational database whose tables need predefined schemas in order to store and retrieve data, a document database allows each document to define its own schema. So the database is made up of collections of documents. **Figure 1** shows a simple example of what an individual JSON document might look like. Notice that it specifies a property name along with the value and that it contains related data.

Not only is this data all in JSON format and self-describing, but it contains related data (ingredients). Another feature common to some of these document databases is that they’re all accessible via HTTP calls. You’ll see more about this later, and again, the earlier

column goes into detail about these and other features that are common to these databases.

Structure of Azure DocumentDB

Figure 1 shows what a typical document stored in a document database looks like. Azure DocumentDB is made up of more than just these documents, however. Documents are considered a *resource* in Azure DocumentDB, and may be grouped into *collections*, which are also resources in DocumentDB. You can create, update, delete and query collections using HTTP calls, just as you can with documents. In fact, DocumentDB is entirely made up of different types of resources. Collections are grouped into a single DocumentDB database. You can have multiple databases in a DocumentDB account and you can have multiple accounts.

All of these resources are first-class citizens in the ecosystem. In addition, there’s another set of resources that accompany your documents. These are named in a way that’s familiar to relational database users: Stored procedures, User-defined functions (UDFs), Indexes and Triggers.

One last resource related to documents is an attachment, which is any kind of binary that’s attached to the JSON document. The binary of the attachment lives in Azure Blob Storage but the metadata is stored in DocumentDB, ensuring you can query for attachments on a variety of properties.

Additionally, DocumentDB has built-in security features and, within that scope, Users and Permissions are also resources you can interact with using the same means as you would documents.

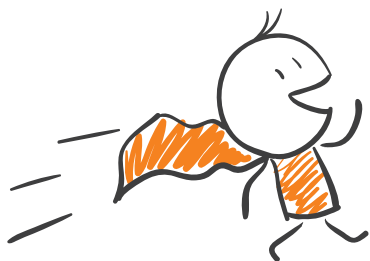
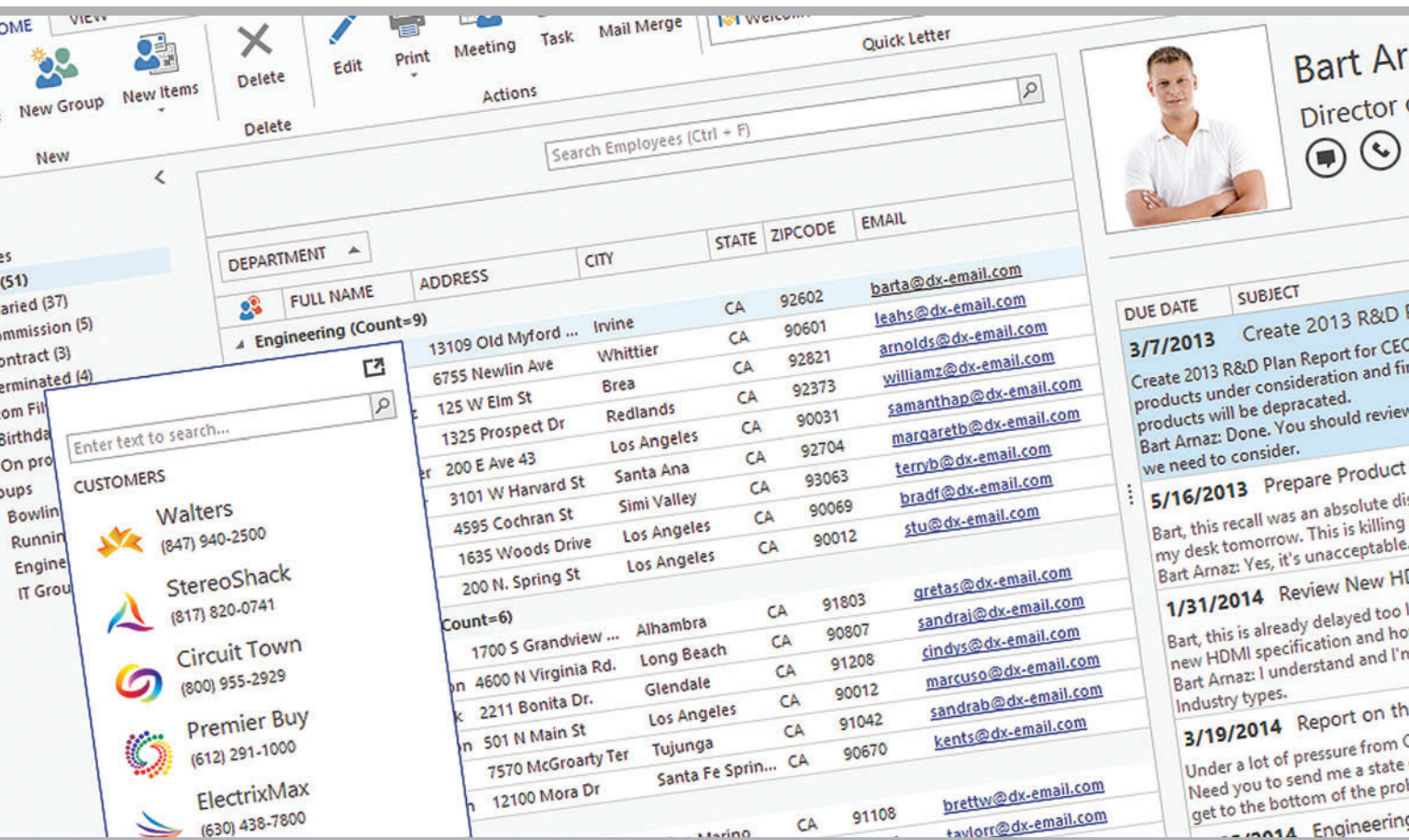
Figure 1 A Simple JSON Document

```
{
  "RecipeName": "Insane Ganache",
  "DerivedFrom": "Café Pasqual's Cookbook",
  "Comments": "Insanely rich. Estimate min 20 servings",
  "Ingredients": [
    {
      "Name": "Semi-Sweet Chocolate",
      "Amount": "1.5 lbs",
      "Note": "Use a bar, not bits. Ghiradelli FTW"
    },
    {
      "Name": "Heavy cream",
      "Amount": "2 cups"
    },
    {
      "Name": "Unsalted butter",
      "Amount": "2 tbs"
    }
  ],
  "Directions": "Combine chocolate, cream and butter in the top ..."
}
```

Code download available at msdn.microsoft.com/magazine/msdnmag0615.

Your Next Great App Starts Here

With DevExpress tools, you'll create high-performance, high-impact .NET solutions that fully replicate the look, feel and user-experience of **Microsoft Office®**



Unleash the **UI Superhero** in You

Download your **free 30-day trial** today
devexpress.com/try

Interacting with DocumentDB

There are a number of ways to work with resources in Azure DocumentDB, including: SQL, REST API and various client APIs including the .NET API, which lets you use LINQ to query the database. You can learn much more about the details of querying at bit.ly/1aLm4bC.

The Azure Portal is where you create and manage a DocumentDB account. (See the documentation at bit.ly/1Cq8zE7.) You can also manage your DocumentDB in the portal, as well as use Document Explorer and Query Explorer to view and query your documents. In Query Explorer, you can use the SQL syntax, as I've done for the simple query in **Figure 2**.

You can also use this SQL in your apps. For example, here's some code from "Build a Node.js Web Application Using DocumentDB" (bit.ly/1E7j5Wg), where the query is expressed in SQL syntax:

```
getOrCreateDatabase: function (client, databaseId, callback) {
    var querySpec = {
        query: 'SELECT * FROM root r WHERE r.id=@id',
        parameters: [{
            name: '@id',
            value: databaseId
        }]
    };
};
```

In these early days of DocumentDB, you might find this SQL syntax limited, but keep in mind that you can supplement the existing SQL with UDFs. For example, you can write your own CONTAINS function for building predicates that evaluate strings, such as CONTAINS(r.name, "Chocolate").

Like many other Azure resources, Azure DocumentDB has a native REST API and can be queried and updated using HTTP. Every resource has a unique URI. Here's an example of an HTTP request for a particular DocumentDB permission:

```
GET https://contosomarketing.documents.azure.com/dbs/ruJjAA==/users/ruJjAFjqQAA=/permissions/ruJjAFjqQABUp3QAAAAA== HTTP/1.1
x-ms-date: Sun, 17 Aug 2014 03:02:32 GMT
authorization: type%3dmaster%26ver%3d1.0%26sig%3dGfwrRDuhd18ZmKCJHW40CeNt5Av065QYFJxLaW8qLmg%3d
x-ms-version: 2014-08-21
Accept: application/json
Host: contosomarketing.documents.azure.com
```

Go to bit.ly/1NUId9 for details about working with the REST API directly. But working with any REST API can be pretty cumbersome. There are a number of client APIs already available for interacting with Azure DocumentDB: .NET, Node.js, JavaScript, Java and Python. Download the SDKs and read the documentation at bit.ly/1Cq9WVJ.

.NET developers will appreciate the .NET library allows you to query using LINQ. While the LINQ method support will definitely grow over time, the currently supported LINQ expressions are: Queryable.Where, Queryable.Select and Queryable.SelectMany.

Before you can perform any interaction with a DocumentDB, you need to specify an account, the database and the collection in which you want to work. The following, for example, defines a Microsoft.Azure.Documents.ClientDocument using the .NET API:

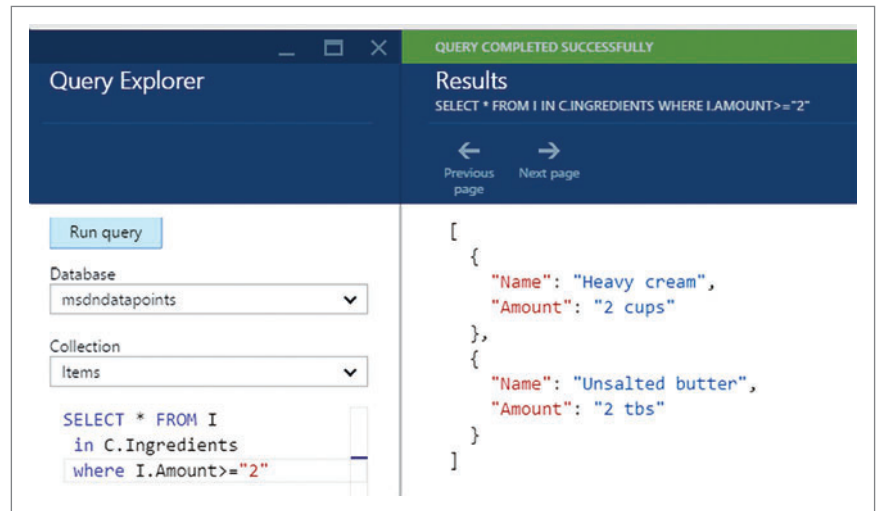


Figure 2 Using SQL Syntax to Query Documents in the Azure Portal Query Explorer

```
string endpoint = ConfigurationManager.AppSettings["endpoint"];
string authKey = ConfigurationManager.AppSettings["authKey"];
Uri endpointUri = new Uri(endpoint);
client = new DocumentClient(endpointUri, authKey);
```

This sample code comes from an ASP.NET MVC and DocumentDB walk-through I followed on the Azure Documentation page (bit.ly/1HS60Ee). The walk-through is quite thorough, beginning with steps to create a DocumentDB account on the Azure Portal. I highly recommend it or, alternatively, one of the walk-throughs that demonstrate DocumentDB with other languages, such as the Node.js article I mentioned earlier. The sample application has a single type, an Item class, shown in **Figure 3**.

Notice that each property of the item class specifies a JsonProperty PropertyName. This isn't required, but it allows the .NET client to map between the stored JSON data and my Item type, and lets me name my class properties however I want, regardless of how they're named in the database. Using the defined client, you can then express a LINQ query that returns an instance of a Microsoft.Azure.Documents.Database given a known database Id:

```
var db = Client.CreateDatabaseQuery()
    .Where(d => d.Id == myDatabaseId)
    .AsEnumerable()
    .FirstOrDefault();
```

From there you can define a collection within the database and finally query the collection with a LINQ expression like the following, which returns a single JSON document:

```
return Client.CreateDocumentQuery(Collection.DocumentsLink)
    .Where(d => d.Id == id)
    .AsEnumerable()
    .FirstOrDefault();
```

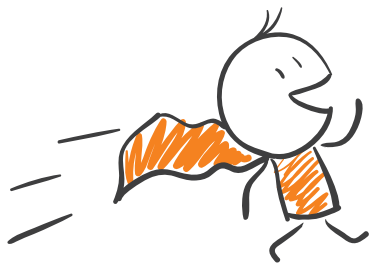
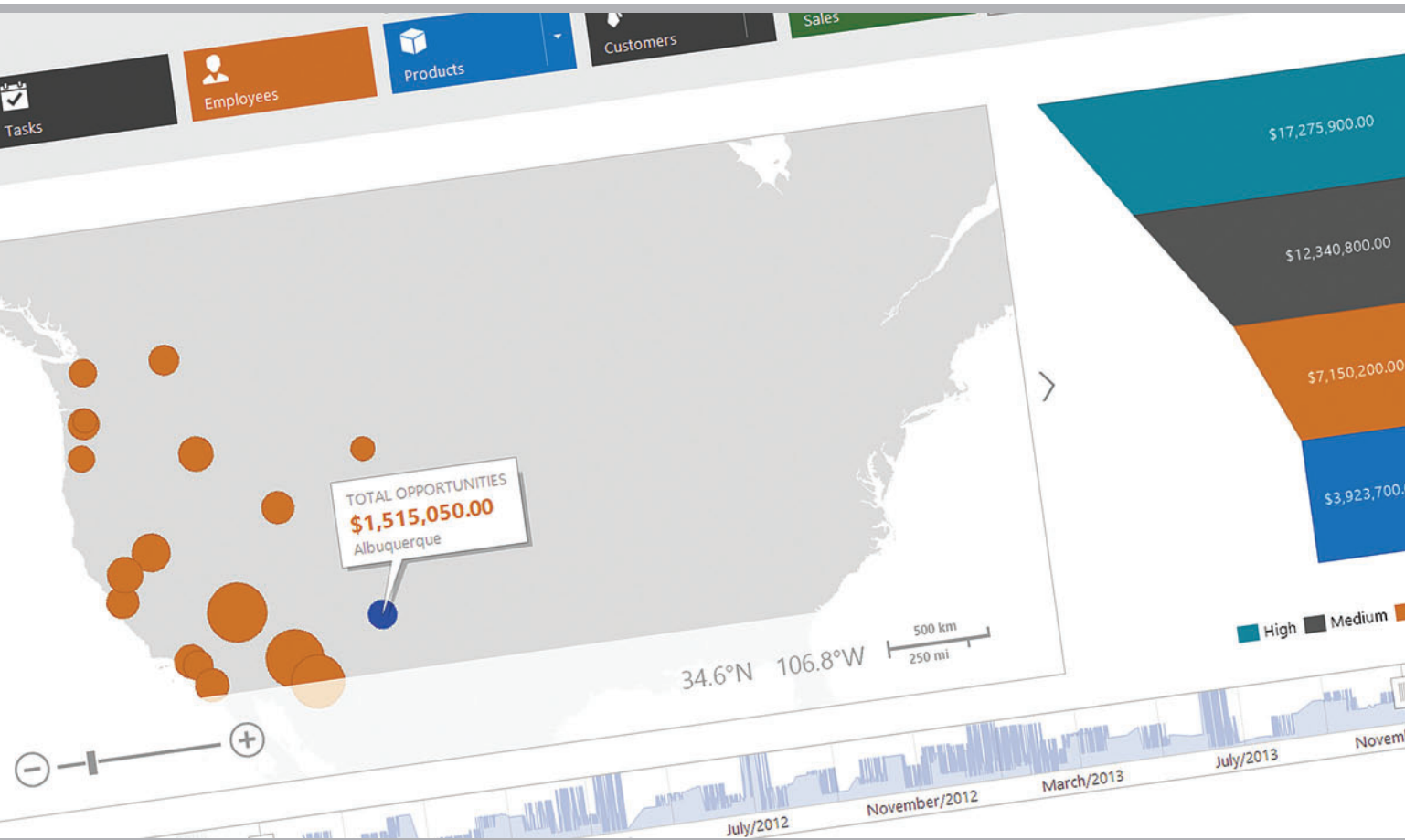
The various objects within the .NET API also enable operations to insert, update, and delete documents with the CreateDocumentAsync, UpdateDocumentAsync and DeleteDocumentAsync (CUD) methods, which wrap the HTTP calls in the REST API. Like the queries, there are relevant CUD methods for other resource types, such as stored procedures and attachments.

A New Twist on CAP

One of the more interesting aspects of DocumentDB that sets it apart from other document databases is that it lets you tune the consis-

Yeah, You Can Touch This

With DevExpress tools, you'll deliver elegant, **touch-first** apps that are built for today and ready for tomorrow



Unleash the **UI Superhero** in You

Download your **free 30-day trial** today
devexpress.com/try

Figure 3 The Item Class

```
public class Item
{
    [JsonProperty(PropertyName = "id")]
    public string Id { get; set; }

    [JsonProperty(PropertyName = "name")]
    public string Name { get; set; }

    [JsonProperty(PropertyName = "descrip")]
    public string Description { get; set; }

    [JsonProperty(PropertyName = "isComplete")]
    public bool Completed { get; set; }
}
```

cy. My earlier article on document databases talked about the CAP Theorem, which says that given guarantees of consistency, availability and partition (CAP) tolerance in a distributed system, only two of the three can be achieved. Relational databases ensure consistency at the cost of availability (for example, waiting for a transaction to complete). NoSQL databases, on the other hand, are more tolerant of eventual consistency, where the data might not be 100 percent current, in order to favor availability.

Azure DocumentDB provides a new way to address the CAP Theorem by letting you tune the level of consistency, thereby offering a chance to also benefit from both availability and partition tolerance at the same time. You can choose between four levels of consistency—strong, bounded staleness, session and eventual—which can be defined per operation, not just on the database. Rather than all or nothing consistency, you can tune the level of consistency to suit your needs throughout your solutions. Read more about this on the Azure DocumentDB Documentation page at bit.ly/1Cq9p3v.

Server-Side JavaScript

Many of you are probably familiar with stored procedures and UDFs in relational databases and, unlike other document databases, Azure DocumentDB includes these concepts, although they're written in JavaScript. JavaScript can natively interact with JSON, so this is extremely efficient for interacting with the JSON documents and other resources. No transformations or translations or mappings are needed. Another benefit of having server-side JavaScript in the form of stored procedures and UDF triggers is that you get atomic transactions across multiple documents—everything in the scope of the transaction will be rolled back if one process fails. Defining stored procedures and UDFs is quite different from what you might be used to in a relational database like SQL Server. The portal doesn't yet provide that capability. Instead, you define your server-side code in your client-side code. I recommend looking at the Server-Side Script section of the Azure DocumentDB .NET Code Samples at bit.ly/1FINK4y.

Now I'll show you how to create and store a stored procedure, then how to execute it. **Figure 4** shows a simple example that uses .NET API code to insert a stored procedure into a DocumentDB.

I've encapsulated all of the logic in a single method for simplicity. My `StoredProcedure` object consists of an ID and a Body. The Body is the server-side JavaScript. You might prefer to create JavaScript files for each procedure and read their contents when creating the `StoredProcedure` object. The code presumes that the `StoredProcedure` doesn't yet exist in the database. In the download example, you'll

see that I call out to a custom method that queries the database to ensure the procedure doesn't already exist before inserting it. Finally, I use `SetupDocDb<T>.Client` property (which provides the `DocumentClient` instance) to create the stored procedure, similar to how I queried for a document earlier.

Now that the stored procedure exists in the database, I can use it. This was a little difficult for me to wrap my head around because I'm used to the way SQL Server works and this is different. Even though I know the procedure's Id is "Hello," with the current API that's not enough to identify it when calling `ExecuteStoredProcedureAsync`. Every resource has a `SelfLink` created by the DocumentDB. A `SelfLink` is an immutable key that supports the REST capabilities of DocumentDB. It ensures every resource has an immutable HTTP address. I need that `SelfLink` to tell the database which stored procedure to execute. That means I must first query the database to find the stored procedure using the familiar Id ("Hello") so I can find its `SelfLink` value. This workflow is causing friction for developers and the DocumentDB team is changing how it works to eliminate any need for `SelfLinks`. That change may even be made by the time this article has gone to press. But for now, I'll query for the procedure as I would for any DocumentDB resource: I'll use the `CreateStoredProcedureQuery` method. Then, with the `SelfLink`, I can execute the procedure and get its results:

```
public static async Task<string> GetHello() {
    StoredProcedure sproc = Client.CreateStoredProcedureQuery(Collection.SelfLink)
        .Where(s => s.Id == "Hello")
        .AsEnumerable()
        .FirstOrDefault();

    var response =
        (await Client.ExecuteStoredProcedureAsync<dynamic>(sproc.SelfLink)).Response;
    return response.ToString();
}
```

Creating UDFs is similar. You define the UDF as JavaScript in a `UserDefinedFunction` object and insert it into the DocumentDB. Once it exists in the database, you can use that function in your queries. Initially, that was possible only using the SQL syntax as a parameter of the `CreateDocumentQuery` method, although the LINQ support was added just prior to the official release of DocumentDB in early April 2015. Here's an example of a SQL query using a custom UDF:

```
select r.name,udf.HelloUDF() AS descrip from root r where r.isComplete=false
```

The UDF simply spits out some text so it takes no parameters.

Notice that I'm using the `JsonProperty` names in the query because it will be processed on the server against the JSON data. With LINQ queries I'd use the property names of the `Item` type instead.

Figure 4 Inserting a Stored Procedure into a DocumentDB

```
public static async Task<StoredProcedure> InsertStoredProcedure() {
    var sproc = new StoredProcedure
    {
        Id = "Hello",
        Body = @"
function() {
    var context = getContext();
    var response = context.getResponse();
    response.setBody('Stored Procedure says: Hello World');
};"
    };

    sproc = await Client.CreateStoredProcedureAsync(setup.Collection.SelfLink, sproc);
    return sproc;
}
```


You'll find a similar query being used in the sample download, although there my UDF is called HelloUDF.

Performance and Scalability

There are so many factors that come into play when talking about performance and scalability. Even the design of your data models and partitions can impact both of these critical facets of any data store. I highly recommend reading the excellent guidance on modeling data in DocumentDB at bit.ly/1Chrfqa. That article addresses the pros and cons of graph design and relationships and how they affect the performance and scalability of DocumentDB. The author, Ryan CrawCour, who is the senior program manager of the DocumentDB team, explains which patterns benefit read performance and which benefit write performance. In fact, I found the guidance to be useful for model design in general, not just for Azure DocumentDB.

How you choose to partition your database should also be determined by your read and write needs. The article on partitioning data in DocumentDB at bit.ly/1y5T4FG gives more guidance on using DocumentDB collections to define partitions and how to define collections depending on how you'll need to access the data.

As another benefit of partitioning, you can create (or remove) more collections or databases as needed. DocumentDB scales elastically; that is, it will automatically comprehend the full collection of resources.

Indexes are another important factor affecting performance and DocumentDB allows you to set up indexing policies across collections. Without indexing, you'd only be able to use the SelfLinks and Ids of resources to perform querying, as I did earlier. The default indexing policy tries to find a balance between query performance and storage efficiency, but you can override it to get the balance you want. Indexes are also consistent, which means searches that leverage the indexing will have immediate access to new data. Read more details about indexing at bit.ly/1GMplDm.

Not Free, but Cost Effective

Managing performance and scalability affects more than the accessibility of your data, it also affects the cost of providing that data. As part of its Azure offerings, DocumentDB does come at a price. There are three price points determined by which of three performance-level units you choose. Because Microsoft is constantly tweaking the cost of its services, it's safest to point you directly to the DocumentDB pricing details page (bit.ly/1KUUMo). Like any NoSQL database, DocumentDB is aimed at providing data storage for huge amounts of data and, therefore, can be dramatically more cost-effective than working with relational data in the relevant scenarios. ■

JULIE LERMAN is a Microsoft MVP, .NET mentor and consultant who lives in the hills of Vermont. You can find her presenting on data access and other .NET topics at user groups and conferences around the world. She blogs at thedatafarm.com and is the author of "Programming Entity Framework" (2010), as well as a Code First edition (2011) and a DbContext edition (2012), all from O'Reilly Media. Follow her on Twitter at twitter.com/julielerman and see her Pluralsight courses at julieme/PS-Videos.

THANKS to the following Microsoft technical expert for reviewing this article:
Ryan CrawCour

NEW Entity Framework Profiler 3.0



Don't waste nights optimizing your database queries.

Don't let database dictate how your application works.



Find out how it can help you!

- OPTIMIZE QUERIES
- FIND BOTTLENECKS
- IMPROVE ENTITY FRAMEWORK PERFORMANCE
- SAVE COUNTLESS TIME

www.hibernatingrhinos.com

Proudly
Developed by



Implementing and Using Data Binding in Xamarin

Laurent Bugnion

Data binding is a popular concept in programming these days, especially for client applications. In this article, I'll discuss what data binding is and how it's used in technologies that support it natively, such as Windows XAML. After I cover the basic principles of data binding and observable properties, and how these can be used in technologies that don't support data binding "out of the box," I'll show how data binding support can be added to Xamarin.Android and Xamarin.iOS with the MVVM Light Toolkit (mvmllight.net). This lightweight toolkit is the most popular Model-View-ViewModel (MVVM) framework for platforms such as Windows Presentation Foundation (WPF), Silverlight, Windows Phone, Windows Store, Xamarin.Android, Xamarin.iOS, Xamarin.Forms and more.

I'll wrap up with a discussion of data binding in Xamarin.Forms, the newest addition to the Xamarin frameworks, which allows you to build the UI in XAML once, and to run it on Android, iOS and Windows Phone.

What Is Data Binding?

Data binding is the process of connecting the UI and the business logic. Through data binding, two connected properties will remain

synchronized. This is great because when you change a value on an object (the data model, often called ViewModel), the corresponding property in the UI will also be automatically updated. Similarly, in the case of "two-way" data binding, modifying a value in the UI will automatically update the ViewModel, as well. Data binding has been available for client applications for some time, even before WPF and all the other XAML-based technologies were introduced. For example, in ASP.NET Web Forms, data controls such as the listbox can be data-bound to a collection of items that need to be represented on the screen. In order to "translate" the data item into a visual representation, an item template is typically defined. This is a small snippet of HTML code in which some of the controls show the values contained in the data item's properties.

In XAML, the concept has been extended and it's reasonable to say that every XAML developer will use data binding to a greater or lesser extent. While it can be tempting to handle most of the data model changes in source code directly (especially for developers coming from more traditional technologies that don't support this mechanism), once the power of data binding has been mastered it's almost impossible to go back to the tedious process of writing the data change management in source code.

More Than Just Data Controls

XAML data binding is for much more than "just" data controls. It allows binding virtually any property of any object to any other property. (Keep in mind that only a special kind of property called a dependency property [DP] can be the target of data binding. DPs don't exist in Android and iOS, which is why the available data-binding frameworks use some workarounds.)

Data binding is seeing a revived interest in the Web world with the introduction of frameworks such as KnockoutJS and AngularJS, which feature data-binding extensions to the JavaScript code running in the background. These data-binding frameworks are heavily inspired by XAML frameworks and it's interesting to see a

This article discusses:

- What data binding is
- Implementing databinding in Xamarin
- Using the data-binding framework in MVVM Light
- Data binding in Xamarin.Forms

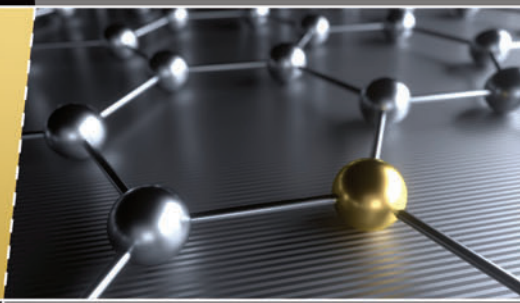
Technologies discussed:

XAML, Xamarin, MVVM Light Toolkit

Code download available at:

galasoft.ch/s/xamabinding

Switch to Amyuni PDF



Create & Edit PDFs in .Net - ActiveX - WinRT

- Edit, process and print PDF 1.7 documents
- Create, fill-out and annotate PDF forms
- Fast and lightweight 32- and 64-bit components for .Net and ActiveX/COM
- New WinRT Component enables publishing C#, C++CX or Javascript apps to Windows Store
- New Postscript/EPS to PDF conversion module



Complete Suite of Accurate PDF Components

- All your PDF processing, conversion and editing in a single package
- Combines Amyuni PDF Converter and PDF Creator for easy licensing, integration and deployment
- Includes our Microsoft WHQL certified PDF Converter printer driver
- Export PDF documents into other formats such as JPeg, PNG, XAML or HTML5
- Import and Export XPS files using any programming environment

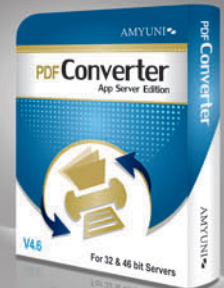


Advanced HTML to PDF & XAML

- Direct conversion of HTML files into PDF and XAML without the use of a web browser or a printer driver
- Easy Integration and deployment within developer's applications
- WebkitPDF is based on the Webkit Open Source library and Amyuni PDF Creator



High Performance PDF Printer For Desktops and Servers



- Our high-performance printer driver optimized for Web, Application and Print Servers. Print to PDF in a fraction of the time needed with other tools. WHQL tested for Windows 32 and 64-bit including Windows Server 2012 R2 and Windows 8.1
- Standard PDF features included with a number of unique features. Interface with any .Net or ActiveX programming language
- Easy licensing and deployment to fit system administrator's requirements



AMYUNI 

All development tools available at

www.amyuni.com

USA and Canada

Toll Free: 1866 926 9864
Support: 514 868 9227
sales@amyuni.com

Europe

UK: 0800-015-4682
Germany: 0800-183-0923
France: 0800-911-248

large crowd adopting the principles (such as the MVVM pattern) that have been developed by the XAML communities.

A Declarative Process

Markup technologies like XAML and HTML are called declarative because they allow you to “declare” your intent for the UI. This is typically done at a different point in time than the code implementation, and sometimes even by a different team. For example, in XAML technologies the UI developers responsible for the XAML are called “integrators” or sometimes “interaction developers,” or even the barbaric “devsigner.” This role implies a good understanding of the designer’s vision for the application, and even though integrators are often developers, they sit between two chairs and have a strong creative responsibility in the application development.

Thanks to data binding, it’s possible even on small development teams to strongly differentiate the process of UI creation. When I work on small apps where I’m the sole developer, I like to say I switch hats during the development process. When I work on code such as the data model, data services and so on, I wear my developer hat. When I switch to creating the UI, I then change to my integrator hat.

This separation of roles reflects the separation of concerns that’s often desirable in modern software development. While data binding doesn’t absolutely demand this separation, it’s a very convenient approach that helps you get into the right mindset.

Is There Data Binding in Xamarin?

Just like in Windows Phone, both Xamarin.Android and Xamarin.iOS allow the UI to be built in source code directly, but also offer a more modern approach using an XML file with declarative markup. In Android, this is an AXML file; in iOS a convenient approach is to use the Storyboard files. This is similar to what we do in Windows Phone with XAML. Note that these declarative markup files are not specific to Xamarin; they’re also used in vanilla Android and iOS development.

While convenient (notably because they allow working with a visual designer tool like the ones provided in Xamarin Studio or in Visual Studio), the declarative markup files in Android and iOS

are less powerful than XAML. In particular, they lack what XAML calls markup extension. Such expressions are enclosed within curly brackets ({}). There are a number of these extensions, for example, to access resources located in a XAML document or to create a data binding, as shown in this code snippet:

```
<Button Content="{Binding ButtonContent}"
        Command="{Binding ExecuteCommand}"
        CommandParameter="{Binding IsChecked, ElementName=LockCheckBox}" />
```

This markup, which is part of a View, says that a Button control has its Content property bound to a source property called ButtonContent, and its Command property is bound to a source property called ExecuteCommand. Those source properties are found within whatever object is serving as the default data or binding context for the Button’s parents (often the page as a whole), which is typically the ViewModel object. However, you can set the binding context at any level you need. The third line of markup, for example, explicitly says that the Button CommandParameter is bound to the IsChecked property of another element on the page whose name is LockCheckBox.

Unfortunately, the concept of data binding doesn’t exist natively in Android or iOS so, in the rest of this article, I’ll show you how you can implement data binding for these two platforms in Xamarin, and then how the MVVM Light Toolkit provides an implementation you can use to facilitate the process. Moreover, Xamarin also provides an extension called Xamarin.Forms that uses a XAML markup language similar to Windows for the UI. Xamarin.Forms supports binding declaration in the markup directly, without adding any external framework.

Implementing Data Binding in Android and iOS

I’m going to build a small application that connects to a simple data service and returns a list of flowers, each with a name and description. You’ll find this application, named XamDataBinding, at galasoft.ch/s/xambinding. To keep things simple, I’ll concentrate on one page only, shown in **Figure 1**. This page, implemented in Android, iOS and Windows Phone, has the following features:

- A list of flowers. Ideally the list should update itself whenever the underlying data source changes.
- A button for refreshing the list.

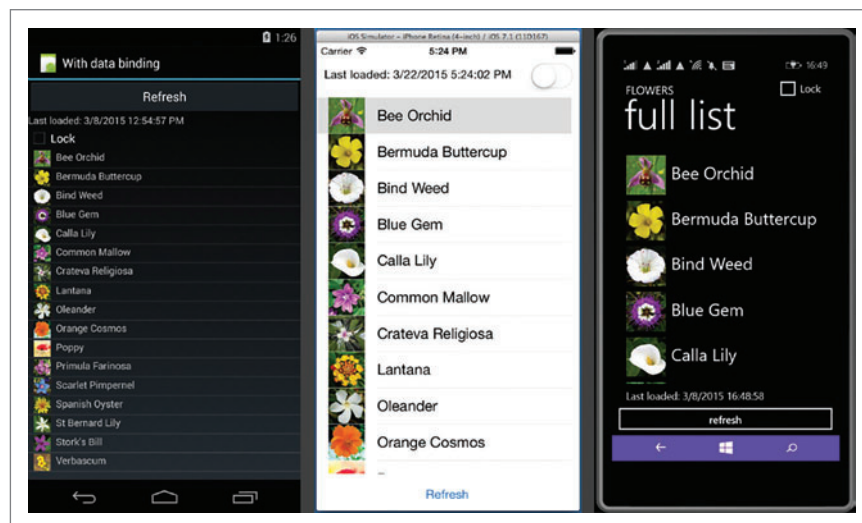


Figure 1 Sample Application in Android, iOS and Windows Phone

- Text showing the date and time of the last refresh.
 - A “lock” checkbox or switch. The purpose of this element doesn’t quite make sense in a production scenario but for the demo, imagine that this “locks” the Refresh button when it’s checked.
- The MVVM pattern relies on a few simple principles:
- Observable objects implement the `INotifyPropertyChanged` interface. This .NET interface exposes the `PropertyChanged` event, which should be raised when an observable property is modified.
 - Lists that can change during the course of an application should be stored in `ObservableCollection<T>` instances.

Such a list raises the `CollectionChanged` event when its content changes (addition, removal or a different sorting order).

- Some objects expose functionality in a property implementing the `ICommand` interface. This interface defines an `Execute` method and a `CanExecute` method. The latter returns true or false, depending on whether the `Execute` method can be executed. In addition, `ICommand` specifies a `CanExecuteChanged` event. This event must be raised when the value returned by the `CanExecute` method changes, as this potentially affects the state of bound UI controls.

In the MVVM Light Toolkit, these interfaces are implemented by concrete classes such as the `ObservableObject` and `ViewModelBase` (`INotifyPropertyChanged`), and the `RelayCommand` (`ICommand`). In the `XamDataBinding` example, the `Model` and `ViewModel` layers are built with the MVVM Light Toolkit, and stored in a portable class library called `XamDataBinding.Data` . This library can be shared between Xamarin and Windows.

In MVVM, each `View` is “driven” by a `ViewModel` . In Windows and Windows Phone, the `ViewModel` is typically used as the `DataContext` , which is a property of the `View` (`Page` , `Window` , `UserControl` and so forth). This is a convenient way to use data binding without having to always specify the source of the binding. In Android and iOS, there’s no concept of `DataContext` . Instead, you declare the `ViewModel` in the `View` and use it directly. In a small application like the `XamDataBinding` sample, the `ViewModel` can be created directly in the `View` (the `Activity` in Android, the `MainViewController` in iOS and the `MainPage` in Windows Phone):

```
private MainViewModel _vm;

public MainViewModel Vm
{
    get
    {
        return _vm ?? (_vm = new MainViewModel());
    }
}
```

Implementing Binding and Commanding Without a Framework

To avoid repeating the same thing over and over again, I’ll concentrate here on the Android application of the `XamDataBinding` sample. However, the exact same concepts apply to the iOS application. Because neither of these platforms supports data binding, I’ll implement binding and commanding manually:

- I’ll handle the `MainViewModel` ’s `PropertyChanged` event in the view, checking the name of the property that changed and updating the view accordingly.
- Some bindings go in two directions (two-way bindings). For example, when a checkbox is clicked, the corresponding Boolean property on the `MainViewModel` needs to be updated. Without a binding framework, I need to subscribe to the checkbox event and update the `ViewModel` ’s property manually from the `View` .
- The `Refresh` button must execute the `RefreshCommand` on the `MainViewModel` . Without a commanding framework, this, too, must be handled manually. I can handle the button’s `Click` event and call the command’s `Execute` method from the `View` directly.



dtSearch®

Instantly Search Terabytes of Text

25+ fielded and full-text search types

dtSearch’s **own document filters** support “Office,” PDF, HTML, XML, ZIP, emails (with nested attachments), and many other file types

Supports databases as well as static and dynamic websites

Highlights hits in all of the above

APIs (including 64-bit) for .NET, Java, C++, SQL, etc.

See
MS Azure
tutorial at
www.dtsearch.com/azure

“lightning fast” Redmond Magazine

“covers all data sources” eWeek

“results in less than a second” InfoWorld

hundreds more reviews and developer case studies at www.dtsearch.com

dtSearch products: Web with Spider

Desktop with Spider Engine for Win & .NET-SDK

Network with Spider Engine for Linux-SDK

Publish (portable media) Engine for Android-SDK **beta**

Document Filters – included with all products, and also available for separate licensing

Ask about fully-functional evaluations

The Smart Choice for Text Retrieval® since 1991

www.dtSearch.com 1-800-IT-FINDS

- Finally, I know that the command can disable the control to which it's bound. In this case, without a commanding framework, I need to subscribe to the command's CanExecuteChanged event. When this event fires, I can call the command's CanExecute method and, depending on the returned value, enable or disable the control.

None of these steps is very complex, but the whole process can be tedious and repetitive. In practice, it's easier to use a binding and commanding framework, such as the one offered by the MVVM Light Toolkit.

Using the MVVM Light Data-Binding Framework

The preceding examples work, but they're annoying to write because you have to handle the various events and think of all the possible scenarios. They are also more difficult to maintain and to understand, especially when there are many UI elements to handle.

The simple data-binding framework in the MVVM Light Toolkit, in contrast, allows you to create the relationship between the UI elements and the code, as well as the UI elements and commands. Thanks to this binding framework, I can create bindings and attach to commands in the OnCreate method, as shown in **Figure 2**.

In **Figure 2**, first I create a one-way binding between the MainViewModel's LastLoadedFormatted property and the Text property of TextView. This single line of code takes care of establishing the permanent relationship between these two properties. Note the use of the extension method SetBinding, which is provided by MVVM Light. This method returns the Binding instance that was created. Because the Binding object is written with WeakReferences to avoid memory leaks, this binding should be saved as a private field to prevent the binding from getting automatically detached.

The second statement creates the two-way binding between the IsLocked property of the MainViewModel and the Checked property of the LockCheckBox. Note that, by default, every binding is one-way, so you must specify the TwoWay BindingMode in the SetBinding method.

The third statement creates a special kind of binding, with only a source and no target. The source here is the Checked property of the LockCheckBox. I'll use this binding in the SetCommand method, to instruct the MainViewModel that the LoadCommand should be reevaluated every time the LockCheckBox's Checked property changes. In XAML, I'd create a binding on the Button's

CommandParameter. Here, I use a similar construct by creating the binding first, and then passing this instance to the SetCommand method.

The fourth statement sets a command on the RefreshButton. Though I handle the Click event here, I could in fact actuate the command for any event of any UI element. The second parameter of the SetCommand method is the command that must be actuated. Finally, the third parameter is optional and is needed only for commands that require a parameter. As I mentioned, I use the binding I created earlier here.

Setting the List

Finally, there's a ListView in the UI. MVVM Light offers a very useful extension method called GetAdapter that can be used on any ObservableCollection or any IList. It returns an Android adapter that can be used as the ListView's Adapter property. Because the MainViewModel's Flowers property is an ObservableCollection, the UI will automatically be updated whenever the collection changes:

```
FlowersList.Adapter = Vm.Flowers.GetAdapter(GetFlowerAdapter);
```

The GetFlowerAdapter method, which is passed to the GetAdapter method, is used to get the row view that will represent one flower. In this example I keep it simple by using a built-in row view that displays the picture and the name of the flower, as **Figure 3** shows. It would be very easy, however, to create a custom row view with more details or a different layout.

If I run the application now, I'll see the UI shown in **Figure 1**. You can try the following features:

- Clicking the Lock checkbox disables the Refresh button. The disabling occurs through the RefreshCommand, and its CanExecute method. Check the MainViewModel's code to see how this is done. Note that the same effect will happen in XAML if you run the included Windows Phone application.
- When the Refresh button is enabled, clicking it loads the list of flowers asynchronously. Note the code that connects to the Web service is included in the MainViewModel, and is fully reusable on any platform supported by this portable class library.
- After the list is loaded, the status of TextView is updated with the "last loaded" information. This, too, is done through the binding I created.

Figure 2 Creating Bindings and Commands

```
_lastLoadedBinding = this.SetBinding(
    () => Vm.LastLoadedFormatted,
    () => LastLoadedText.Text);

_lockBinding = this.SetBinding(
    () => Vm.IsLocked,
    () => LockCheckBox.Checked,
    BindingMode.TwoWay);

_refreshCommandBinding = this.SetBinding(
    () => LockCheckBox.Checked);

RefreshButton.SetCommand("Click", Vm.RefreshCommand, _
refreshCommandBinding);

FlowersList.Adapter = Vm.Flowers.GetAdapter(GetFlowerAdapter);
```

Figure 3 Creating a Custom ListView Row

```
private View GetFlowerAdapter(
    int position,
    FlowerViewModel flower,
    View convertView)
{
    if (convertView == null)
    {
        convertView = LayoutInflater.Inflate(
            Android.Resource.Layout.ActivityListItem, null);
    }

    var text = convertView.FindViewById<TextView>(Android.Resource.Id.Text1);
    text.Text = flower.Model.Name;

    var image = convertView.FindViewById<ImageView>(Android.Resource.Id.Icon);
    image.SetImageBitmap(GetImageBitmapFromUrl(flower.ImageUri.AbsoluteUri));

    return convertView;
}
```

- Running the iOS or the Windows Phone version of the application creates the exact same results. In iOS, the bindings are created in the MainViewController's ViewDidLoad method. In Windows Phone, I created the bindings in MainPage.xaml.

Data Binding in Xamarin.Forms

Last year, Xamarin released the Forms framework, which allows sharing not only code, but also the UI among all supported platforms. This is ideal for building a UI quickly to test code; for example, during the prototyping phase. The Forms framework can also be used for production code, such as line-of-business applications that have to run on multiple platforms.

In contrast to Xamarin.Android and Xamarin.iOS, Xamarin.Forms supports data binding natively. Because it also supports XAML (although a different version than the one available in WPF, Windows Phone and Windows Store), you can write the data binding in the XAML markup directly. This means you can write something like this:

```
<Label Text="{Binding LastLoadedFormatted}"
      VerticalOptions="Center"
      HorizontalOptions="Center" />
```

The included sample can be simplified even more by using Xamarin.Forms and reusing the UI on each platform. Of course, for applications where extra care must be taken with the UX, Xamarin.Forms might not be the best solution. But for a whole range of apps, Forms will simplify UI development considerably, freeing you to work on other tasks, such as improving application performance and stability, adding features, and so forth.

Wrapping Up

MVVM Light bindings are different from XAML bindings in that they're created in code instead of markup. However, many hours were spent making sure that the API is clean. I take this occasion to thank my good friend Corrado Cavalli for his support on the binding implementation and for helping me create a syntax that's as close as possible to the XAML binding workflow.

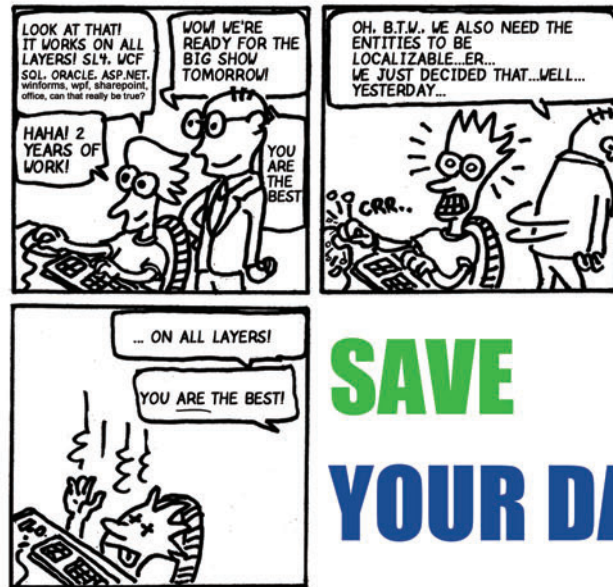
Also, for their help, I want to thank the Xamarin crew and especially James Montemagno who reviewed early versions of this binding framework and gave me super valuable advice.

Finally, for more information and a complete code sample, including Windows Phone, Xamarin.Android, Xamarin.iOS and Xamarin.Forms, see my Xamarin Evolve presentation at galasoft.ch/s/evolve14. ■

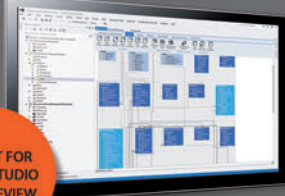
LAURENT BUGNION is senior director for IdentityMine Inc., a Microsoft partner working with technologies such as WPF, Silverlight, Xbox, Kinect, Windows Store, Windows Phone, Xamarin and UX. He's based in Zurich, Switzerland. He is also a Microsoft MVP, Microsoft Regional Director and Xamarin MVP.

THANKS to the following technical expert for reviewing this article:

Kraig Brockschmidt (Microsoft), Norm Estabrook (Microsoft) and James Montemagno (Xamarin)



**SAVE
YOUR DAY!**



**SUPPORT FOR
VISUAL STUDIO
2015 PREVIEW**

Use CodeFluent Entities


CodeFluent Entities is a unique product integrated into Visual Studio that allows you to generate database scripts, code (C#, VB), web services and UIs.

"CodeFluent Entities is a masterpiece software product envisioned and implemented by a group of very talented and experienced people. All that is achieved upon delivering high quality and standardized code and database layers, neatly stitched together and working in unison, lifting the burden from the developers to worry about this aspect on the development process, which could be called plumbing."

Renato Xavier - Colombaroli - Brazil

* Source : <http://visualstudiogallery.msdn.microsoft.com/B6299BBF-1EF1-436D-B618-66E8C16AB410>

To get a license worth \$399 for free
Go to www.softfluent.com/forms/msdn-2015



tools for developers, by developers

More information: www.softfluent.com Contact us: info@softfluent.com

MapReduce Without Hadoop Using the ASP.NET Pipeline

Doug Duerner and Yeon-Chang Wang

Have you ever wanted to add the power of MapReduce over Big Data to your smartphone apps or rich data analytics on your tablet or other small device, but thought it would be too difficult?

Have you ever wanted to transform your existing single-node application into a distributed system quickly and easily, without having to re-architect the entire application?

These questions are what prompted us to embark on an adventure to create an extremely easy to set up and use RESTful MapReduce component.

Products like Hadoop excel at the challenges of Big Data. We created a solution that sacrifices some of that functionality for

simplicity and agility in order to make it easier to develop Big Data applications. This way, you don't have to be an expert to get a working system up and running in a short time. The simplicity of the mesh versus the complexity of setting up Hadoop, and the agility of our solution versus the "elephantness" of a Hadoop cluster make it a compelling proposition.

In a nutshell, we created a very simple infrastructure that can use MapReduce to either do computationally intensive processing out on the "mesh" nodes or, alternatively, do data collection out on those nodes, with the results being correlated and aggregated into one final result that's returned to the client.

This article discusses:

- Using ASP.NET pipeline as a MapReduce pipeline
- Scalability benefits
- Using MapReduce to emulate a distributed computing pattern like "scatter-gather"
- "Chaining" MapReduce to emulate a distributed workflow engine
- Comparing the sample project with Hadoop

Technologies discussed:

IIS Web Server, ASP.NET Pipeline, HttpModule, Task Parallel Library, Dynamic Language Runtime, MapReduce

Code download available at:

msdn.microsoft.com/magazine/msdnmag0615

Background

The IIS Web Server (with its ASP.NET pipeline) has proven to be a highly scalable, enterprise-grade Web server. But these technologies aren't limited to simply serving up Web pages and hosting Web sites. There's really no technical reason you can't use them as a general-purpose pipeline mechanism accessed via HTTP. The ASP.NET pipeline steps execute in sequence (not moving to the next step until the previous step has completed), but each step can execute asynchronously in parallel. The IIS Web Server can be configured to run multiple ASP.NET pipelines (multiple w3wp.exe) servicing HTTP requests.

Using the ASP.NET pipeline as a general-purpose pipeline (that just happens to be accessed via HTTP), instead of serving up Web pages and hosting Web sites, might seem a bit unorthodox, but an

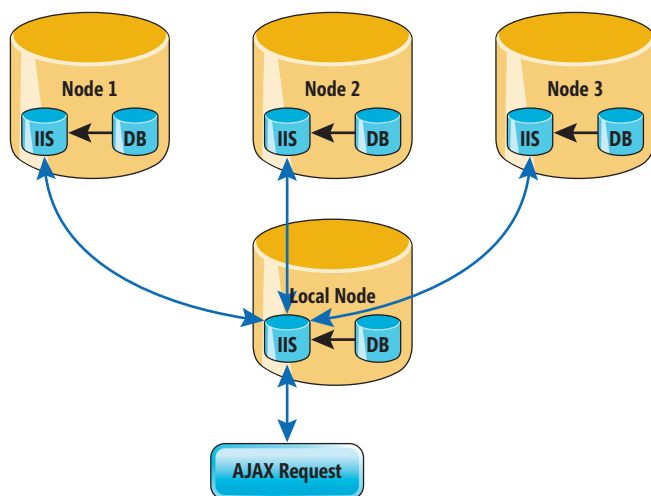


Figure 1 Emulate Distributed Computing Patterns Such as "Scatter-Gather"

ASP.NET pipeline (with asynchronous pipeline steps) is actually quite similar to CPU instruction pipelining in microprocessors (bit.ly/1Difv0), and the ability to have multiple w3wp.exe files (with an ASP.NET pipeline in each w3wp.exe) is quite similar to super-scalar design in microprocessors (bit.ly/1zMr6KD). These similarities, along with proven scalability, are what make using the IIS Web Server and ASP.NET pipeline for anything that needs pipelining functionality a compelling proposition.

There are lots of products that already do RESTful MapReduce (Hadoop, Infinispan, Riak, CouchDB, MongoDB and more), but our research suggests they're rather difficult to set up or require specialized expertise.

We wanted to simply use our existing Windows IIS servers that are already up and running; use our existing data API methods that have already been written; get the data for our UI screens on demand; and have the whole distributed MapReduce system up and running in minutes (all with limited knowledge of distributed systems or MapReduce system design and architecture). This way, you could quickly and easily transform an existing small-scale application into a larger distributed system with minimal effort or knowledge, on your own servers or on the cloud. Or, if you wanted to add rich data analytics to your existing smartphone app, you could do so with minimal effort.

This RESTful MapReduce component is a bolt-on that doesn't require the existing application to be rewritten, and it's a prime candidate when your goal is to merely add basic distributed functionality to an existing application that already has an extensive public data API. It's possible to quickly and easily emulate distributed computing patterns such as "scatter-gather," as shown in **Figure 1**.

The sample project that accompanies this article provides the starting point of a simple basic infrastructure that demonstrates this design philosophy and can be expanded going forward. The compelling factor of this design is not that it's better than the other products, but that it's easier.

It's simply an easy-to-use design alternative to the large enterprise MapReduce systems in use today. The design is by no means a replacement for the mature enterprise MapReduce products like Hadoop, and we're not implying that it's even close to containing all the functionality of the leading products.

MapReduce

In simple terms, MapReduce is a way of aggregating large stores of data. The Map step executes on many distributed processing server nodes. It usually executes a task on each distributed server node to retrieve data from the data nodes, and can optionally transform or pre-process the data while it's still on the distributed server node. The Reduce step executes on one or more final processing server nodes and consolidates all the results from the Map steps into one final result set using many different combining algorithms.

In simple terms, MapReduce is a way of aggregating large stores of data.

In the context of a business object API, the Map step executes a business object API method to get data, and the Reduce step combines all the Map step result sets into one final result set (doing a union by primary key, or an aggregation like a sum by group, for example) that's returned to the client that made the request.

One of the key benefits of MapReduce is it lets you "scale out" instead of "scale up." In other words, you simply keep adding more

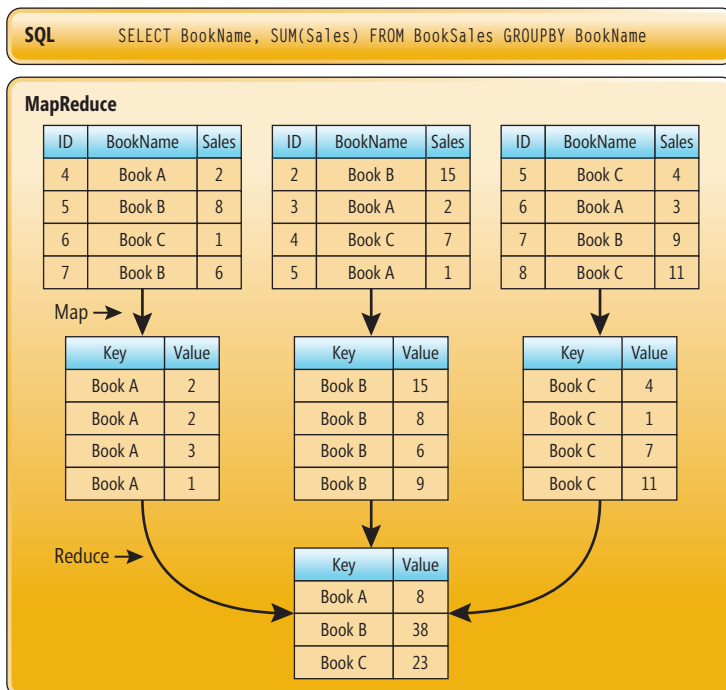


Figure 2 Simple SQL Relational Database Query Versus Same Query with MapReduce

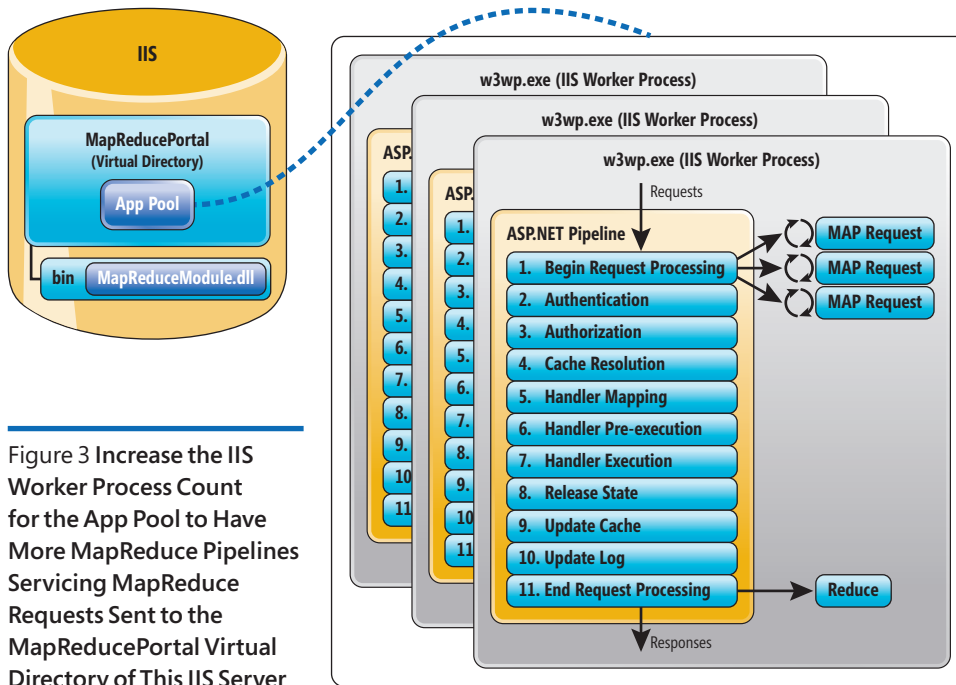


Figure 3 Increase the IIS Worker Process Count for the App Pool to Have More MapReduce Pipelines Servicing MapReduce Requests Sent to the MapReducePortal Virtual Directory of This IIS Server

normal server nodes rather than purchasing better hardware for the one main server node to scale. Scaling out is generally the cheaper, more flexible choice because it uses regular commodity hardware, while scaling up is typically much more expensive because the cost of the hardware tends to exponentially increase as it becomes more sophisticated.

As an interesting side note, MapReduce excels when it comes to extremely large volumes of data (Internet scale) and the data is partially structured or unstructured, like log files and binary blobs. In contrast, SQL relational databases excel when you have

normalized structured data with schemas, at least up to a certain limit when the overhead of the relational database is unable to deal with the huge amount of data.

Figure 2 shows a high-level overview of the MapReduce process and compares a simple SQL relational database query with the corresponding query in a Big Data MapReduce process.

REST

Representational State Transfer (REST) defines a public API running over HTTP that uses a create, read, update, delete (CRUD) paradigm, based respectively on the HTTP verbs Post, Get, Put and Delete, to return a representation of an object from the server to the client that made the request. REST seeks to allow public access to the

object itself as an entity, not just functional operations on the object. It isn't a specification or an RFC; it is simply a design recommendation. You can adhere closely to the pure REST design and require the URL be formatted to treat the object as an entity, like this:

```
http://server/MapReducePortal/BookSales/Book A
```

Or you can opt for a more RPC-style design and require the URL to be formatted with the class and method names to execute, as follows:

```
http://server/MapReducePortal/BookSales/GetTotalBookSalesByBookName/Book A
http://server/MapReducePortal/BookSales/GetTotalBookSalesByBookName?bookName=Book A
```

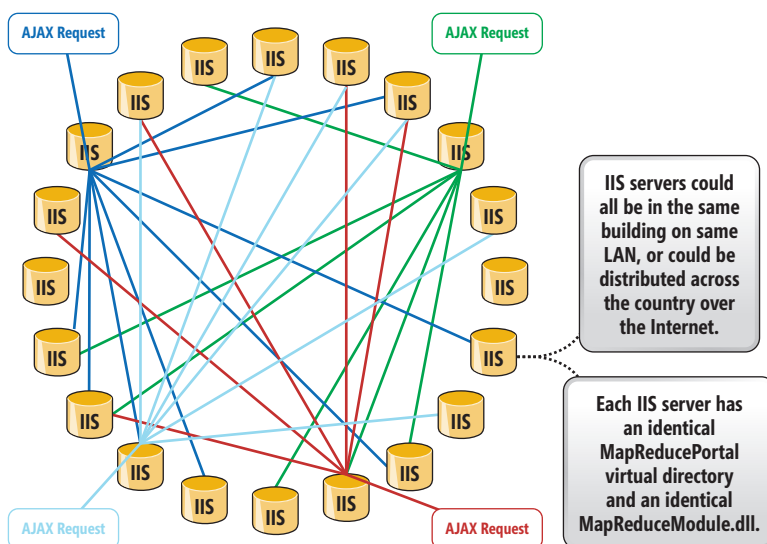


Figure 4 Any Server Node Can Initiate the MapReduce Request, and Any Number of Other Distributed Server Nodes Listed in the AJAX URL Can Execute the Map Step Parts of That Request in Parallel

RESTful MapReduce

RESTful MapReduce means doing MapReduce operations over HTTP for the API and for the transport mechanism among distributed server nodes.

REST over HTTP for the API and transport has several advantages that make it attractive:

- The HTTP protocol over port 80 is firewall-friendly.
- Client applications from almost any platform can easily consume resources without the need for platform-specific dependencies.
- The HTTP verbs (Get, Post, Put, Delete) are a simple, elegant paradigm for requesting resources.
- Gzip compression can assist in reducing payload sizes.
- The HTTP protocol itself has additional advantages, such as built-in caching.

Currently, the sample project uses REST over HTTP for the API and transport and supports only Get and Post, no write operations, and communicates entirely in JSON. It's similar to some of the well-known

DATA QUALITY 101

MASTERING THE FUNDAMENTALS IS THE KEY TO YOUR SUCCESS.

You can't turn Big Data into Big Value if it's messy. Garbage in, garbage out will always be a problem. Go back to the basics – cleaning all of your data is still the best, first step for success. Melissa Data provides the data quality tools and methods you need to correct, consolidate, and enrich contact data for improved data integration, business intelligence, and CRM initiatives. Always start with quality data – it's the easy way to get straight A's.

- **Verify addresses, phones & emails for over 240 countries**
- **Add geocodes and demographics for better insight**
- **Match duplicate records for a single view of the customer**
- **Identify change-of-address records before mailing**
- **On-premise and Cloud solutions**
- **Free trials with 120-day ROI guarantee**



ADDRESS
VERIFICATION



PHONE
VERIFICATION



NAME
VERIFICATION



EMAIL
VERIFICATION

BETTER DATA MEANS BETTER BUSINESS

MELISSA DATA®

www.MelissaData.com 1-800-MELISSA

methodologies for accessing the Hadoop Distributed File System (HDFS) externally (like Hadoop YARN and Hadoop WebHDFS), but supports only the absolute minimum necessary for the system to operate. We're not trying to replace Hadoop, or match all of its extensive functionality. We're merely trying to provide an extremely rudimentary, easy-to-use alternative, at the expense of functionality.

MapReduce Configuration

For the sample project, simply copy the MapReduceModule.dll into the \bin directory of the virtual directory on each IIS server node you want to use as a distributed server node in your MapReduce system, and then put an entry in the modules section of the web.config, like so:

```
<modules>
  <add name="MapReduceModule" type="MapReduce.MapReduceModule" />
</modules>
```

You're done. It's as easy as that.

One of the compelling factors of the sample project's design is the scalability gained by using the ASP.NET pipeline as a MapReduce pipeline to execute the MapReduce process.

If there's no virtual directory on the IIS server node, create a new virtual directory with a \bin directory, make it an Application and make sure it's using a Microsoft .NET Framework 4 Application Pool. Increase the w3wp.exe worker process count on the Application Pool that services the MapReducePortal virtual directory to provide more processing pipelines for MapReduce requests. The other advanced configuration options used for tuning the IIS Server have usually already been set by the IT department that manages the server and are beyond the scope of this article, but if that's not the case, they're readily available on the Microsoft Web site.

REST Configuration

For the sample project, simply place the PathInfoAttribute on any of your existing business object data API methods and specify the PathInfo string that will be used to map the URL to the method and method arguments. That's it.

One of the cool features of the sample code is whatever data types the existing business object data API methods are currently returning can stay the same and don't need to be changed. The infrastructure can handle pretty much any types automatically because it uses a .NET Dynamic-Object to dynamically represent the data types returned. For example, if the existing method returns a collection of Customer objects, then the DynamicObject represents a Customer data type.

The PathInfoAttribute PathInfo string uses the same .NET Uri-Template class that Windows Communication Foundation (WCF) uses and allows you to do all the same fancy things you can do in a WCF Web HTTP REST project or an ASP.NET Web API 2 project, such as argument variable name substitution, wild cards, and so forth. You choose what URL maps to what methods. You have total control and are free to implement your REST API any way you like. You can stick closer to a pure REST API and make your URL segments represent your objects like first-class entities:

http://server/MapReducePortal/BookSales/Book A

```
[PathInfoAttribute(PathInfo="/BookSales/{bookName}", ReturnItemType="Book")]
public BookSales GetTotalBookSalesByBookName(string bookName)
{
}
```

Or, if you prefer, you can loosely follow REST and make your URL segments specify the class name and method name you want to execute in the URL segments:

http://server/MapReducePortal/BookSales/GetTotalBookSalesByBookName/Book A

```
[PathInfoAttribute(PathInfo="/BookSales/GetTotalBookSalesByBookName/{bookName}",
  ReturnItemType="Book")]
public BookSales GetTotalBookSalesByBookName(string bookName)
{
}
```

It's entirely up to you.

Compelling Factors

One of the compelling factors of the sample project's design is the scalability gained by using the ASP.NET pipeline as a MapReduce pipeline to execute the MapReduce process. Because the ASP.NET pipeline operates sequentially, it's suitable for performing both the Map and Reduce steps. And what's cool is that though the pipeline is sequential and will not move to the next step until the previous step has completed, each step can still be executed asynchronously. This allows the pipeline to continue to receive and process new MapReduce requests even while the pipeline is blocked waiting for the Map calls to return from the other distributed server nodes.

As **Figure 3** shows, each w3wp.exe houses one ASP.NET pipeline acting as a MapReduce pipeline. The w3wp.exe (IIS worker process) is managed by the application pool assigned to the MapReducePortal virtual directory. By default, the application pool has one w3wp.exe processing new incoming requests to the virtual directory, but can very easily be configured to have as many w3wp.exes as you like. This lets you have multiple MapReduce pipelines on a single standalone server node, all teaming up to process the incoming MapReduce requests to the MapReducePortal

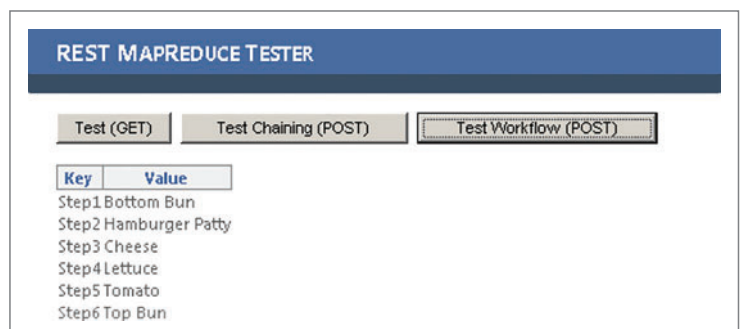


Figure 5 Displaying Output Resulting from Chaining in a Test Page



Boost ASP.NET Performance
Read our Whitepaper online

Extreme Performance Linear Scalability

For .NET & Java Apps

(Available on Microsoft Azure & AWS)

Cache data, reduce expensive database trips, and scale your apps to extreme transaction processing (XTP) with NCache.

In-Memory Distributed Cache

- Extremely fast & linearly scalable with 100% uptime
- Mirrored, Replicated, Partitioned, and Client Cache
- Entity Framework & NHibernate Second Level Cache

ASP.NET Optimization in Web Farms

- ASP.NET Session State storage
- ASP.NET View State cache
- ASP.NET Output Cache provider

Full Integration with Microsoft Visual Studio

- NuGet Package for NCache SDK
- Microsoft Certified for Windows Server 2012 R2



FREE Download

sales@alachisoft.com

US: +1 (925) 236 3830

www.alachisoft.com

virtual directory. The asynchronous nature of the individual ASP.NET pipeline allows many requests to be processed in parallel. The ability to have multiple w3wp.exe facilitating multiple ASP.NET pipelines takes you to the next level.

The sample project's design also lets you keep adding as many IIS servers as you like to form a larger and larger "mesh" of server nodes, as shown in **Figure 4**. The larger the mesh grows, potentially the larger the problem that can be handled by breaking it into smaller and smaller pieces to be solved and the greater the level of parallelism that can potentially be achieved. The asynchronous ASP.NET pipeline, combined with multiple pipelines per server, enable parallelism across a single server's CPU cores. The mesh of servers provides another level of parallelism across many server machines. It's a snap to add more IIS servers to the mesh; all you have to do is copy the MapReduceModule.dll to the \bin folder under the virtual directory and add an entry to the web.config file. Because the IIS servers are all simply standalone servers, no additional configuration is required. Products like Hadoop, in contrast, generally require more effort, planning and expertise because the servers must typically be configured as an actual server "cluster."

You don't even need specially built IIS servers. You can simply use any available IIS servers merely by copying the MapReduceModule.dll to any virtual directory that's already on the server. That's all it takes. The next AJAX call can now include the new IIS server in the distributednodes list parameter in the URL QueryString.

Another benefit of the server mesh design is that it doesn't rely on a Master node to function. In products like Hadoop, the Master node manages the server cluster and the location of the data across that server cluster. And it's the Master node that's been the source of failure when Hadoop was scaled to its limit in production, rather than the amount of data or the infrastructure.

In this server mesh design, there's no Master node. Any server node can initiate the MapReduce request, and the data lives on the node that collects it. As **Figure 4** shows, any server node can be the requester of data and the provider of data at the same time in parallel. A server can request data from any other nodes in the server mesh that are performing the Map function, and can receive the results, combining them into one final result set in the Reduce step. At the same time, that same server node can

also be acting as an edge server node handling a Map step, returning its partial results for a MapReduce request that originated from another server node and is going to be reduced back on that node.

Currently, the clients that are making the requests identify the location of the data (via the distributednodes list in the URL QueryString). Instead, you could modify the design to store this list (or just this node's nearest neighbor nodes or the nodes hosting data split across multiple nodes) in a database table on each individual node and programmatically add them to the URL at run time. In a sense, this would turn the single Master node notion into a distributed Master node concept, where each node knows where to get its data. It would be as if the Master node were spread across the mesh, allowing it to scale with the mesh.

Because this mesh design uses a range of tried-and-true Microsoft products—Windows Server, IIS Web Server and SQL Server databases—you get the robust redundancy and fault tolerance features (such as Network Load Balancing [NLB] on Windows Server for IIS, AlwaysOn Availability Groups with Automatic Page Repair or Mirroring with Automatic Page Repair for SQL Server) that are already built into these commercial products. Details for these features are readily available on Microsoft Web sites.

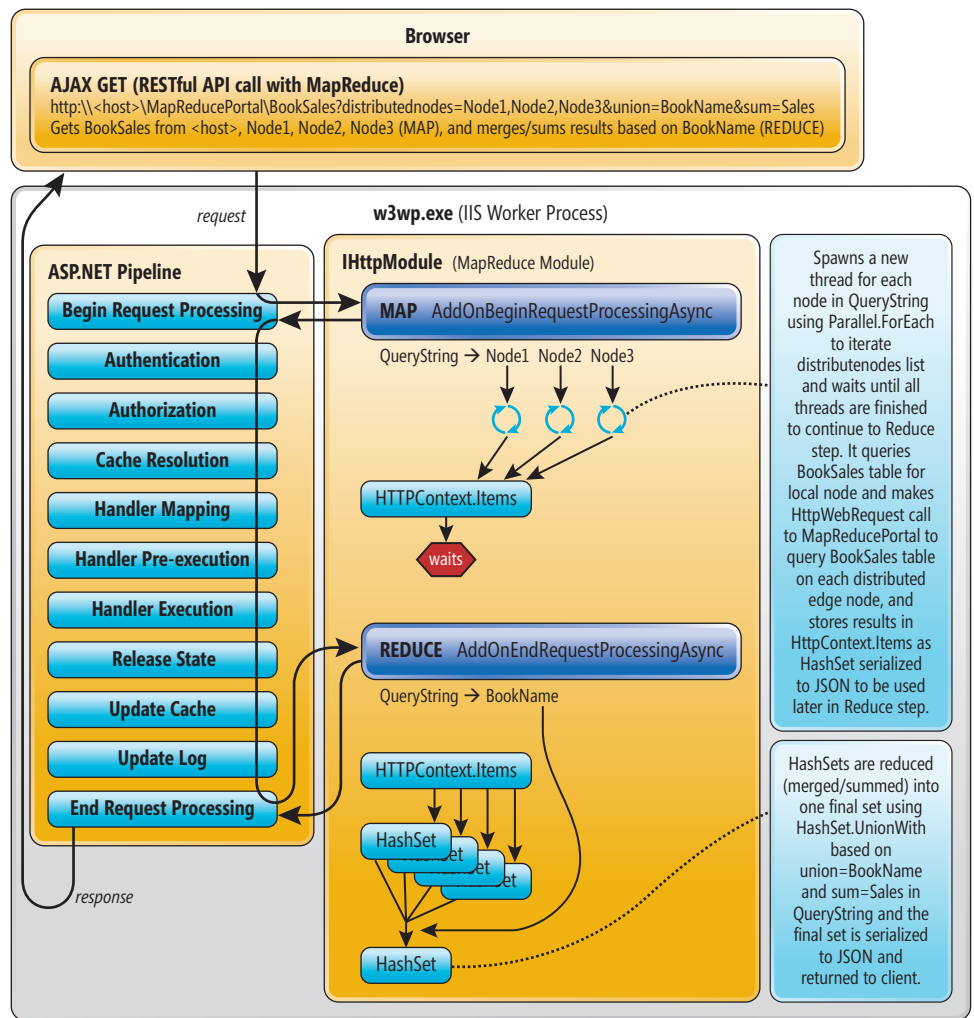
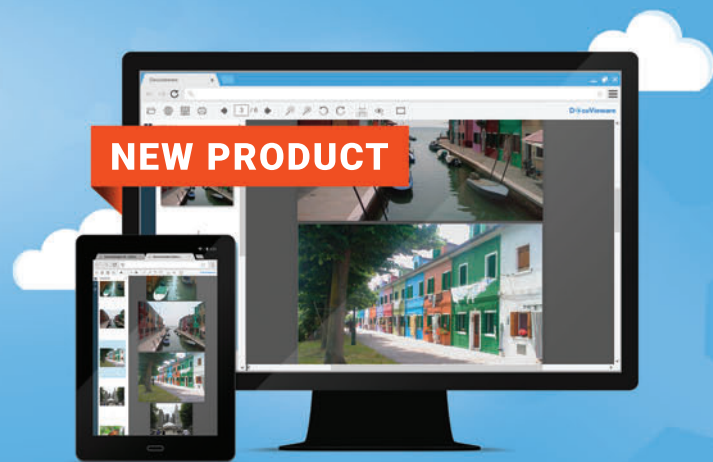


Figure 6 High-Level Design Overview of Processing Flow for MapReduceModule.dll



Universal HTML5 Viewer and Document Management Kit

View, annotate and manage any document, on any device, on any browser through a fully customizable zero-footprint HTML5 Control. Supports 90+ file formats, including PDF, TIFF and SVG.



Why DocuViewware?



Cross-platform, any browser, **zero-footprint** solution (no client-side install).



Super-**easy integration** within existing web applications.



Fast and **crystal-clear rendering** of documents and annotations.



Fully **customizable UI** look and feel, including convenient **snap-ins**.



Mobile devices optimization & **responsive** design.



Built-in **annotations**, **thumbnails**, **bookmarks** and **text search** tools.

Works in all modern browsers ...

IE9+, Chrome, Chrome for Android, Firefox, Firefox for Android, Opera, Safari 5+, Mobile Safari.



... so it works on all devices.

PC, Mac, tablets and smartphones.



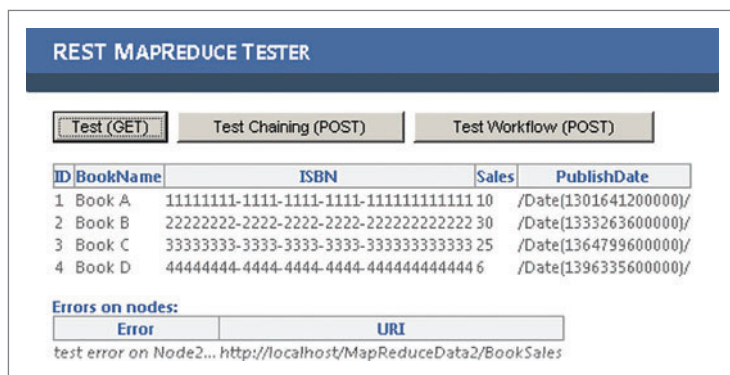


Figure 7 The Resulting Output in a Test Page

The sample project's design also allows multiple MapReduce requests to be "chained" together to form workflows where the starting input to one MapReduce request is the results from the previous MapReduce request. This is accomplished by changing the MapReduce request to a Post instead of a Get and including the previous MapReduce request results in the body of the Post request. Figure 5 shows an example of the resulting output in the test page.

Sample Project Overview

In essence, the MapReduceModule.dll transforms the ASP.NET pipeline into a MapReduce pipeline. It uses an HttpModule to implement both Map and Reduce functionality. As an interesting side note, some of the combining operations (like union) that are executed during the Reduce step rely on an IEqualityComparer<T>, where T is a DynamicObject that, in a sense, allows you to do the equality comparisons based on a property name as a string value at run time, even though IEqualityComparer<T> requires a concrete type to be defined at compile time. Pretty cool!

Figure 6 is a high-level overview of the design of the MapReduceModule.dll, showing the processing flow as it passes through the MapReduceModule.dll. MapReduceModule is the only dll required, and it needs to be on every server node that you want to participate in the MapReduce infrastructure. Adding the MapReduceModule.dll to the server is a snap and is accomplished by simply copying the MapReduceModule.dll to the \bin folder under the virtual directory and adding an entry to the web.config file.

In Figure 6, the IHttpModule uses the first step in the ASP.NET pipeline for the MAP functionality by subscribing to

the AddOnBeginRequestProcessingAsync event that's fired during the Begin Request Processing step in the ASP.NET pipeline. The IHttpModule uses the last step in the ASP.NET pipeline for the REDUCE functionality by subscribing to the AddOnEndRequestProcessingAsync event that's fired during the End Request Processing step in the ASP.NET pipeline.

In short, you subscribe only to the Begin Request Processing and End Request Processing events in the ASP.NET pipeline. They execute sequentially and don't move to the next step until the previous step has completed.

During the Begin Request Processing step, the IHttpModule initiates all the MAP requests by querying the local node and by sending an HTTP Web request to each of the

distributed server nodes present in the distributednodes list parameter in the URL QueryString. The HTTP Web request sent to each of the distributed server nodes uses the same URL that initiated this request, but with no distributednodes parameter in its URL.

In essence, the
MapReduceModule.dll
transforms the ASP.NET pipeline
into a MapReduce pipeline.

Out on the distributed server nodes that receive the MAP request, the same two ASP.NET pipeline steps are sequentially executed, but because there is no distributednodes parameter in its URL, the Begin Request Processing and the End Request Processing steps essentially query only that node. The MAP data retrieval method specified with the PathInfoAttribute is executed out on that edge distributed server node in order to get the local data from that node. The data that's returned in the response stream from each edge distributed server node to the server node that initiated the original request is then stored in the HttpContext using the URL as the key so it can be retrieved later during the final REDUCE step.

On the local server node that initiated the original request, the MAP data retrieval method specified with the PathInfoAttribute is executed in order to get the local data that's on the local server node that initiated the original request.

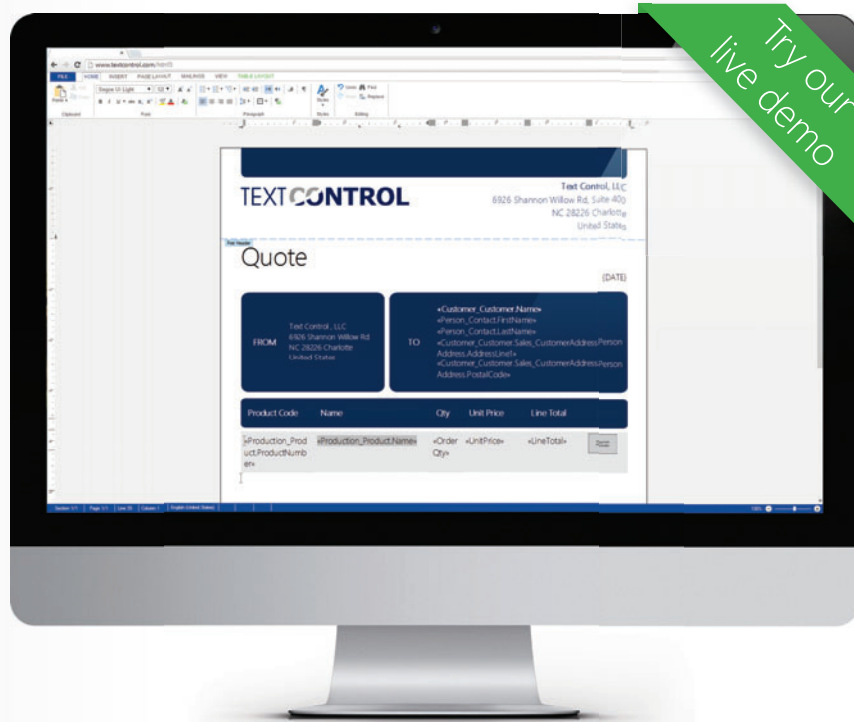
The data from the local server node is then stored in the HttpContext using the URL as the key so it can be retrieved in the final REDUCE step.

During the End Request Processing step, the IHttpModule executes the REDUCE step by looking in the HttpContext for all the data and the REDUCE parameters that were supplied in the URL QueryString (which can consist of predefined options like sum= and union=,

Figure 8 A Comparison of Basic MapReduce Functionality

Hadoop	Sample Project
Java MAP job function that counts words	Any method decorated with PathInfoAttribute is like a MAP job function
Java REDUCE job function that sums the word counts	Reduce parameters in the URL QueryString (such as sum=) is like a REDUCE job function that does sum operation
Writable interface (serialization)	[Serializable()] attribute (serialization)
WritableComparable interface (sorting)	IComparer<T> interface (sorting) IEqualityComparer<T> interface (sum,union)
The input to the MAP job is a set of <key,value> pairs, and the REDUCE job output is a set of <key,value> pairs	The arguments for the methods marked with PathInfoAttribute are like input to the MAP job, and the reduce parameters in the URL QueryString do reduce operation and serialize results to JSON like the REDUCE job output

Finally!



Cross-browser, true WYSIWYG document editing for ASP.NET



Edit and create
MS Word
documents



Create and modify
Adobe® PDF
documents

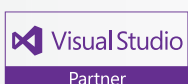


Create reports
and mail merge
templates



Integrate with
Microsoft®
Visual Studio

Try it online and download your 30-day trial version today:
www.textcontrol.com/html5



TEXTCONTROL

© 2015 Text Control GmbH. All rights reserved. All other product and brand names are trademarks and/or registered trademarks of their respective owners.

Figure 9 “Word Count Sample” Pseudocode Comparison

Hadoop MAP	Sample Project MAP
<pre>public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException { String line = value.toString(); StringTokenizer tokenizer = new StringTokenizer(line); while (tokenizer.hasMoreTokens()) { word.set(tokenizer.nextToken()); output.collect(word, one); } }</pre>	<pre>http://server/.../WordCount/Test. txt?distributednodes=Node1,Node2,Node3&union=Word&sum=Count [PathInfoAttribute(PathInfo="/WordCount/{fileName}", ReturnItemType="Row")] public HashSet<Row> GetWordCount(string fileName) { HashSet<Row> rows = new HashSet<Row>(); byte[] bytes = File.ReadAllBytes(fileName); string text = Encoding.ASCII.GetString(bytes); string[] words = text.Split(new char[] { ' ', '\r', '\n' }); foreach(string word in words) { dynamic row = new Row(); row["Word"] = word; row["Count"] = 1; } return rows; }</pre>
Hadoop REDUCE	Sample Project REDUCE
<pre>public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException { int sum = 0; while(values.hasNext()) { sum += values.next().get(); } output.collect(key, new IntWritable(sum)); }</pre>	<pre>http://server/.../WordCount/Test. txt?distributednodes=Node1,Node2,Node3&union=Word&sum=Count</pre>

sort=, or custom function options like reduce=CustomReduce-Function). Next, it merges/reduces all the data sets from all the nodes into one final result set using the specified REDUCE parameter. Finally, it serializes the final result set to JSON and returns that result set in the response stream to the client that initiated the original AJAX MapReduce request. If no REDUCE parameters are specified, then all the raw data from all the nodes is returned. **Figure 7** shows an example of the resulting output in the test page.

Comparing the Sample Project with Hadoop

Figure 8 compares basic MapReduce functionality in Hadoop and the sample project.

One common scenario in which MapReduce excels is counting the number of times a specific word appears in millions of documents. **Figure 9** shows a comparison of some basic pseudocode that implements the Big Data equivalent of the famous “Hello World” sample program—the “Word Count Sample.” The figure shows the Hadoop Java code implementation and the corresponding C# code that could be used to accomplish the equivalent in the sample project. Keep in mind this code is merely pseudocode and is by no means correct or complete. It’s shown merely to illustrate possible ways to accomplish similar functionality in the two designs. **Figure 10** shows the resulting output in the test page.

Figure 11 shows how to accomplish basic MapReduce functionality in the sample project. Notice how the object entity in the URL is mapped to the equivalent of the MAP step function with the PathInfoAttribute, and how the REDUCE parameter options in the URL QueryString, like sum= and reduce=, equate to the equivalent REDUCE step functionality in Hadoop.

Additional Examples

Figure 12 shows additional ways to accomplish MapReduce-type functionality and how the RESTful URL maps to the methods. The method’s implementation code is omitted for the sake of brevity. The code could be

implemented in many different ways, ranging from: an algorithm that counts words; to BookSales table data in the database at each book store in a chain of book stores; to a business object data API method that returns a collection of business object classes; to sensor data from distributed locations across the country. It’s all up to your imagination—have fun!

One common scenario in which MapReduce excels is counting the number of times a specific word appears in millions of documents.

Wrapping Up

In this article we presented a simple and basic infrastructure for MapReduce functionality that can be accessed RESTfully over HTTP

REST MAPREDUCE TESTER	
<div> <div>Test (GET)</div> <div>Test Chaining (POST)</div> <div>Test Workflow (POST)</div> </div>	
Word	Count
This	4
is	4
test	4
file	4
on	4
local	1
node	1
node2	1
node3	1
node1	1

Figure 10 The Resulting Output in a Test Page

Figure 11 Basic MapReduce Functionality in the Sample Project

(Like Hadoop MAP)	(Like Hadoop REDUCE)
<pre>http://server/.../BookSales?distributednodes=Node1,Node2,Node3&union=BookName&sum=Sales [PathInfoAttribute(PathInfo="/BookSales", ReturnItemType="Book")] public BookSales GetTotalBookSales() { }</pre>	
<pre>(Like Hadoop MAP) http://server/.../Alarms?distributednodes=Node1,Node2,Node3&reduce=UnionIfNotDeleted [PathInfoAttribute(PathInfo="/Alarms", ReturnItemType="Alarm")] public Alarms GetAlarms() { }</pre>	<pre>(Like Hadoop REDUCE) private static HashSet<Alarm> UnionIfNotDeleted(HashSet<Alarm> originalData, HashSet<Alarm> newData) { }</pre>

Figure 12 Miscellaneous Ways to Accomplish MapReduce with the Sample Project

<p>Example</p> <pre>http://server/.../BookSales/Book A?distributednodes=Node1,Node2,Node3&union=BookName&sum=Sales [PathInfoAttribute(PathInfo="/BookSales/{bookName}", ReturnItemType="Book")] public BookSales GetTotalBookSales(string bookName) { }</pre>
<p>Example</p> <pre>http://server/.../Alarms?distributednodes=Node1,Node2,Node3&union=AlarmID [PathInfoAttribute(PathInfo="/Alarms", ReturnItemType="Alarm")] public Alarms GetAlarms() { }</pre>
<p>Example</p> <pre>http://server/.../ Alarms?distributednodes=Node1,Node2,Node3&reduce=UnionIfNotDeleted [PathInfoAttribute(PathInfo="/Alarms", ReturnItemType="Alarm")] public Alarms GetAlarms() { }</pre> <pre>private static HashSet<Alarm> UnionIfNotDeleted(HashSet<Alarm> originalData, HashSet<Alarm> newData) { }</pre>
<p>Example</p> <pre>http://server/.../ SensorMeasurements/2?distributednodes=Node1,Node2,Node3&union=SensorID [PathInfoAttribute(PathInfo="/SensorMeasurements/{sensorID}", ReturnItemType="SensorMeasurement")] public SensorMeasurements GetSensorMeasurements(int sensorID) { }</pre>
<p>Example</p> <pre>http://server/.../ MP3Songs?distributednodes=Node1,Node2,Node3&union=SongTitle [PathInfoAttribute(PathInfo="/MP3Songs", ReturnItemType=" MP3Song")] public MP3Songs GetMP3Songs() { }</pre>

and consumed on a small device such as a smartphone or tablet. We also touched on transforming a single node application into a basic distributed system.

There are lots of extensive MapReduce infrastructures that do pretty much everything under the sun, but the goal and focus of this article was to make a basic MapReduce mechanism that is extremely easy to set up, and is simple to use.

The simplicity of setting up and expanding our solution gives you the ability to test your idea small (on several laptops) and easily scale up big once your idea has been proven (on as many servers as you need.)

The sample project allows you to use your existing business object data API methods in the Map step by simply applying an attribute to the method that maps the URL path to that method. It also allows you to control the Reduce step by adding simple commands to

the URL QueryString, such as a combining operation (like union) on the data based on a primary key.

By applying the attribute to the data API methods in an existing business object and specifying a union command based on a primary key field in the URL, you get a simple mechanism that can transform parts of a single node application into a basic distributed system with very little effort, providing the ability to have a centralized global view of the entire distributed system in one place. For example, a business data object that normally only retrieves items on that single node can now retrieve items on multiple nodes, merged based on a primary key field in the item. Data for the local offices could all be correlated or aggregated on demand and viewed in one screen at the headquarters.

For small devices, the “heavy lifting” happens on the IIS servers in the mesh and not on the small device. Thus, for example, a smartphone app can enjoy the MapReduce paradigm by making one simple HTTP call, using minimal phone resources. ■

DOUG DUERNER is a senior software engineer with more than 15 years designing and implementing large-scale systems with Microsoft technologies. He has worked for several Fortune 500 banking institutions and for a commercial software company that designed and built the large-scale distributed network management system used by the Department of Defense's Defense Information Systems Agency (DISA) for its “Global Information Grid” and the Department of State (DoS). He is a geek at heart, focusing on all aspects, but enjoys the most complex and challenging technical hurdles, especially those that everyone says “can't be done.” Duerner can be reached at coding.innovation@gmail.com.

YEON-CHANG WANG is a senior software engineer with more than 15 years designing and implementing large-scale systems with Microsoft technologies. He, too, has worked for a Fortune 500 banking institution and for a commercial software company that designed and built the large-scale distributed network management system used by the Department of Defense's Defense Information Systems Agency (DISA) for its “Global Information Grid” and the Department of State (DoS). He also designed and implemented a large-scale Driver Certification System for one of the world's largest chip manufacturers. Wang has a master's degree in Computer Science. He eats complex problems for dinner and can be reached at yeon_wang@yahoo.com.

THANKS to the following Microsoft technical experts for reviewing this article: Mikael Sitruk and Mark Staveley

Building Responsive Web Sites with Bootstrap

Keith Pijanowski

It's only a matter of time before a customer taps your site's URL on their mobile device. Will the user see a rendering of your site that's impossible to use on a phone because the site was designed for desktop only? If so, that user will most likely move on to a more mobile-friendly site. What if that user could instead encounter a UX designed specifically for a mobile form factor, and enjoy easily interacting with your site?

In the past, a Web site that supported mobile and desktop required different code bases. Today, however, there are UI frameworks that allow a site's Web pages to support mobile, tablet, desktop and even large-screen desktops—all with a single code base. Though these tools aren't a panacea for mobile Web developers, they can significantly ease the development process. Still, thought must be given to determine if one is right for your site.

This article presents an overview of one of these tools, Bootstrap, which is included in many of the Visual Studio 2013 Web Application

templates. Bootstrap is a framework that enables responsive Web design (RWD)—an approach to designing Web sites that aims to provide an acceptable viewing experience across all form factors with a single code base. After the overview, I'll consider some factors that can help you decide if a framework like Bootstrap is right for your site.

The Beginning

It has taken a certain amount of evolution to bring the software industry to the point where a solution like Bootstrap emerges. After a fitful start, beginning in 1998, in which Wireless Markup Language (WML), Compact HTML, and XHTML Mobile Profile surfaced and then disappeared, HTML5 finally became the next-generation HTML standard for all devices in 2009. But standards alone aren't enough. While the standard for mobile Web markup was evolving, significant advances also occurred with hardware, mobile browsers and bandwidth. Processors became faster, mobile browsers advanced to the point where they could display any Web page (even if the page hadn't been designed for a mobile device), and bandwidth increased. This created an environment for developers to take over.

Developers love to make tools for other developers, and the mobile Web is no exception. There are more than a dozen different UI frameworks on the market today. It would be unfair to label one of these frameworks "the best" because there are different approaches and Web sites have different requirements. I chose Bootstrap for this article because it has a solid level of adoption, is fully documented, and is included in the ASP.NET Web Application templates.

This article discusses:

- Bootstrap and responsive Web design
- Trying out the Bootstrap grid system
- Bootstrap components and JQuery plug-ins
- Deciding if Bootstrap is right for your site

Technologies discussed:

Bootstrap, Visual Studio 2013, CSS, JavaScript

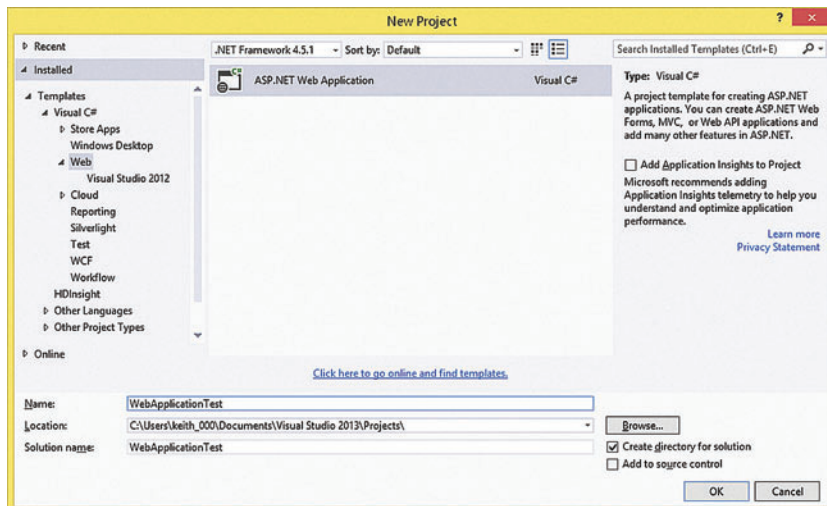


Figure 1 Visual Studio 2013 New Project Dialog for the ASP.NET Web Application Template

What Is Bootstrap?

Bootstrap is a responsive UI framework for building Web sites. The Bootstrap framework, like many other UI frameworks for Web development, is a collection of CSS classes, UI components and JQuery plug-ins. Bootstrap is considered a lightweight framework. In other words, it uses CSS more than JavaScript to do its work. Even though Bootstrap is considered lightweight, a Web page that uses Bootstrap tooling still requires more processing to render than a page written specifically for a given form factor. For this reason, performance should be an important consideration when designing, developing and testing pages that use any part of the Bootstrap framework.

If you prefer to use the latest and greatest versions of Bootstrap and JQuery, use NuGet to update your project.

Bootstrap does a lot for you automatically. It also allows you to easily customize the default behavior for a specific form factor while keeping your page looking good on other form factors. Version 3 of Bootstrap comes with CSS classes that specifically target browser widths commonly found on mobile browsers, tablet browsers, desktop browsers and even browsers running on large desktop screens. You can use the Bootstrap CSS classes to annotate HTML5 elements. The most commonly used CSS classes involve the Bootstrap grid system, a collection of classes that organize the layout of a page using rows and columns. I'll describe the grid system in more detail later in this article.

Bootstrap has a number of UI components for creating a site's UI, including Button dropdown, Button group, Dropdown, Navbar, Breadcrumb, Media object, Pagination and Progress bar, just to name

a few. Many of these components are responsive themselves, meaning that they render differently based on the width of the browser. For example, the Navbar is a powerful component that transitions automatically from a menu bar that displays menu options across the entire width of the screen on desktops to a nice-looking compact version that presents options via a dropdown menu bar that's activated when the user touches the Navbar.

For more advanced UI features, Bootstrap also has a collection of custom JQuery plug-ins, including Carousel, Collapsible panel, Modal dialog prompts and Popover messages.

Getting Started

The Bootstrap CSS and JavaScript files are automatically included in your project when you use the ASP.NET Web Application template for Visual Studio 2013 to create your project. The new project

dialog for Web Application templates is shown in **Figure 1**. Bootstrap also requires JQuery. When you get Bootstrap from the Visual Studio 2013 Web Application template, you get Bootstrap v3.0.0 with JQuery v1.10.2. If you use the templates for Visual Studio 2012, these files won't be included in your project. Also, Bootstrap is not included automatically if you create a new Web site using File | New Web Site.

If you prefer to use the latest and greatest versions of Bootstrap and JQuery, use NuGet to update your project. And use NuGet to get Bootstrap if you have a version of Visual Studio prior to Visual Studio 2013. The Bootstrap site (getbootstrap.com) shows how to link to a CDN-hosted version of Bootstrap, which can provide a significant performance advantage.

Experimenting with the Bootstrap Grid System

Bootstrap provides what's known as a 12-column responsive layout. You can lay out your pages using any number of rows, but you must use 12 columns. These 12 columns are sized equally across the width of the browser and you use the Bootstrap CSS classes

Figure 2 A Common Layout Pattern Using the Bootstrap Grid System

```
<body>
  <div class="container">
    <hr />

    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 gray">1</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 orange">2</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 gold">3</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 lightGreen">4</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 green">5</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 lightBlue">6</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 purple">7</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 gray">8</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 orange">9</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 gold">10</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 lightGreen">11</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 green">12</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 lightBlue">13</div>
    <div class="col-xs-6 col-sm-4 col-md-3 col-lg-2 purple">14</div>

  </div> <!-- /container -->
</body>
```

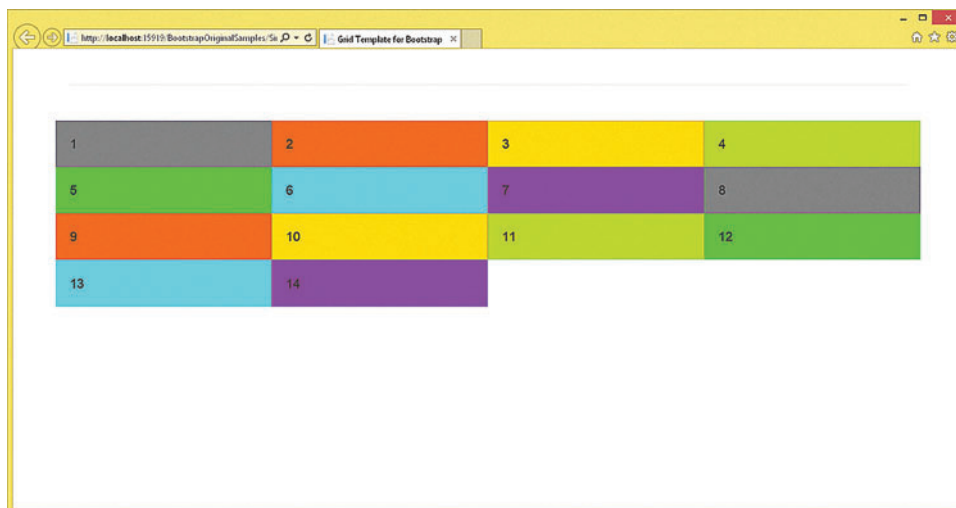


Figure 3 The Example from Figure 2 Rendered on a Desktop Using the col-md-3 Class

to specify how many columns an HTML element should occupy. This is a different type of grid system than XAML developers for Windows Phone and Windows 8 are accustomed to. The XAML grid system allows a page layout to have any number of rows and columns. Developers then place each control in a cell by specifying a row number and a column number. What makes the Bootstrap grid system especially powerful is that you can specify a different column span for large desktops, normal desktops, tablets and phones. For example, a common layout pattern is to create a series of divs and for each div specify that it should occupy 2 columns on a large desktop (this means that 6 divs would fit on each row), 3 columns on a normal desktop, 4 columns on a tablet and 6 columns on a phone (which would produce 2 divs per row). This scenario is coded in **Figure 2** using the col* classes that make up the Bootstrap grid system. The color classes (gray, orange and so on) merely color the background of a div. **Figure 3** and **Figure 4** show how this page would be rendered on a desktop and a phone, respectively.

It's important to note that Bootstrap doesn't actually detect the device type. Rather, Bootstrap uses media queries to determine whether a CSS class should be applied. The four categories of classes are shown in **Figure 5**, along with the width ranges that will cause each category to be applied.

You can use any combination of these categories when laying out a specific HTML element. For example, the following line of code uses one class from each category:

```
<div class="col-xs-6 col-sm-4 col-md-3 col-lg-2">
{Other HTML elements here.} </div>
```

This is equivalent to the following pseudo code:

If the browser's width is less than 768 pixels, then this div will span 6 of 12 available columns, which takes up one-half (6/12) of the screen.

Else if the browser's width is between 768 pixels and 991 pixels, then this div will span 4 of 12 available columns, taking up one-third (4/12) of the screen.

just need to specify a class for phone and tablet. like so:

```
<div class="col-xs-6 col-sm-4"> {HTML elements here.} </div>
```

Bootstrap contains more than 20 components, which can be used in their current state or easily extended.

Bootstrap will use the tablet setting for both the desktop and large desktop form factors. Also, the default class for phone is col-xs-12, which places each div on its own row. If this is an acceptable layout for a phone, the preceding line of code can be further reduced:

```
<div class="col-sm-4"> {HTML elements here.} </div>
```

Notice in **Figure 2** that rows aren't specified. The Bootstrap grid system will automatically move a div to the next row if the current row has already been used up or if there's not enough room for the div. You can force a new row by using the row class like this:

```
<div class="row">
{Place columns here.}
</div>
```

Why 12 columns? 12 may sound like a randomly chosen number for this grid system, but it's not. The number 12 provides a lot of divisibility and a lot of room for composability. For example, the numbers 1, 2, 3, 4, 6 and 12 all divide evenly into 12. Using these values for column spans can produce a content layout of 12 columns, 6 columns, 4 columns, 2 columns and 1 column, respectively. Moreover, combinations such as 3+9, 9+3, 4+8 and 8+4 are visually appealing combinations for laying out content.

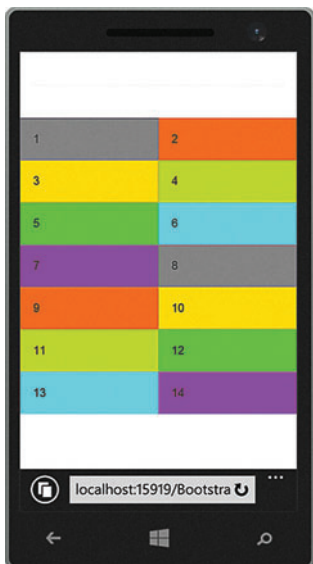


Figure 4 The Example from Figure 2 Rendered on a Phone Using the col-xs-6 Class

PRECISELY PROGRAMMED FOR SPEED

DynamicPDF—Comprehensive PDF Solutions for .NET Developers

ceTe Software's DynamicPDF products provide real-time PDF generation, manipulation, conversion, printing, viewing, and much more. Providing the best of both worlds, the object models are extremely flexible but still supply the rich features you need as a developer. Reliable and efficient, the high-performance software is easy to learn and use. If you do encounter a question with any of our components, simply contact ceTe Software's readily available, industry-leading support team.



DynamicPDF

[WWW.DYNAMICPDF.COM](http://www.DynamicPDF.com)



TRY OUR PDF SOLUTIONS FREE TODAY!

www.DynamicPDF.com/eval or call 800.631.5006 | +1 410.772.8620

ceTe software

Figure 5 Bootstrap Grid System Class Categories

Category Prefix	Form Factor	Width in Pixels
col-xs-*	Phone	Less than 768
col-sm-*	Tablet	768 to 991
col-md-*	Desktop	992 to 1,200
col-lg-*	Large Desktop	Greater than 1,200

Bootstrap Components

Bootstrap contains more than 20 components, which can be used in their current state or easily extended. Check out the Bootstrap site for a description of each component and a usage example. The documentation for the media object even makes use of an REO Speedwagon music video to demonstrate its use. Even if you didn't grow up during the 80s, consider giving it a listen.

I'll take a closer look at the Navbar component, which is used to create a menu bar at the top of a page. **Figure 6** shows how to use the Navbar component to create a simple menu that contains a site's brand and a few menu options. When a browser's width is wider than 768 pixels, the Navbar is rendered as shown in **Figure 7**. If the browser width is smaller than 768 pixels, the Navbar is rendered as shown in **Figure 8**. In this case, the menu options are displayed when the user taps the button containing the horizontal lines.

Obviously, this is a very simple example. The Navbar can also contain dropdown menu buttons, and you can even incorporate search functionality in the Navbar. Also, the Brand portion of the Navbar, which is text in the example, can be an image.

JQuery Plug-Ins

Bootstrap has about a dozen JQuery plug-ins, which are documented under the JavaScript tab at the Bootstrap site. At first glance they look similar in function to the components described in the previous section in that they're CSS classes that are used to augment HTML5 elements. However, these CSS classes trigger JQuery activity when the HTML element is rendered or when various events fire off the HTML element. Pages that make heavy use of these plug-ins should be tested for performance issues, especially on devices with slower CPUs and limited memory.

The JavaScript needed to run these plug-ins is located in bootstrap.js. Each plug-in is also available as a standalone file, which enables you to include only the plug-ins your app needs, resulting in faster first-page rendering.

Figure 9 shows how to use the Carousel plug-in, which provides slideshow functionality. If your page already has a link to bootstrap.js or bootstrap.min.js, you're all set. Or you can link to carousel.js if the Carousel plug-in is the only plug-in your page requires.

Figure 10 shows the Carousel plug-in rendered for Windows

Figure 6 Using the Bootstrap Navbar Component

```
<nav class="navbar navbar-default">
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
        data-target="#bs-example-navbar-collapse-1">
        <span class="sr-only">Toggle navigation</span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="#">Brand</a>
  </div>
  <!-- Collect the nav links, forms, and other content for toggling -->
  <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Link 1
        <span class="sr-only">(current)</span></a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
    </ul>
  </div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
```

Phone. The Carousel plug-in stretches itself appropriately for tablet, desktop and large desktop form factors.

Considerations

As I noted earlier, Bootstrap won't be right for every situation. To determine if Bootstrap is right for your site, there are a few things you'll want to consider:

1. If you have an existing site that has been performance-tuned over the years and has been optimized for your users, you'll want to proceed with caution. Prototyping complex, highly used pages is also a good idea. If a Bootstrap-enabled page doesn't perform adequately, consider setting up an m-site and coding everything yourself. Dino Esposito's article, "Mobilize an Existing Web Site" (bit.ly/1CaVEWR), shows how to set up an m-site.
2. If you're building a new site, prototype your most complex pages and the pages you believe will be most heavily used. Make sure they work well on low-end devices and in areas

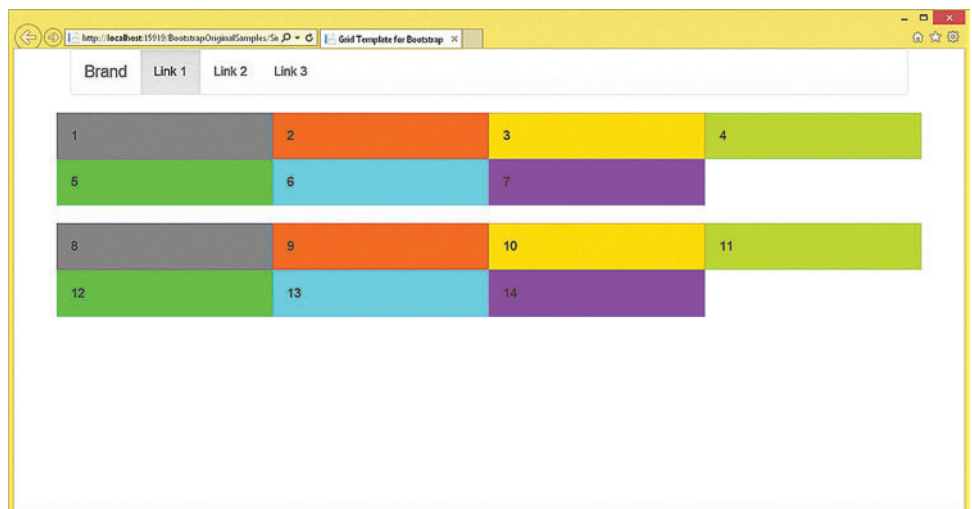


Figure 7 The Navbar Rendered for Browser Widths Equal to or Greater Than 768 Pixels

LightningChart



- Entirely DirectX GPU accelerated
- Superior 2D and 3D rendering performance
- Optimized for real-time data monitoring
- Touch-enabled operations
- Supports gigantic data sets
- On-line and off-line maps
- Great customer support
- Compatible with Visual Studio 2005...2013
- Now with Audio I/O components





Figure 8 The Navbar Rendered for Browser Widths Less Than 768 Pixels

of low bandwidth. If a page proves to be problematic then try to redesign it such that it will perform adequately. An m-site is also an option for new sites, but this should be done only if a redesign will not work.

3. Make sure that the Bootstrap grid system and the components used for navigation aren't too limiting with respect to site design. Check with your UX designers early in the process of investigating Bootstrap. In my opinion, a tool like Bootstrap forces you to design for mobile, which forces you to keep your pages clean and uncluttered. However, your UX designer can confirm whether Bootstrap can be made to properly represent your brand and provide an experience suitable for your users.

Figure 9 Using the Carousel Plug-In

```
<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#carousel-example-generic" data-slide-to="0"
        class="active"></li>
    <li data-target="#carousel-example-generic" data-slide-to="1"></li>
    <li data-target="#carousel-example-generic" data-slide-to="2"></li>
  </ol>
  <!-- Wrapper for slide content -->
  <div class="carousel-inner" role="listbox">
    <div class="item">
      
      <div class="carousel-caption">
        ...
      </div>
    </div>
    ... Only one slide for brevity.
  </div>
  <!-- Controls -->
  <a class="left carousel-control" href="#carousel-example-generic" role="button"
     data-slide="prev">
    <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="right carousel-control" href="#carousel-example-generic" role="button"
     data-slide="next">
    <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

More Information and Tools

If you're going to develop your site using Bootstrap, you'll want to review all the components and JQuery plug-ins documented on the Bootstrap site, where you'll also find many useful examples. You should also check out:

- Bootswatch (bootswatch.com) contains free themes for Bootstrap.
- Wrap Bootstrap (wrapbootstrap.com) contains inexpensive Bootstrap themes and templates.
- Font Awesome (fontawesome.io) is a site that provides free scalable vector icons that can be easily customized with CSS.
- "Deploy an ASP.NET MVC 5 Mobile Web Application on Azure Websites" (bit.ly/1CM0Gwq) shows how to apply Bootstrap to an existing Web site to make it mobile-friendly. The article also shows how to deploy a site to Microsoft Azure.

Make sure that the Bootstrap grid system and the components used for navigation aren't too limiting with respect to site design.

Wrapping Up

The material presented here is just a brief overview of the Bootstrap UI framework. It by no means completely describes all the CSS classes, components and plug-ins that come with Bootstrap. You'll want to go to the Bootstrap site to get a complete understanding of all the tools available, and to explore the sample pages that demonstrate different UX designs. These samples are a good place to start if you need to get a feel for what's possible with Bootstrap.

Before moving forward with a serious development effort using Bootstrap or any other framework, do consider performance and usability. Prototyping key pages is the best way to confirm that Bootstrap can be incorporated into your site and give you the results you want. ■

KEITH PIJANOWSKI is an engineer, entrepreneur and business guy. He has more than 20 years of experience in the software industry and has worked for startups and large companies in roles that range from writing code to business development. Reach him at keithpij@msn.com or twitter.com/keithpij.

THANKS to the following Microsoft technical experts for reviewing this article:
Rick Anderson and Cephas Lin

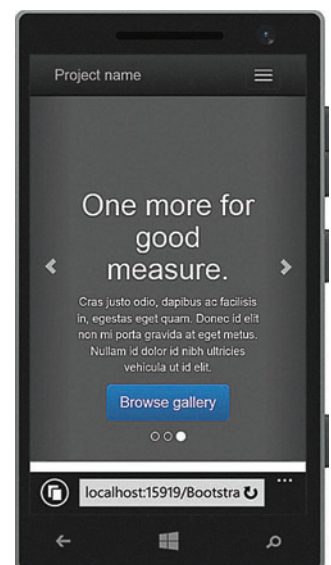


Figure 10 The Carousel Example Rendered in the Windows Phone Emulator

Join us on the Ultimate Code Trip in 2015!

HURRY—
LAST CHANCE!



San Francisco

JUNE 15 - 18

THE FAIRMONT

vslive.com/sf

See pages 48-49 for more info



Redmond

AUGUST 10 - 14

MICROSOFT HEADQUARTERS

vslive.com/redmond

See pages 62-65 for more info



New York

SEPTEMBER 28 - OCTOBER 1

NY MARRIOTT AT BROOKLYN BRIDGE

vslive.com/newyork

See pages 50-51 for more info



Orlando

NOVEMBER 16 - 20

LOEWS ROYAL PACIFIC RESORT

A Part of Live! 360

live360events.com

See pages 78-79 for more info

CONNECT WITH VISUAL STUDIO LIVE!



twitter.com/vslive – @VSLive



facebook.com – Search "VSLive"



linkedin.com – Join the
"Visual Studio Live" group!

TURN THE PAGE FOR
MORE EVENT DETAILS.





CODE BY THE BAY

Visual Studio Live! returns to **San Francisco June 15 – 18** for the first time since 2009! Bring on the cable cars, Chinatown, Pier 39, Alcatraz, and the Golden Gate Bridge. We can't wait to Code by the Bay!

Join us as we explore the latest features of Visual Studio, JavaScript/HTML5, ASP.NET, Database Analytics, and more over 4 days of sessions and workshops. Code with industry experts, get practical answers to your current challenges, and immerse yourself in what's to come on the .NET horizon.

DEVELOPMENT TRACKS INCLUDE:

- Visual Studio / .NET
- Web Development
- Cloud Computing
- Mobile Client
- Database and Analytics
- Windows Client

SESSIONS
ARE FILLING
UP QUICKLY—
REGISTER TODAY!



Scan the QR code to register or for more event details.

Use promo code VSLJUN5

vslive.com/sf

Cloud Computing	Mobile Client	Database and Analytics	Web Development	Visual Studio / .NET	Windows Client
-----------------	---------------	------------------------	-----------------	----------------------	----------------

START TIME	END TIME	Visual Studio Live! Pre-Conference Workshops: Monday, March 16, 2015 (Separate entry fee required)		
9:00 AM	6:00 PM	M01 Workshop: Big Data, Analytics and NoSQL: Everything You Wanted to Learn But Were Afraid to Ask - Andrew Brust	M02 Workshop: Native Mobile App Development for iOS, Android and Windows Using C# - Marcel de Vries and Roy Cornelissen	M03 Workshop: ALM and DevOps with the Microsoft Stack - Brian Randell
6:45 PM	9:00 PM	Dine-A-Round with Speakers		

START TIME	END TIME	Visual Studio Live! Day 1: Tuesday, June 16, 2015			
8:00 AM	9:00 AM	KEYNOTE: The Future of Application Development - Visual Studio 2015 and .NET 2015 - Jay Schmelzer, Director of Program Management, Visual Studio Team, Microsoft			
9:15 AM	10:30 AM	T01 Azure 10-10: Top 10 Azure Announcements in "T-10" Months - Vishwas Lele	T02 AngularJS 101 - Deborah Kurata	T03 UX Design Principle Fundamentals for Non-Designers - Billy Hollis	T04 A Lap Around Visual Studio 2015 - Robert Green
10:45 AM	12:00 PM	T05 Cloud or Not, 10 Reasons Why You Must Know "Web Sites" - Vishwas Lele	T06 AngularJS Forms and Validation - Deborah Kurata	T07 Designing and Building UX for Finding and Visualizing Data in XAML Applications - Billy Hollis	T08 Managing Dev/Test Environments in Azure - Claude Remillard
12:00 PM	1:30 PM	Lunch — Visit Exhibitors			
1:30 PM	2:45 PM	T09 Building Mobile Cross-Platform Apps with C# and Xamarin - Nick Landry	T10 ASP.NET 5 in all its Glory - Adam Tuliper	T11 Getting Started with Universal Apps for Windows and Windows Phone - Philip Japikse	T12 Mobile Apps for the JavaScript Developer - Ryan J. Salva
3:00 PM	4:15 PM	T13 Building Mobile Cross-Platform Apps in C# with Azure Mobile Services - Nick Landry	T14 Hack Proofing Your Web Applications - Adam Tuliper	T15 Building Windows 10 LOB Apps - Robert Green	T16 What's New in ALM - Brian Randell
4:15 PM	5:30 PM	Welcome Reception			

START TIME	END TIME	Visual Studio Live! Day 2: Wednesday, June 17, 2015			
8:00 AM	9:00 AM	KEYNOTE: Keynote: Cloud Developer - Are You One? - Kris Lankford, Sr. Product Manager, Microsoft			
9:15 AM	10:30 AM	W01 iOS Development - What They Don't Tell You - Jon Flanders	W02 Implementing M-V-VM (Model-View-View Model) for WPF - Philip Japikse	W03 Moving Web Apps to the Cloud - Eric D. Boyd	W04 Stop the Waste and Get Out of (Technical) Debt - Richard Hundhausen
10:45 AM	12:00 PM	W05 Swift for .NET Developers - Jon Flanders	W06 Strike Up a Conversation with Cortana on Windows Phone - Walt Ritscher	W07 Solving Security and Compliance Challenges with Hybrid Clouds - Eric D. Boyd	W08 Best Practices for Using Open Source Software in the Enterprise - Marcel de Vries
12:00 PM	1:30 PM	Birds-of-a-Feather Lunch — Visit Exhibitors			
1:30 PM	2:45 PM	W09 Building Cross Platform UI with Xamarin.Forms - Walt Ritscher	W10 Take a Gulp, Make a Grunt, and Call Me Bower - Adam Tuliper	W11 Introduction to R for C# Programmers - James McCaffrey	W12 Automated Build, Test & Deploy with TFS, ASP.NET, and SQL Server - Benjamin Day
3:00 PM	4:15 PM	W13 Adding Analytics to Your Mobile Apps on Any Platform with Microsoft Application Insights and Hockeyapp - Marcel de Vries	W14 Securing Angular Apps - Brian Noyes	W15 Microservices. What's the Big Deal? - Rick Garibay	W16 Professional Scrum Development Using Visual Studio 2015 - Richard Hundhausen
4:30 PM	5:45 PM	W17 Creating Applications Using Android Studio - Kevin Ford	W18 - Build Data-Centric HTML5 Single Page Applications with Breeze - Brian Noyes	W19 A Pragmatic Reference Architecture for The Internet of Things - Rick Garibay	W20 Load Testing ASP.NET & WebAPI with Visual Studio - Benjamin Day
7:00 PM	9:00 PM	Visual Studio Live! Evening Event			

START TIME	END TIME	Visual Studio Live! Day 3: Thursday, June 18, 2015			
8:00 AM	9:15 AM	TH01 Using Multi-Device Hybrid Apps to Create Cordova Applications - Kevin Ford	TH02 AngularJS & ASP.NET MVC Playing Nice - Miguel Castro	TH03 Windows, NUI, and You - Brian Randell	TH04 To Git or Not to Git for Enterprise Development - Benjamin Day
9:30 AM	10:45 AM	TH05 Everything You Always Wanted To Know About REST (But Were Afraid to Ask) - Jon Flanders	TH06 Knocking it Out of the Park with Knockout.JS - Miguel Castro	TH07 Building Rich Data Web APIs with ASP.NET OData - Brian Noyes	TH08 What's New in C# 6.0 - Jason Bock
11:00 AM	12:15 PM	TH09 Comparing Performance of Different Mobile Platforms - Kevin Ford	TH10 To Be Announced	TH11 Power BI 2.0: Analytics in the Cloud and in Excel - Andrew Brust	TH12 Microsoft's .NET is Now Open Source and Cross-Platform. Why it Matters. - Mark Rosenberg
12:15 PM	1:30 PM	Lunch			
1:30 PM	2:45 PM	TH13 Programming WPF with MVVM - Advanced Topics - Miguel Castro	TH14 Busy JavaScript Developer's Guide to ECMAScript 6 - Ted Neward	TH15 Big Data and Hadoop with Azure HDInsight - Andrew Brust	TH16 Managing the .NET Compiler - Jason Bock
3:00 PM	4:15 PM	TH17 Extending XAML To Overcome Pretty Much Any Limitation - Miguel Castro	TH18 Better Unit Tests through Design Patterns for ASP MVC, WebAPI, and AngularJS - Ben Day	TH19 Busy Developer's Guide to NoSQL - Ted Neward	TH20 Async and Await Best Practices - Mark Rosenberg

Sessions and speakers subject to change.



THE CODE THAT NEVER SLEEPS

Visual Studio Live! is hitting the open road on the ultimate code trip to help you navigate the .NET Highway. The next stop? NYC, and we're geared up to be back in the big apple for the first time in 2012.

From September 28 - October 1, Visual Studio Live! is bringing its unique brand of practical, unbiased, Developer training to Brooklyn, offering four days of sessions, workshops and networking events - all designed to help you avoid road blocks and cruise through your projects with ease.

FEATURED KEYNOTE SPEAKERS



Brian Harry,
Corporate
Vice President,
Microsoft



Mary Jo Foley,
Journalist and
Author

Register Now and Save \$300!



Use promo code VSLJUN5

Scan the QR code to register
or for more event details.

VSLIVE.COM/NEWYORK

Cloud Computing	Database and Analytics	Design	Mobile Client	Visual Studio / .NET	Web Development	Windows Client
-----------------	------------------------	--------	---------------	----------------------	-----------------	----------------

START TIME	END TIME	Visual Studio Live! Pre-Conference Workshops: Monday, September 28, 2015 (Separate entry fee required)		
9:00 AM	6:00 PM	M01 - Workshop: Big Data, Analytics and NoSQL: Everything You Wanted to Learn But Were Afraid to Ask - Andrew Brust	M02 - Workshop: Native Mobile App Development for iOS, Android and Windows Using C# - Marcel de Vries and Roy Cornelissen	M03 - Workshop: ALM and DevOps with the Microsoft Stack - Brian Randell
6:45 PM	9:00 PM	Dine-A-Round		

START TIME	END TIME	Visual Studio Live! Day 1: Tuesday, September 29, 2015			
8:00 AM	9:00 AM	KEYNOTE: To Be Announced - Brian Harry, Corporate Vice President, Microsoft			
9:15 AM	10:30 AM	T01 - AngularJS 101 - Deborah Kurata	T02 - Azure 10-10: Top 10 Azure Announcements in "T-10" Months - Vishwas Lele	T03 - UX Design Principle Fundamentals for Non-Designers - Billy Hollis	T04 - Microsoft Session To Be Announced
10:45 AM	12:00 PM	T05 - AngularJS Forms and Validation - Deborah Kurata	T06 - Cloud or Not, 10 Reasons Why You Must Know "Web Sites" - Vishwas Lele	T07 - Designing and Building UX for Finding and Visualizing Data in XAML Applications - Billy Hollis	T08 - Microsoft Session To Be Announced
12:00 PM	1:30 PM	Lunch — Visit Exhibitors			
1:30 PM	2:45 PM	T09 - Building Mobile Cross-Platform Apps with C# and Xamarin - Nick Landry	T10 - Inside the Azure Resource Manager - Michael Collier	T11 - Windows, NUI and You - Brian Randell	T12 - Microsoft Session To Be Announced
3:00 PM	4:15 PM	T13 - Building Mobile Cross-Platform Apps in C# with Azure Mobile Services - Nick Landry	T14 - Automating Your Azure Environment - Michael Collier	T15 - Building Windows 10 LOB Apps - Billy Hollis	T16 - Defensive Coding Techniques in C# - Deborah Kurata
4:15 PM	5:30 PM	Welcome Reception			

START TIME	END TIME	Visual Studio Live! Day 2: Wednesday, September 30, 2015			
8:00 AM	9:00 AM	KEYNOTE: Microsoft 3.0: New Strategy, New Relevance - Mary Jo Foley, Journalist and Author; with Andrew Brust, Senior Director, Datameer			
9:15 AM	10:30 AM	W01 - iOS Development - What They Don't Tell You - Jon Flanders	W02 - Moving Web Apps to the Cloud - Eric D. Boyd	W03 - XAML Antipatterns - Ben Dewey	W04 - What's New in C# 6.0 - Jason Bock
10:45 AM	12:00 PM	W05 - Swift for .NET Developers - Jon Flanders	W06 - Solving Security and Compliance Challenges with Hybrid Clouds - Eric D. Boyd	W07 - Extending XAML to Overcome Pretty Much Any Limitation - Miguel Castro	W08 - Using Microsoft Application Insights to Implement a Build, Measure, Learn Loop - Marcel de Vries
12:00 PM	1:30 PM	Birds-of-a-Feather Lunch — Visit Exhibitors			
1:30 PM	2:45 PM	W09 - Building Cross-Platform C# Apps with a Shared UI via Xamarin.Forms - Nick Landry	W10 - To Be Announced	W11 - Real World SQL Server Data Tools - Benjamin Day	W12 - Enhancing Application Quality Using Visual Studio 2015 Premium Features - Anthony Borton
3:00 PM	4:15 PM	W13 - Creating Applications Using Android Studio - Kevin Ford	W14 - ASP.NET MVC: All Your Tests Are Belong To Us - Rachel Appel	W15 - Transact-SQL for Application Developers - Attendees Choose Topics - Kevin Goff	W16 - Automated Cross Browser Testing of Your Web Applications with Visual Studio CodedUI - Marcel de Vries
4:30 PM	5:45 PM	W17 - Using Multi-Device Hybrid Apps to Create Cordova Applications - Kevin Ford	W18 - Build Data-Centric HTML5 Single Page Applications with Breeze - Brian Noyes	W19 - SQL Server Reporting Services - Attendees Choose Topics - Kevin Goff	W20 - Managing the .NET Compiler - Jason Bock
7:00 PM	9:00 PM	Visual Studio Live! Evening Event			

START TIME	END TIME	Visual Studio Live! Day 3: Thursday, October 1, 2015			
8:00 AM	9:15 AM	TH01 - Everything You Always Wanted To Know About REST (But Were Afraid To Ask) - Jon Flanders	TH02 - Securing Angular Apps - Brian Noyes	TH03 - Implementing Data Warehouse Patterns - Attendees Choose - Kevin Goff	TH04 - Not Your Grandfather's Build - A Look at How Build Has Changed in 2015 - Anthony Borton
9:30 AM	10:45 AM	TH05 - Comparing Performance of Different Mobile Platforms - Kevin Ford	TH06 - I Just Met You, and "This" is Crazy, But Here's My NaN, So Call(Me), Maybe? - Rachel Appel	TH07 - Power BI 2.0: Analytics in the Cloud and in Excel - Andrew Brust	TH08 - Async Patterns for .NET Development - Ben Dewey
11:00 AM	12:15 PM	TH09 - To Be Announced	TH10 - Build Real-Time Websites and Apps with SignalR - Rachel Appel	TH11 - Busy Developer's Guide to NoSQL - Ted Neward	TH12 - DevOps and ALM-Better Together Like Peanut Butter and Chocolate - Brian Randell
12:15 PM	1:30 PM	Lunch			
1:30 PM	2:45 PM	TH13 - WCF & Web API: Can We All Just Get Along?! - Miguel Castro	TH14 - Busy JavaScript Developer's Guide to ECMAScript 6 - Ted Neward	TH15 - Big Data and Hadoop with Azure HDInsight - Andrew Brust	TH16 - To Git or Not to Git for Enterprise Development - Benjamin Day
3:00 PM	4:15 PM	TH17 - Recruiters: The Good, The Bad, & The Ugly - Miguel Castro	TH18 - Busy Developer's Guide to MEANJS - Ted Neward	TH19 - Predictive Analytics and Azure Machine Learning - Andrew Brust	TH20 - Load Testing ASP.NET & WebAPI with Visual Studio - Benjamin Day

Sessions and speakers subject to change.

Create Your Own Visual Studio Templates with SideWaffle

Tyler Hughes and Sayed Ibrahim Hashimi

Thanks to the release of the new Community Edition of Visual Studio, developers who have used the Express Editions in the past can now use the extensions they've heard so much about. In case you're wondering, extensions are plug-ins that can extend the functionality of Visual Studio. There are all sorts of extensions available.

We're always looking for ways to code more efficiently and, about a year ago, we came across an extension called SideWaffle (sidewaffle.com). SideWaffle adds new project and item templates made by other developers to Visual Studio, and it lets you create your own templates and code snippets. In this article, we'll take a look at SideWaffle and show you how to create your own templates and snippets.

Why Use SideWaffle?

As noted, SideWaffle adds new project and item templates to Visual Studio. You might be wondering, "How does this help me?" Let's say you're a freelance Web developer who develops Web sites for

clients. You probably have specific things you repeat for each site, perhaps using a third-party JavaScript library such as Knockout.js, or setting up a specific project folder structure. No matter how you set up a project, SideWaffle can turn it into a reusable template. Plus, SideWaffle includes a reusable NuGet package, TemplateBuilder, which builds templates into extensions..

Creating a Template Pack

Before creating a template, you need to set up your development environment. The first item you need, of course, is Visual Studio. For this article, we used the Community Edition of Visual Studio 2013 (bit.ly/1GUTLo3). Next, you need the Visual Studio 2013 SDK (bit.ly/1NZVkd9). Finally, download and install the SideWaffle extension from sidewaffle.com.

With all the tools installed and ready to go, start by opening Visual Studio and creating a new project. There are two types of projects you can use to create a template pack. The first is a VSPackage project, which gives you the ability to add features like context menus or menu items in Visual Studio. The second is a VSIX project, which is just a container to hold project files and resources and doesn't provide additional functionality. You can find both of these in the New Project dialog under the Extensibility node. Today, we want just a basic template, so a VSIX project will work. You can leave the default project name.

Once the project is created, its manifest file will open automatically. The Author field is required, so add your name (or company name) here. The Description is what will be shown in the Visual Studio extension dialog box, so be sure to add some useful text. Then save the file and close it. The next step is to add the TemplateBuilder

This article discusses:

- Creating a Visual Studio template project and adding the NuGet TemplateBuilder package
- Creating a project template and item templates
- Adding snippets to a VSIX project

Technologies discussed:

Visual Studio 2013 Community Edition, C#, SideWaffle

Code download available at:

bit.ly/19QIBU3

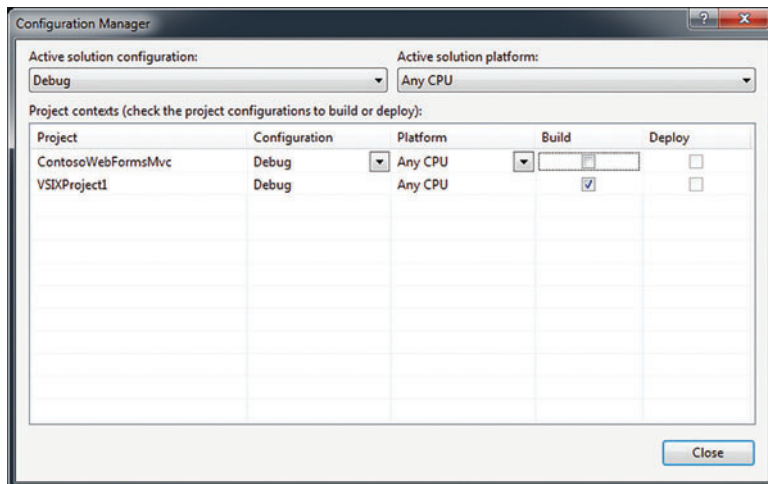


Figure 1 Disabling the Debug Configuration from Being Built

NuGet package to the project, which you can do by using either the Manage NuGet Packages dialog or the Package Manager Console.

Let's stop for a second now and take a look behind the scenes. When you install TemplateBuilder into a project, here's what happens:

1. A reference to TemplateBuilder.dll is added. This assembly contains custom wizards that can be used during the template-creation process, as well as other helpers.
2. The project file is modified to import an additional MSBuild targets file, allowing the build process to support adding templates to the generated VSIX. Because this is implemented with MSBuild, you can build your extension on continuous integration servers and from the command line.
3. A props file is added to Properties\template-builder.props. You can specify the default leaf name of the node that should contain your templates. You can also use this file to control specific aspects of TemplateBuilder.

Figure 2 A Generic_project.vstemplate.xml File

```
<VSTemplate Version="3.0.0"
  xmlns="http://schemas.microsoft.com/developer/vstemplate/2005"
  Type="Project">
  <TemplateData>
    <Name>Your name here</Name>
    <Description>Project description here</Description>
    <DefaultName>ContosoWebFormsMvc</DefaultName>

    <ProjectType>CSharp</ProjectType>
    <ProjectSubType></ProjectSubType>
    <SortOrder>1000</SortOrder>
    <CreateNewFolder>true</CreateNewFolder>
    <ProvideDefaultName>true</ProvideDefaultName>
    <LocationField>Enabled</LocationField>
    <EnableLocationBrowseButton>true</EnableLocationBrowseButton>
    <Icon>sw-file-icon.png</Icon>

    <!-- Indicates how many parent folders this item template should appear in -->
    <NumberOfParentCategoriesToRollUp>1</NumberOfParentCategoriesToRollUp>
  </TemplateData>
  <TemplateContent>
    <Project TargetFileName="ContosoWebFormsMvc.csproj"
      File="ContosoWebFormsMvc.csproj"
      ReplaceParameters="true">
    </Project>
  </TemplateContent>
</VSTemplate>
```

4. The VSIX manifest (typically named source.extension.vsixmanifest) is modified to include Asset tags for the generated templates.

With TemplateBuilder installed, you can start creating templates. When you build the VSIX project, TemplateBuilder automatically kicks in and starts processing the templates.

Adding TemplateBuilder to the project completes all of the preliminary requirements. Now we can add the project template to the currently open solution.

Creating the First Project Template

We're going to use an existing project, which happens to be a generic Contoso ASP.NET Web Forms/MVC project. This project uses Bootstrap 3, jQuery and Modernizr, which will help speed up development any time we create a new project based on this template. As a rule, it's a good idea to first run any project after adding it to the solution. This is just a precaution to make sure the project works as expected before creating the template. So run this project (Ctrl+F5) and make sure it's working.

This project uses Bootstrap 3, jQuery and Modernizr, which will help speed up development any time we create a new project based on this template.

To create a project template, you need to do the following:

- Add the project to the solution and disable it from building.
- Add a Template Reference to the VSIX project.
- Add metadata files to the source project for template information.

To keep the template project from being built any more by Visual Studio in this solution (you can still build and run this project in other solutions), click on the dropdown menu for the build configuration and select Configuration Manager from the list. This opens the Configuration Manager dialog. To keep the project from being built, uncheck it in the Build column. Once you've done that for the Debug configuration, select the Active solution configuration dropdown and repeat the process for the Release configuration.

There may come a time when you want to create a template containing multiple projects. If you end up working on a complex project like that and get to this step, keep in mind the only thing that should be checked for both the Debug and Release configurations is either the VSIX project or the VSPackage project, depending on which one you chose earlier. **Figure 1** shows the Configuration Manager Window after stopping the project from being built.

Next, you want to connect the VSIX project to the main project, which you'll do by adding a Template Reference. Right-click on

the VSIX project and select Add | Add Template Reference (SideWaffle project) from the options.

A Project Selector dialog will appear. From the list, select the project you want to use as your template, and click OK. This will make some changes to your VSIX project and will prompt you to reload the project. You can select the Reload All option and then wait for SideWaffle to finish.

Now, let's add the template metadata files to the Web project. Right-click on your main project and select Add | New Item from the options listed. In the Add New Item dialog that appears, go to the Extensibility section and select SideWaffle Project Template Files. You can use the default name given by Visual Studio. Once you've selected the Project Template Files item template, click the Add button.

The Project Template files item adds two new files to the main project. First, it adds `_preprocess.xml`, which tells Visual Studio where to place your template when you open the New Project dialog. Here's the default content for this file:

```
<?xml version="1.0" encoding="utf-8" ?>
<Preprocess>
  <TemplateInfo Path="CSharp\Web\SideWaffle"/>
  <Replacements Include="*.*)" Exclude="*.vstemplate;
    *.jpg;*.png;*.ico;_preprocess.xml;_project.vstemplate.xml">
    <add key="ContosoWebFormsMvc" value="$safeprojectname$"/>
  </Replacements>
</Preprocess>
```

Figure 4 A Basic HTML5 Bootstrap Template

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Basic Bootstrap Template</title>
  <link href="css/bootstrap.min.css" rel="stylesheet">

  <!-- HTML5 shim and Respond.js for IE8 support of -->
  <!-- HTML5 elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
  <!--[if lt IE9]>
  <script src=
    "https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
  <script src=
    "https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
  <![endif]>
</head>
<body>
  <h1>Hello, world!</h1>

  <script src=
    "https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js">
  </script>
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

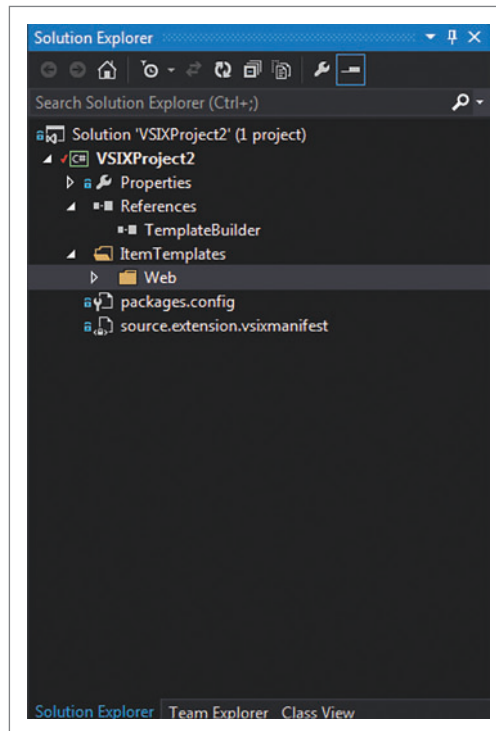


Figure 3 Your Template Will Be Displayed in the Web Folder

In this file, you'll find the following information:

- The path for the node where the template will appear in the New Project dialog.
- Information about source replacements.

The Path attribute on the TemplateInfo element specifies the location where the template will appear in the New Project Dialog.

Within the Replacements element, you can declare any replacements that should be applied during the template build process. You get a default replacement for the project name, which will be replaced with `$safeprojectname$` in the generated template. This will typically update namespace declarations. By using `$safeprojectname$` when the template is used in Visual Studio, the namespace declarations will be updated with the name specified by the user. You're not limited to one replacement; you can add as many as needed. Sometimes you may need to modify sources

to make selections unique so that replacements don't conflict.

Now, let's move on to the other file that was added to the project, `_project.vstemplate.xml`, which contains metadata about the template. Figure 2 shows the default content for the file, where ContosoWebFormsMvc is the name of the project.

This is a standard `.vstemplate` file; you can find the reference for this file at bit.ly/18T090m. During the template build process, files from the project will be merged into this to create the final `.vstemplate` file that's used for the template. There are a few items to note here. You should update the following elements under TemplateData:

- Name: This is the name of your template as shown in the New Project Dialog.
- Description: This is the description of your project that's shown on the right hand side of the New Project Dialog when the template is selected.
- DefaultName: This is the default name of the new project being created. A number will be automatically appended to this based on directory content.
- SortOrder: This determines the position of the templates with respect to other templates. This must be a multiple of 10; the lower the number, the higher the item is sorted in Visual Studio.
- NumberOfParentCategoriesToRollup: This determines how many parent nodes in which the template will be shown.

You should also double-check the values for the TargetFileName and File attributes on the Project tag. These should match the name of the project on disk. When TemplateBuilder processes this file, the final `.vstemplate` file will merge source files listed in the project. If you need special handling for any particular files, you can list those under the Project element. See the `.vstemplate` reference linked previously.

facebook



Microsoft
SharePoint 2010



Linked in



twitter

SEE THE WORLD AS A DATABASE

ADO.NET ▪ JDBC ▪ ODBC ▪ SQL SSIS ▪ ODATA
MYSQL ▪ EXCEL ▪ POWERSHELL



Microsoft Visual Studio Java ODBC Microsoft SQL Server Microsoft Excel Microsoft BizTalk MySQL OData

Work With Relational Data, Not Complex APIs or Services

Whether you are a developer using ADO.NET, JDBC, OData, or MySQL, or a systems integrator working with SQL Server or Biztalk, or even an information worker familiar with ODBC or Excel – our products give you bi-directional access to live data through easy-to-use technologies that you are already familiar with. If you can connect to a database, then you will already know how to connect to Salesforce, SAP, SharePoint, Dynamics CRM, Google Apps, QuickBooks, and much more!



Give RSSBus a try today and see what mean:

visit us online at www.rssbus.com to learn more or download a free trial.

rssbus

INTEGRATION YOUR WAY

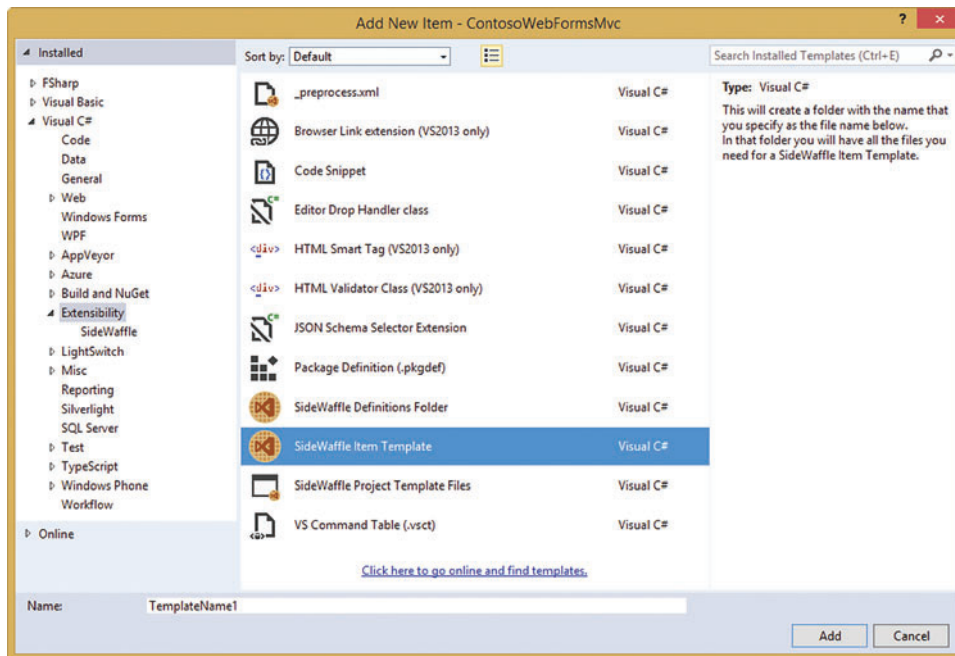


Figure 5 The SideWaffle Item Template Is Located in the Extensibility Node of the Add New Item Dialog

Now that you've customized both `_preprocess.xml` and the `.vstemplate.xml` file, it's time to test the template. Make sure the VSIX project is configured as the startup project in Visual Studio and then press `Ctrl+F5`. This launches the Experimental Instance of Visual Studio, and your project template will be loaded automatically. Open the New Project dialog. If you go to the location that was set in your `_preprocess.xml` file, you should see your newly created template. Go ahead and create a new project using your new template to test it out. Make sure everything is working as expected.

If everything is working well, congratulations are in order as you just created your first project template. If you want to install this template or share it with others, go to your `bin` folder and find the VSIX file to install. If you or your team plans to share your project via a Git repository, there's one additional step you need to do before uploading it.

When you create a VSIX or VSPackage project, Visual Studio updates your project file with values for the `StartProgram` and `StartArguments` tags. These values are used to ensure that `F5` and `Ctrl+F5` work correctly. However, these changes are configured only on a per-user basis. When you add your project to your repo, Visual Studio doesn't include these changes. They're saved in a `.user` file that's not checked in, so when another user opens the solution and tries to run the project in an Experimental Instance, it may not work. To fix this, simply open

your project's `.csproj` file and add the following lines just before the end of the first property group:

```
<StartProgram Condition=" '$(StartProgram)'!=' ' ">
  "$(DevEnvDir)\devenv.exe
</StartProgram>
<StartArguments Condition="
  " '$(StartArguments)'!=' ' " ">
  /rootsuffix Exp</StartArguments>
```

Now run the project again just to make sure nothing got messed up. If it works as expected, you're ready to add the project to your Git repository and share it with everyone.

You've just created a Visual Studio extension that contains a single project template. You can repeat this process to include additional templates in your extension. There's no limit to the number of templates in an extension. You can share your working template with others so that they can try it. To do that you need to

distribute the `.vsix` file that's in the `bin` folder.

If you want to make your extension publicly available, you can upload it to the Visual Studio Gallery at bit.ly/115mBzm. If your extension contains more than one template (either a project or item template), the Gallery won't accept those extensions today. Instead of uploading them directly, you can upload those extensions as a link in the Visual Studio Gallery. If you do this, you'll need to host the `.vsix` file on a publicly available URL. You can use vsixgallery.com to host the `.vsix` file and you can use it to host nightly builds. You can easily automate publishing nightly builds to vsixgallery.com using Windows PowerShell or C#.

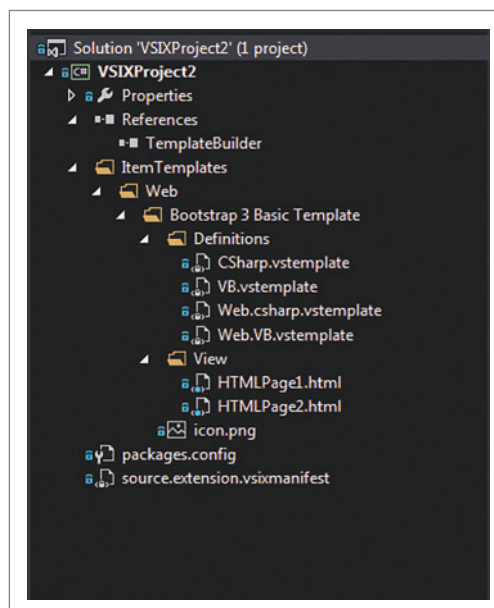


Figure 6 The SideWaffle Item Template Adds a Definitions Folder and Four vstemplate Files

We've covered just the basics of creating project templates. There are other scenarios, such as multi-project templates, using custom wizards and more. You can look at the SideWaffle wiki on GitHub for more information. Now it's time to move on to item templates.

Adding Item Templates

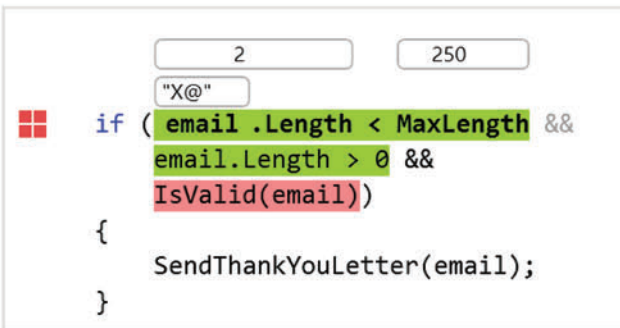
In addition to letting you create your own project templates, SideWaffle gives you the ability to create your own item templates. We'll add to the extension project we just created.

We noted earlier that the project used Bootstrap 3, jQuery and Modernizr. Once the basic project has been created, anytime we want to create a new page for the Web site we have to manually create a new HTML file and include all the CSS and JavaScript files. It would be nice to create a custom item template that does all this for us.

Debug Like a Boss

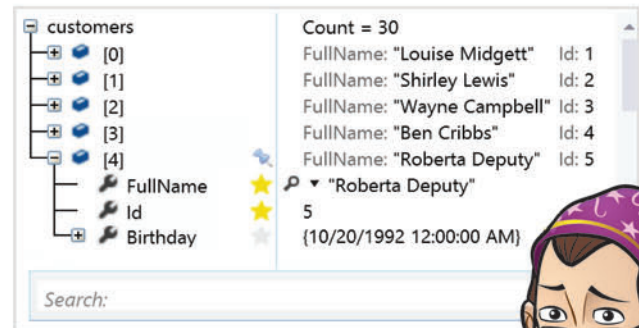
Enhance Your C# Debugging Experience

Simplify



Make sense of complex expressions and predict the next result

Reveal



Visualize objects and focus on data that actually matters

Search

Find a needle in a haystack of data

Compare

View differences between objects

Trace

Log, view and filter debug messages

Exception Trail

Easily navigate inner exceptions

Quick Actions

Automate your everyday debugging

Show All Instances

Track down an object in memory

Supports Visual Studio 2010, 2012, 2013



Get an exclusive
DISCOUNT



oz-code.com | info@oz-code.com

Powered by
CodeValue

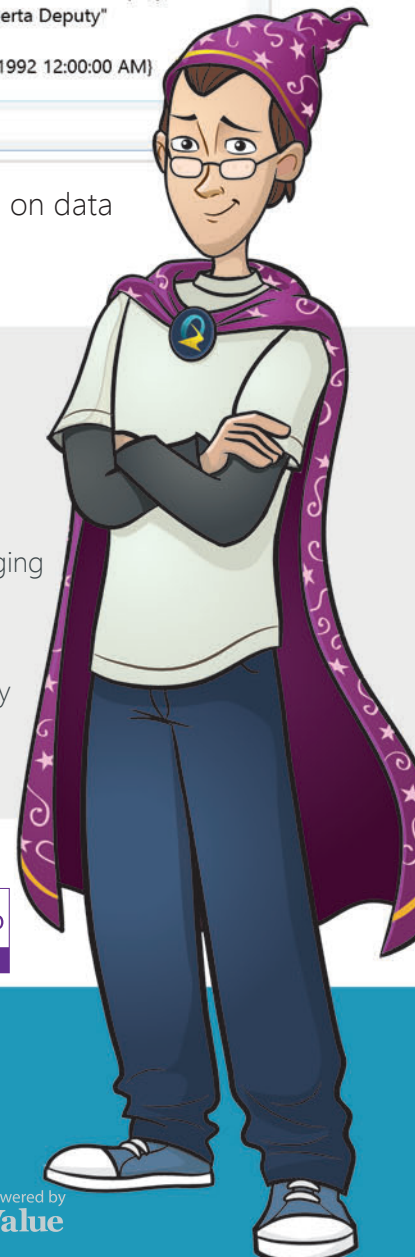


Figure 7 A Sample _project.vstemplate.xml File with Multiple Files Added to the Item Template

```
<VSTemplate Version="3.0.0"
  xmlns="http://schemas.microsoft.com/developer/vstemplate/2005" Type="Item">
  <TemplateData>
    <ProjectType>CSharp</ProjectType>

    <DefaultName>Bootstrap 3 Basic Template.txt</DefaultName>
    <Name>Bootstrap 3 Basic Template</Name>
    <Description>A basic HTML5 template using Bootstrap 3</Description>
    <Icon>icon.png</Icon>
    <NumberOfParentCategoriesToRollUp>1</NumberOfParentCategoriesToRollUp>
  </TemplateData>
  <TemplateContent>
    <References />

    <ProjectItem TargetFileName="View/HTMLPage1.html" >
      View/HTMLPage1.html</ProjectItem>
    <ProjectItem TargetFileName="View/HTMLPage2.html" >
      View/HTMLPage2.html</ProjectItem>

  </TemplateContent>
</VSTemplate>
```

Luckily, TemplateBuilder processes project and item templates in the same way to keep things consistent for working with both, so some of the steps for creating an item template are identical to those we already did for the project template. During the template build, item and project templates are essentially processed in the same way. If you're getting started with a new project, make sure to add the TemplateBuilder NuGet package first.

Next, create a folder called ItemTemplates inside the VSIX project. As shown in **Figure 3**, this is the folder where all item templates will be stored.

Directly under the ItemTemplates folder you'll create directories for your item template files. The name of the first child of ItemTemplates will be the node in which the template will appear by default. For example, in **Figure 3**, you can see that we created a Web folder, so templates in that folder will appear under Visual C# \Web, or Visual Basic \Web for VB or similarly for other languages. You can also use _preprocess.xml to change this, but for this sample, we'll place all the item templates under the Web folder.

Now let's create our first item template. For each item template, we'll create a unique folder under Web, then add our first item template, a basic bootstrap template. To get started, create a new folder under Web named Bootstrap 3 Basic Template. This template will contain two HTML files (HTMLPage1.html and HTMLPage2.html), so place them into a folder named View. For this sample, you can use the content in **Figure 4**, which was taken from the Bootstrap 3 docs at bit.ly/1iKHGX3, for both files.

After adding the source files for creating the template, we'll add the files

that include the metadata about the item template. To do this, right-click on the Bootstrap 3 Basic Template folder and select Add New Item. As shown in **Figure 5**, select the SideWaffle Item Template under the Extensibility node.

In addition to letting you create your own project templates, SideWaffle gives you the ability to create your own item templates.

The value in the Name text box is ignored here because the files have hardcoded filenames. When you use the SideWaffle Item Template template, you get four files in a new folder named Definitions (see **Figure 6**). Those four files are CSharp.vstemplat-, VB.vstemplat-, Web.csharp.vstemplat- and Web.VB.vstemplat, corresponding to the top-level languages/projects supported. CSharp.vstemplate is used for templates that will show up under the Visual C# node for C# projects. VB.vstemplate is used for templates that will show up under Visual Basic in the New Item dialog. The two Web files are to show the template for Web projects. Note that to display your template, you need to change the extension from CSharp.vstemplat- to CSharp.vstemplate. You can delete the files in which you're not interested. As shown in **Figure 6**, we'll keep all four and change the extension to .vstemplate.

Now we need to edit the .vstemplate files to include the two HTML source files in the template, so we'll add some ProjectItem elements to the CSharp.vstemplate file under the TemplateContent element. The content of the file should look like what's shown in **Figure 7**.

As you can see, we added two ProjectItem entries corresponding to the two files that were created. The value of the element points to the file on disk where the template content is. The value of TargetFileName will determine the names of the files that appear when the New Item dialog is invoked.

Once you've added the item template and included your files, you can test the item template. Like last time, press Ctrl+F5 to open the Experimental Instance of Visual Studio, then create or open a sample project and try out your template.

Next, we'll discuss snippets, but first a quick note about replacements. In the Project Template section, you saw how _preprocess.xml could be used to update the project content during the template build process. If you need to perform

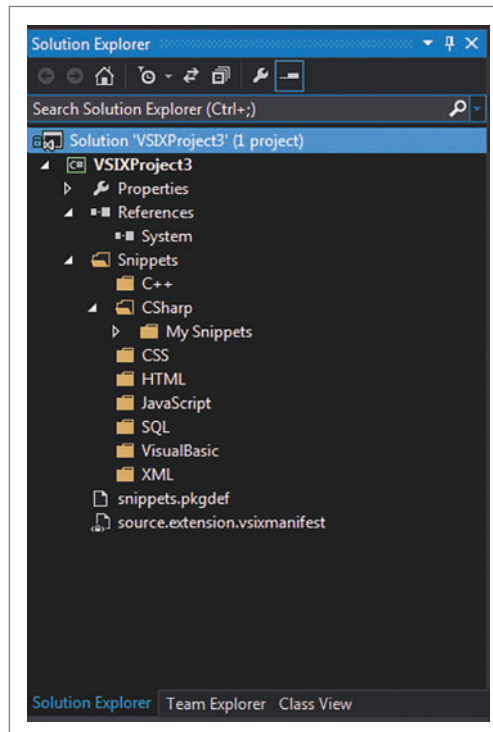


Figure 8 The Folders Shown Here Are Automatically Recognized by Visual Studio

We know how application development can be.



Let DotImage take some of the bite out of your challenges.

**Atalsoft
DotImage®**

Connecting the dots is a no-brainer. DotImage image-enables your .NET-based web application faster, more cost effectively, and less painfully than if done on your own. This proven SDK is versatile, with options including OCR capabilities, WingScan compatibility, and support for a range of formats. Coupled with dedicated assistance from our highly knowledgeable and skilled engineers, DotImage helps your business connect with powerful information hidden inside your documents, making the big picture much easier to see.

Figure 9 A Package Definition File Containing Registry Information for Each Language

```
// ----- SNIPPETS -----  
  
// Visual Basic  
[$RootKey\Languages\CodeExpansions\Basic\Paths]  
"My Snippets"="$PackageFolder$\Snippets\VisualBasic\My Snippets"  
  
// C++  
[$RootKey\Languages\CodeExpansions\C\C++\Paths]  
"My Snippets"="$PackageFolder$\Snippets\C++\My Snippets"  
  
// C#  
[$RootKey\Languages\CodeExpansions\CSharp\Paths]  
"My Snippets"="$PackageFolder$\Snippets\CSharp\My Snippets"  
  
// CSS  
[$RootKey\Languages\CodeExpansions\CSS\Paths]  
"My Snippets"="$PackageFolder$\Snippets\CSS\My Snippets"  
  
// HTML  
[$RootKey\Languages\CodeExpansions\HTML\Paths]  
"My Snippets"="$PackageFolder$\Snippets\HTML\My Snippets"  
  
// JavaScript  
[$RootKey\Languages\CodeExpansions\JavaScript\Paths]  
"My Snippets"="$PackageFolder$\Snippets\JavaScript\My Snippets"  
  
// SQL  
[$RootKey\Languages\CodeExpansions\SQL_SSDT\Paths]  
"My Snippets"="$PackageFolder$\Snippets\SQL\My Snippets"  
  
// XML  
[$RootKey\Languages\CodeExpansions\XML\Paths]  
"My Snippets"="$PackageFolder$\Snippets\XML\My Snippets"
```

any replacements for item templates, you can use the same technique. Just as with project templates, we've covered only the basics of creating item templates. There's a whole lot of ground we haven't gotten to—using custom wizards, embedding icons as resources, nesting files and more. For information on these topics, you can refer to existing content for standard Visual Studio templates. You can also look at the wiki on the SideWaffle GitHub project page.

How to Add Snippets

We all have snippets of code we use regularly in our projects. If you've been wondering if there's a way to get Visual Studio to keep track of your favorite snippets, you're in luck. Just like before, start out by creating a VSIX project and updating the author and description fields of the source.extension.vsixmanifest.

To have Visual Studio read your snippet files, you need to create a folder structure it recognizes. **Figure 8** shows the structure if you were to create snippets for each programming language.

Once you've created the folders you need, you can add your snippet files to the SideWaffle folder for the snippet's respective language. As you add each file, click on the file and look at its Build Action in the properties window. Each file should have its Build Action property set to Content so Visual Studio will automatically include the file in the VSIX project when it's built.

Next, you need to add a package definition file in the root directory of your project. The .pkgdef file will set some registry keys, allowing you to use the snippets added in the last step. Because Visual Studio doesn't come with an Item Template for .pkgdef files, you'll have to create a text file and rename it to use the .pkgdef extension. You can name the .pkgdef file whatever you like, but to keep things simple we named ours snippets.pkgdef. As shown in **Figure 9**, a different registry key needs to be updated for each language you use.

The last step before you can test your new snippets is to register the .pkgdef file. Open the source.extension.vsixmanifest file, switch to the Assets tab and click on the button labeled New. The Add New Asset dialog should appear, giving you several options. As shown in **Figure 10**, select Microsoft.VisualStudio.VsPackage for the asset type.

Next, select File on filesystem as the source type. This will cause the dialog to expand, giving you the ability to browse to the .pkgdef file you created. Select the .pkgdef file and click OK.

Now you're ready to test your snippets. As before, hit Ctrl+F5 to open an Experimental Instance of Visual Studio. Then open the project used earlier for testing and navigate to Tools | Code Snippets Manager. If everything loads correctly, you're done.

We've now looked at the main benefits of SideWaffle so far: project and item templates and code snippets. What's next?

The development team is currently working on a dynamic templates feature for SideWaffle that we hope to have ready for the next release. This will make it much simpler to create and maintain templates. The idea is that you can publish your templates to a Git repo or a network folder and share them with friends and colleagues. Then you can configure SideWaffle to pick up templates from the remote location. The end user will be able to control how often to check for updates. To keep up with this feature, visit the Dynamic Template page at bit.ly/18DepKM.

Wrapping Up

As you've seen in this article, creating your own templates is quite simple. Now that you know how to do this, you can show your friends and colleagues how much time you're saving with SideWaffle. Remember, when sharing your template pack with the community, you're always welcome to upload them to the Visual Studio Gallery, and vsixgallery.com is always available if you need it. If you

want to share your own templates with others via SideWaffle, send us a pull request on GitHub. Now, get started creating your template pack, and let us know if you need any help. ■

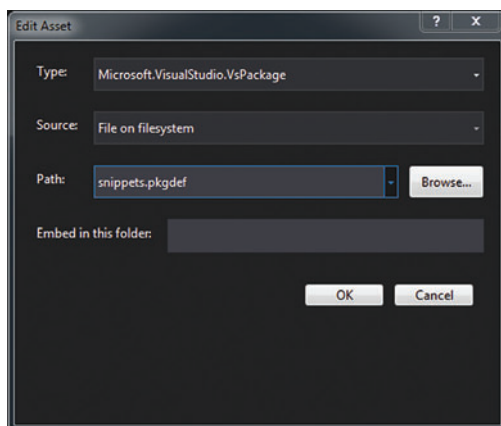


Figure 10 Registering the Package Definition File with Visual Studio

TYLER HUGHES is a recent graduate of Jacksonville State University where he studied Computer Information Systems. He has been a hobbyist developer for five years and is currently part of the SideWaffle core developer team.

SAYED IBRAHIM HASHIMI is a senior program manager at Microsoft on the Visual Studio Web team. He has written several books on Microsoft technologies and is the creator of SideWaffle and TemplateBuilder, as well as co-creator of OmniSharp.

Spreadsheets Made Easy.



Fastest Calculations

Evaluate complex Excel-based models and business rules with the fastest and most complete Excel-compatible calculation engine available.



Comprehensive Charting

Enable users to visualize data with comprehensive Excel-compatible charting which makes creating, modifying, rendering and interacting with complex charts easier than ever before.



Windows
Forms



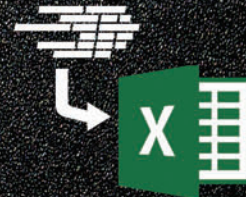
Silverlight



WPF

Powerful Controls

Add powerful Excel-compatible viewing, editing, formatting, calculating, filtering, sorting, charting, printing and more to your WinForms, WPF and Silverlight applications.



Scalable Reporting

Easily create richly formatted Excel reports without Excel from any ASP.NET, Windows Forms, WPF or Silverlight application.

Download your free fully functional evaluation at SpreadsheetGear.com



SpreadsheetGear

Toll Free USA (888) 774-3273 | Phone (913) 390-4797 | sales@spreadsheetgear.com



No code trip would be complete without a stop where it all began, so we're heading to the idyllic **Microsoft Headquarters in Redmond, WA, **August 10 – 14, 2015!****

Join us as we explore hot topics like Visual Studio, JavaScript/HTML5, ASP.NET, Database and Analytics, and more in over 70+ sessions and workshops. Rub elbows with Microsoft insiders, have lunch with Blue Badges, visit the company store, and experience code at the source.





DEVELOPMENT TRACKS INCLUDE:

- Visual Studio/.NET
- Web Development
- Design
- Mobile Client
- Windows Client
- Database and Analytics
- Cloud Computing
- Microsoft Sessions



REGISTER BY JUNE 3 AND SAVE \$400!



Scan the QR code to register
or for more event details.

Use promo code REDJUN4
vslive.com/redmond

Join us on the Ultimate Code Trip in 2015!



CONNECT WITH VISUAL STUDIO LIVE!



[@vslive](https://twitter.com/vslive)



facebook.com – Search “VSLive”



linkedin.com – Join the
“Visual Studio Live” group!

TURN THE PAGE FOR
MORE EVENT DETAILS.



Visual Studio Live! has partnered with the Hyatt Regency Bellevue for conference attendees at a special reduced rate.



REGISTER BY JUNE 3 AND SAVE \$400!



Scan the QR code to register or for more event details.

Use promo code REDJUN4
vslive.com/redmond

AGENDA AT-A-GLANCE

Cloud Computing

Database and Analytics

START TIME	END TIME	Visual Studio Live! Pre-Conference Workshops: Monday, August 10, 2015	
8:00 AM	12:00 PM	M01 – Workshop: Building Universal Apps for Desktop, Store, and Phone – Philip Japikse	
12:00 PM	2:30 PM	Lunch @ The Mixer – Visit the Microsoft Company Store & Visitor Center	
2:30 PM	6:00 PM	M01 – Workshop Continues	
7:00 PM	9:00 PM	Dine-A-Round Dinner	
START TIME	END TIME	Visual Studio Live! Day 1: Tuesday, August 11, 2015	
8:30 AM	9:30 AM	Keynote: Details Coming Soon	
9:45 AM	11:00 AM	T01 – Azure 10-10: Top 10 Azure Announcements in “T-10” Months – Vishwas Lele	T02 – Hack Proofing Your Web Applications – Adam Tuliper
11:15 AM	12:30 PM	T06 – Cloud or Not, 10 Reasons Why You Must Know “Web Sites” – Vishwas Lele	T07 – Take a Gulp, Make a Grunt, and Call Me Bower – Adam Tuliper
12:30 PM	2:30 PM	Lunch – Visit Exhibitors	
2:30 PM	3:45 PM	T11 – Building Mobile Cross-Platform Apps with C# and Xamarin – Nick Landry	T12 – AngularJS 101 – Deborah Kurata
3:45 PM	4:15 PM	Sponsored Break – Visit Exhibitors	
4:15 PM	5:30 PM	T16 – Building Mobile Cross-Platform Apps in C# with Azure Mobile Services – Nick Landry	T17 – AngularJS Forms and Validation – Deborah Kurata
5:30 PM	7:00 PM	Microsoft Ask the Experts & Exhibitor Reception	
START TIME	END TIME	Visual Studio Live! Day 2: Wednesday, August 12, 2015	
8:00 AM	9:00 AM	Keynote: Details Coming Soon	
9:15 AM	10:30 AM	W01 – iOS Development - What They Don't Tell You – Jon Flanders	W02 – Implementing M-V-VM (Model-View-View Model) for WPF – Philip Japikse
10:45 AM	12:00 PM	W06 – Building Cross Platform UI with Xamarin.Forms – Walt Ritscher	W07 – To Be Announced
12:00 PM	1:30 PM	Birds-of-a-Feather Lunch – Visit Exhibitors	
1:30 PM	2:45 PM	W11 – Swift for .NET Developers – Jon Flanders	W12 – Strike Up a Conversation with Cortana on Windows Phone – Walt Ritscher
2:45 PM	3:15 PM	Sponsored Break – Exhibitor Raffle @ 2:55 pm (Must be present to win)	
3:15 PM	4:30 PM	W16 – Creating Applications Using Android Studio – Kevin Ford	W17 – Securing Angular Apps – Brian Noyes
8:00 PM	10:00 PM	Lucky Strike Evening Out Party	
START TIME	END TIME	Visual Studio Live! Day 3: Thursday, August 13, 2015	
8:00 AM	9:15 AM	TH01 – Using Multi-Device Hybrid Apps to Create Cordova Applications – Kevin Ford	TH02 – Build Data-Centric HTML5 Single Page Applications with Breeze – Brian Noyes
9:30 AM	10:45 AM	TH06 – Everything You Always Wanted To Know About REST (But Were Afraid To Ask) – Jon Flanders	TH07 – Build Real-Time Websites and Apps with SignalR – Rachel Appel
11:00 AM	12:15 PM	TH11 – To Be Announced	TH12 – Knocking it Out of the Park with Knockout.JS – Miguel Castro
12:15 PM	2:15 PM	Lunch @ The Mixer – Visit the Microsoft Company Store & Visitor Center	
2:15 PM	3:30 PM	TH16 – Extending XAML to Overcome Pretty Much any Limitation – Miguel Castro	TH17 – ASP.NET MVC: All Your Tests Are Belong To Us – Rachel Appel
3:45 PM	5:00 PM	TH21 – Agile Database Development – Richard Hundhausen	TH22 – Busy JavaScript Developer's Guide to ECMAScript 6 – Ted Neward
START TIME	END TIME	Visual Studio Live! Post-Conference Workshops: Friday, August 14, 2015	
8:00 AM	12:00 PM	F01 – Workshop: SQL Server 2014 for Developers – Andrew Brust and Leonard Label	
12:00 PM	1:00 PM	Lunch	
1:00 PM	5:00 PM	F01 – Workshop Continues	

Sessions and speakers subject to change.

DETAILS COMING SOON!

With all of the announcements coming out of Build and Ignite, we'll be finalizing the Redmond Keynotes and the **FULL Track** of Microsoft-led sessions shortly.



NAVIGATE THE .NET HIGHWAY

Design	Mobile Client	Visual Studio / .NET	Web Development	Windows Client	Microsoft Sessions
(Separate entry fee required)					
M02 – Workshop: UX Design Process Essentials– Steps and Techniques – <i>Billy Hollis</i>			M03 – Workshop: ALM and DevOps with the Microsoft Stack – <i>Brian Randell</i>		
M02 – Workshop Continues			M03 – Workshop Continues		
T03 – UX Design Principle Fundamentals for Non-Designers – <i>Billy Hollis</i>		T04 – A Lap Around Visual Studio 2015 – <i>Robert Green</i>		 T05 – Microsoft Session, Details Coming Soon	
T08 – Designing and Building UX for Finding and Visualizing Data in XAML Applications – <i>Billy Hollis</i>		T09 – What’s New in ALM – <i>Brian Randell</i>		 T10 – Microsoft Session, Details Coming Soon	
T13 – Windows, NUI and You – <i>Brian Randell</i>		T14 – Hosting ASP.NET Applications Cross Platform with Docker – <i>Adam Tuliper</i>		 T15 - Microsoft Session, Details Coming Soon	
T18 – Building Windows 10 LOB Apps – <i>Robert Green</i>		T19 – SOLID Design Patterns for Mere Mortals – <i>Philip Japikse</i>		 T20 – Microsoft Session, Details Coming Soon	
W03 – Moving Web Apps to the Cloud – <i>Eric D. Boyd</i>		W04 – Enhancing Application Quality Using Visual Studio 2015 Premium Features – <i>Anthony Borton</i>		 W05 – Microsoft Session, Details Coming Soon	
W08 – Solving Security and Compliance Challenges with Hybrid Clouds – <i>Eric D. Boyd</i>		W09 – Load Testing ASP.NET & WebAPI with Visual Studio – <i>Benjamin Day</i>		 W10 – Microsoft Session, Details Coming Soon	
W13 – To Be Announced		W14 – To Git or Not to Git for Enterprise Development – <i>Benjamin Day</i>		 W15 – Microsoft Session, Details Coming Soon	
W18 – Building for the Internet of Things: Hardware, Sensors & the Cloud – <i>Nick Landry</i>		W19 – Not Your Grandfather’s Build – A Look at How Build Has Changed in 2015 – <i>Anthony Borton</i>		 W20 – Microsoft Session, Details Coming Soon	
TH03 – Database Development with SQL Server Data Tools – <i>Leonard Lobel</i>		TH04 – Stop the Waste and Get Out of (Technical) Debt – <i>Richard Hundhausen</i>		 TH05 – Microsoft Session, Details Coming Soon	
TH08 – Programming the T-SQL Enhancements in SQL Server 2012 – <i>Leonard Lobel</i>		TH09 – What’s New in C# 6.0 – <i>Jason Bock</i>		 TH10 – Microsoft Session, Details Coming Soon	
TH13 – Power BI 2.0: Analytics in the Cloud and in Excel – <i>Andrew Brust</i>		TH14 – Async and Await Best Practices – <i>Mark Rosenberg</i>		 TH15 – Microsoft Session, Details Coming Soon	
TH18 – Busy Developer’s Guide to NoSQL – <i>Ted Neward</i>		TH19 – Microsoft’s .NET is Now Open Source and Cross-Platform. Why it Matters. – <i>Mark Rosenberg</i>		 TH20 – Microsoft Session, Details Coming Soon	
TH23 – Big Data and Hadoop with Azure HDInsight – <i>Andrew Brust</i>		TH24 – Managing the .NET Compiler – <i>Jason Bock</i>		 TH25 – Microsoft Session, Details Coming Soon	
(Separate entry fee required)					
F02 – Workshop: Service Oriented Technologies: Designing, Developing, & Implementing WCF and the Web API – <i>Miguel Castro</i>					
F02 – Workshop Continues					

CONNECT WITH VISUAL STUDIO LIVE!



vslive.com/redmond

Event Hubs for Analytics and Visualization, Part 2

Bruno Terkaly

When using Microsoft Azure Event Hubs to process massive amounts of data, it's important to focus on both the delivery and consumption side. This is part two of a three-part series of articles about data visualization for Internet of Things (IoT) scenarios. In the April issue (msdn.microsoft.com/magazine/dn948106), I discussed ingesting data at massive scale using Azure Event Hubs, a technology that's part of the Service Bus offering and supports information publishers (Raspberry Pi) and information consumers (C# Program) at scale. In this article, I'll focus on the consumption side, C# code that reads from Event Hubs and persists data in a permanent store.

This article is broken into four pillars. I'll start with the architecture and illustrate message consumption and storage. Second, I'll discuss the tasks to complete at the Azure Portal. Third, I'll describe the two different data stores I'll use and explain why I chose them. And finally, I'll conclude with an explanation of the client C# code that reads the messages from Event Hubs, as well as the code that stores the messages in Azure SQL Database and DocumentDB.

Event Hubs Architecture

After reading messages from Azure Event Hubs, you'll want to persist those messages to both Azure SQL Database and DocumentDB. Azure SQL Database is a relational Database as a Service (DaaS) that's scalable to thousands of databases. DocumentDB is a fully managed, scalable, NoSQL document database service (read more about DocumentDB at bit.ly/1oWZP7i). This architecture is displayed in **Figure 1**.

This article discusses:

- Consuming data with Event Hubs
- Using different data stores
- Creating, storing and querying messages

Technologies discussed:

Microsoft Azure, Azure Event Hubs, Visual Studio 2013

Code download available at:

bit.ly/1DF4WdP

Tasks to Complete

To make the solution come to life, I'll need to accomplish several tasks at the portal. There are two categories of tasks to perform at the Azure Portal. The first category is provisioning for Event Hubs, Azure SQL Database and DocumentDB. I provisioned the Event Hubs in the April article.

In order to read messages from Event Hubs and then persist those messages into Azure SQL Database and DocumentDB, I'll need to obtain connection information from the Azure Portal. All the settings will be placed in the App.config in the Visual Studio solution for the C# application. I'll create an ordinary console application using the Event Hubs SDK.

Finally, to assist with database operations, I'll write a couple of stored procedures for Azure SQL Database. You'll find the code for creating the table, as well as the two stored procedures, in DatabaseCode.txt as part of the Visual Studio solution. You can use SQL Server Management Studio to build out the database table and the stored procedures. Use the connection information from the Azure Portal to attach the Azure SQL Database sitting in the Microsoft datacenter to your local copy of Visual Studio. For more information on how to do this, read the Azure Documentation on it at bit.ly/1K1BleM.

In April, I illustrated the provisioning process for Service Bus Event Hubs. To walk through some simple tutorials, check out "Get Started with Event Hubs" at bit.ly/1F2gp9H.

Once you've provisioned Event Hubs, you'll receive an endpoint. You'll need to copy the connection information for this and the point that will be placed into App.config when writing the C# code. You'll also need to provision an Azure SQL Database at the Azure Portal. For more on this process read the documentation at bit.ly/1Enln5c.

Once you've created the database, there are three more easy tasks. The first task is to create the temperatures table. The two remaining tasks have to do with two stored procedures you'll use to clear out old messages and insert new messages. In the Visual Studio project, there's a text file called DatabaseCode.txt with the table definition for temperatures, as well as the two stored procedures, CleanTable and InsertTempData.

Now that you've taken care of Event Hubs and the Azure SQL Database components, turn your attention to DocumentDB. You can learn more about provisioning this NoSQL data store at bit.ly/1BnGQ5. You can also watch the video from Ryan CrowCour, one of the senior program managers on the team who has shepherded the product since inception.

Once you've provisioned DocumentDB at the portal, you'll need to define a collection name and a database name. The way to think about DocumentDB is as a series of collections, and a collection as a series of documents. The analogy in the database world is the collection is the table and a document is the record. This is a loosely defined analogy because NoSQL data stores are generally schema-less. The collection name used in our application is called CityTempCollection and the database is called TemperatureDB.

Understand Data Storage Options

The starting point for this discussion is to establish the need for secondary data stores beyond what Event Hubs can natively provide. For example, one obvious reason to think about persistence is messages stored in Event Hubs aren't permanent. Second, you can't query messages with any sort of query language. You can only read them serially from Event Hub partitions, where each client reader maintains its own cursor in the message store. Third, Event Hubs gives you 84GB of event storage for the default 24-hour retention period.

Although there are many options, I've chosen Azure SQL Database and DocumentDB as the primary data stores for temperature-related messages/data. The advantages to Azure SQL Database are two-fold. The first relates to the fact that it's cloud-hosted, which provides several other valuable attributes. It's almost instantaneous to provision, it's economical and it's triple-replicated, making it fairly robust.

Aside from those obvious benefits, there's also a lot of business intelligence tooling built around Azure SQL Database, such as Power BI. This lets you build interactive reports with sophisticated visualization. You can build powerful dashboards that you can then consume from a browser and a mobile device.

Use DocumentDB as the second of two data stores. This has some clear advantages. It's a fully managed service, so you don't have to bother with infrastructure and individual virtual machines (VMs).

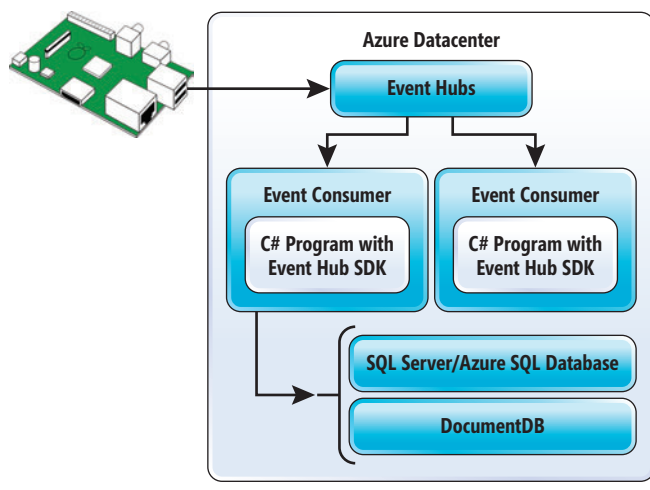


Figure 1 Overview of Azure Event Hubs Architecture

Also, it comes with an enterprise service-level agreement, which is the case with most Azure-based services. Finally, DocumentDB is also schema-free.

Schema-free document stores are often considered desirable in the context of object orientation and inheritance. That's because inheritance means you have objects with some attributes in common, but also some attributes specific to an object subtype.

With most relational databases, you would need a table for all possible attributes, leaving many of them null for fields that don't apply. In a schema-less database, however, you can store different sets of optional properties. This works well when rendering HTML because JavaScript can check for an unknown optional property and call the appropriate function to output to a table for display.

One of the other advantages (which can also be seen as a disadvantage for some) for a schema-less database is it provides additional agility during development. You can add new features without restructuring the database. This makes it easy to maintain backward compatibility to data created by a previous version. The downside of this approach is writing queries against optional fields can become complex and convoluted.

One of the areas where a JSON-based data store like DocumentDB really shines is when using Node.js applications to expose the data to mobile or Web applications. The native JSON format, which is compact and expressive, makes it easy and natural with which to work. Consuming JSON-based data stores with HTTP and REST is also straightforward whether you're using the Microsoft .NET Framework, JavaScript, Ruby, Java, PHP or Python. I'll focus on Node.js applications reading DocumentDB data and exposing the data to mobile applications in the next installment.

Consume and Persist Messages

There are three common approaches to consuming messages from Event Hubs. As always, there's a trade-off between what's easy and what's flexible. The easy way is with Stream Analytics. Another way is with direct receivers. Direct receivers are responsible for their own coordination of access to partitions within a consumer group, whereby your code directly addresses a partition ID when creating the receiver for the EventHubConsumerGroup object. Another way is with higher-level abstractions, such as EventProcessorHost. I'll use this method because it's simple enough, yet provides enough flexibility.

Stream Analytics is a fully managed service that makes it easy to ingest messages from Event Hubs. It's easy to consume, transform and place those messages into an Azure SQL Database. Read more about this approach at bit.ly/1IRvPDc.

It's a simple matter of creating a database, table and a series of queries. For example, you use a simple select statement to read messages from Event Hubs, "SELECT DeviceId, Temperature FROM input," and place those messages into SQL Server. You can even chain queries recursively and create a series of queries that transform the data using a pipeline approach. This ability to filter messages through a series of SQL queries is a powerful abstraction.

One of the more interesting extensions available through Stream Analytics has to do with windowing. There's often the requirement to perform some set-based computation (aggregation) or other

Figure 2 Code to Consume and Persist Messages

```

+-----+
| Section 1 |
+-----+
internal class Program
{
    private static string eventHubConnectionString =
        ConfigurationManager.AppSettings["eventHubConnectionString"];
    static private string eventHubName =
        ConfigurationManager.AppSettings["eventHubName"];
    static private string storageAccountName =
        ConfigurationManager.AppSettings["storageAccountName"];
    static private string storageAccountKey =
        ConfigurationManager.AppSettings["storageAccountKey"];
    static private string storageConnectionString =
        string.Format("DefaultEndpointsProtocol=https;" +
            "AccountName={0};AccountKey={1}",
            storageAccountName, storageAccountKey);
    static private string eventProcessorHostName = Guid.NewGuid().ToString();

    private static void Main(string[] args)
    {
        // Start the main message consumption engine
+-----+
| Section 2 |
+-----+
        var eventProcessorHost =
            new EventProcessorHost(eventProcessorHostName,
                eventHubName, EventHubConsumerGroup.DefaultGroupName,
                eventHubConnectionString, storageConnectionString);

        // Asynchronously consume message from Event Hub
        eventProcessorHost.
            RegisterEventProcessorAsync<SimpleEventProcessor>().Wait();

        Console.WriteLine("Receiving. Press enter key to stop worker.");
        Console.ReadLine();
    }
}
+-----+
| Section 3 |
+-----+

// SimpleEventProcessor.cs

class SimpleEventProcessor : IEventProcessor
{
    private DataManager dataManager = new DataManager(new SQLiteDatabaseManager());
    // ... Means omitted for brevity
    public SimpleEventProcessor() ...
    async Task IEventProcessor.CloseAsync(PartitionContext context,
        CloseReason reason) ...
    Task IEventProcessor.OpenAsync(PartitionContext context) ...

    async Task IEventProcessor.ProcessEventsAsync(PartitionContext context,
        IEnumerable<EventData> messages)
    {
        // Loop through messages to insert
        foreach (EventData eventData in messages)
        {
            string data = Encoding.UTF8.GetString(eventData.GetBytes());
            // Comma separated so divide up fields
            string[] msg = data.Split(',');
            if (msg.Length > 2)
            {
+-----+
| Section 4 |
+-----+
                // Insert into SQL
                dataManager.InsertSqlMessage(msg[0], Convert.ToInt32(msg[1]),
                    Convert.ToDouble(msg[2]));

                // Insert into global DocumentDB object
                // (global because of thread timing issues)
                Globals.DocDb.InsertEntry(msg[0], Convert.ToInt32(msg[1]),
                    Convert.ToDouble(msg[2]));
            }

            Console.WriteLine(string.Format("Message received. Partition: '{0}', " +
                "Data: '{1}'", context.Lease.PartitionId, data));
        }
    }

    // Call checkpoint every 5 minutes, so that worker can resume
    // processing from the 5 minutes back if it restarts
    if (this.checkpointStopWatch.Elapsed > TimeSpan.FromMinutes(5))
    {
        await context.CheckpointAsync();
        lock (this)
        {
            this.checkpointStopWatch.Reset();
        }
    }
}
+-----+
| Section 5 |
+-----+

// DocumentDBManager.cs
public class DocumentDbManager
{
    // Omitted variables for brevity
    public string collectionName = "CityTempCollection";
    public string databaseName = "TemperatureDB";

    public async Task<bool> InsertEntry(
        string city, int month, double temperature)
    {
        dynamic document = null;
        try
        {
+-----+
| Section 6 |
+-----+
            // Check if City exists
            document = client.CreateDocumentQuery(documentCollection.DocumentsLink)
                .Where(d => d.Id == city).AsEnumerable().FirstOrDefault();

        }
        catch (Exception ex)
        {
            throw;
        }

        bool docExists = (document != null);

        // Document DOESN'T exist, yet
        if (!docExists)
        {
+-----+
| Section 7 |
+-----+
            var cityMonthTemperature = new CityMonthTemperature
            {
                Id = city,
                City = city
            };
            cityMonthTemperature.Temperatures[month - 1] = temperature;

            try
            {
+-----+
| Section 8 |
+-----+
                // Create, and set document to the return, yes --
                // here is where you reset the document object
                document = await client.CreateDocumentAsync(
                    documentCollection.DocumentsLink, cityMonthTemperature);
            }
            catch (DocumentClientException ex)
            // Omitted for brevity
        }
    }
}

```


AWSInsider.net

Your independent resource for news, how-tos, tips and more on the Amazon Web Services cloud platform.



operations over subsets of events that fall within some period of time. Because time is crucial in complex event-processing systems, it's important to have a simple way to work with the time component of query logic in the system. In Azure Stream Analytics, these subsets of events are defined through windows to represent groupings by time. For a detailed explanation of the various types of supported time windows, check out bit.ly/1DlizzfM.

Direct receivers let you target specific Event Hub partition IDs. Partitions provide value in two ways. First, they let you scale publishing and consuming messages. Second, they also let you segregate data into separate silos.

I'll take the EventProcessorHost approach, which is an intelligent agent for .NET consumers that manages partition access and per partition offset for consumers. For a more detailed explanation, please read the ServiceBus Blog post at bit.ly/1a05l19. Use EventProcessorHost to make it easy to read messages, transform them and write them to permanent storage.

I'll write a custom C# program that leverages the Event Hubs SDK and write to both Azure SQL Database and DocumentDB (check out the Event Hubs Programming Guide at bit.ly/1lBrpNz). You can get all the source code for this at bit.ly/1aSFf99. I've stripped the password and secret keys, but you can get the connection information from the portal. To get all the code, install Git and run the issuing command: `git clone`. Some of the files you'll find in the Visual Studio console application are shown in **Figure 2**. The main driver code can be found in Program.cs.

Section 1 should come as no surprise. It retrieves some of the connection information from App.config. You can get all the connection information can be obtained from the portal. Section 2 is where you'll actually instantiate EventProcessorHost, the core object in the Event Hubs SDK that lets you retrieve messages. You'll find the underlying events tied to this object in Section 3, within the class SimpleEventProcessor.

The asynchronous callback, ProcessEventsAsync, is called by the SDK. This passes the parameter `IEnumerable<EventData>` messages, which you then parse to retrieve the messages stored in Event Hubs. Then parse the messages parameter, and insert the parsed message into SQL Database and DocumentDB in Section 4. You'll find all the low-level code details for `InsertSqlMessage` and `InsertEntry` in the Visual Studio solution.

Section 5 represents the class that actually does the insert operation for DocumentDB. The database name is TemperatureDB and collection name is CityTempCollection. Section 6 represents a query wherein you search for a city using a LINQ query. The logic here is a city may have been previously inserted. What you really want to do is update temperature data if the city exists.

Section 7 represents the scenario where the city hasn't been added. You create a simple .NET object that's transformed into JSON data once the insert takes place. The underlying SDK takes care of this transformation. You insert the temperature into the

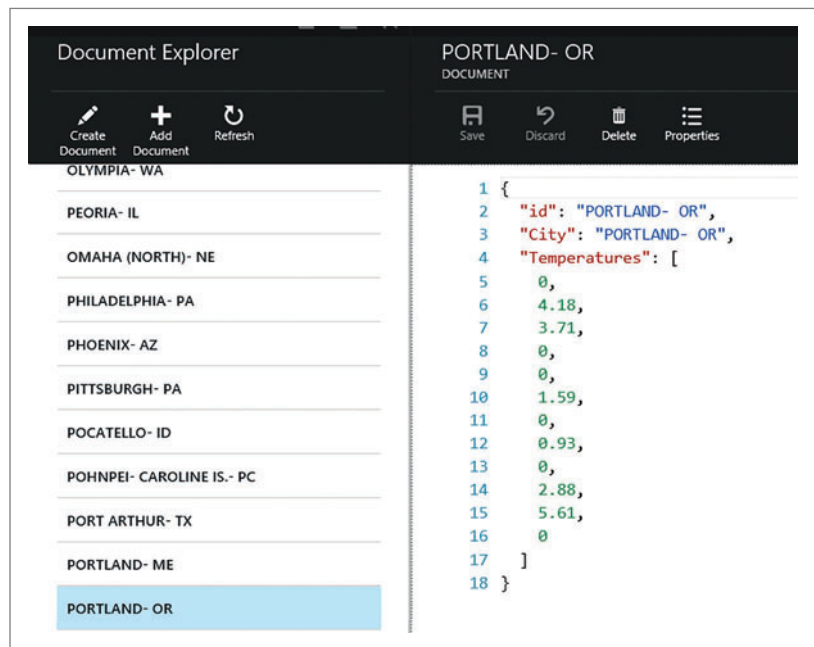


Figure 3 Use DocumentDB to Create, Query and View Data

appropriate month offset of the temperatures array. Finally, in Section 8, you actually update the document object.

The code to perform just an update of the temperature in the scenario where the city has already been inserted was omitted from this code snippet for brevity purposes, but you can find it in the GitHub repository, along with the entire Visual Studio solution, at bit.ly/1aSFf99.

Because DocumentDB is part of a preview offering, you need to use the new Azure Preview Portal at portal.azure.com. One of the nice features of DocumentDB is that there's a Document Explorer that lets you create documents and data, as well as query data and view existing data, as shown in **Figure 3**.

In the April installment, I created a C program running in Linux to leverage the AMQP transport protocol to insert messages into Event Hubs. That code ran in an Azure-hosted Linux VM so I could easily port it into a Debian-based Raspberry Pi implementation. In short, the first article was about producing message publishing. This installment has been all about message consumption and persistent storage of messages. And in the final installment, I'll address the ability to expose the persistent data to mobile clients and provide visualization of the underlying data. ■

BRUNO TERKALY is a principal software engineer at Microsoft with the objective of enabling development of industry-leading applications and services across devices. He's responsible for driving the top cloud and mobile opportunities across the United States and beyond from a technology-enablement perspective. He helps partners bring their applications to market by providing architectural guidance and deep technical engagement during the ISV's evaluation, development and deployment. Terkaly also works closely with the cloud and mobile engineering groups, providing feedback and influencing the roadmap.

THANKS to the following Microsoft technical experts for reviewing this article: Ryan CrawCour, Dan Rosanova



YOUR .NET Resources

msdn
magazine

Visual Studio
Magazine

Visual Studio **LIVE!**
EXPERT SOLUTIONS FOR .NET DEVELOPERS

ONLINE | NEWSLETTERS | PRINT | CONFERENCES



Firefly Algorithm Optimization

In machine learning, a numerical optimization algorithm is often used to find a set of values for variables—usually called weights—that minimize some measure of error. For example, logistic regression classification uses a math equation where, if there are n predictor variables, there are $n+1$ weight values that must be determined. The process of determining the values of the weights is called training the model. The idea is to use a collection of training data that has known correct output values. An optimization algorithm is used to find the values of the weights that minimize the error, which is the difference between computed output values and correct output values.

There are many different optimization algorithms. This article explains a relatively new (first published in 2009) technique called firefly algorithm (FA) optimization. FA optimization loosely models the behavior of a swarm of fireflies.

A good way to get an idea of what FA optimization is and to see where this article is headed is to take a look at the demo program in **Figure 1**. The goal of the demo program is to use FA optimization to find the minimum value of the Michalewicz function with five input variables. The Michalewicz function is a standard benchmark function used to evaluate the effectiveness of numerical optimization algorithms. With five input values, the function has a minimum value of $z = -4.6877$ located at $x = (2.2029, 1.5707, 1.2850, 1.9231, 1.7205)$.

The Michalewicz function is difficult for most optimization algorithms because it has several local minimum values and several flat areas (where all z values are almost equal). It's not possible to easily visualize the Michalewicz function with five input values, but you can get an idea of the function's characteristics by examining a graph of the function for two input values, shown in **Figure 2**. The function's definition in math terms is shown at the bottom of the figure.

The demo program sets the number of fireflies to 40. Each firefly has a virtual position that represents a possible solution to the minimization problem. More fireflies increase the chance of finding the true optimal solution at the expense of performance. FA optimization typically uses 15 to 40 fireflies.

The demo sets the problem dimension to 5 because there are five input values. FA is an iterative process and requires a maximum loop counter value. A loop counter variable in machine learning optimization is often named epoch and the demo sets the maximum

value to 1,000 iterations. The maximum number of iterations will vary from problem to problem, but 1,000 is a reasonable starting value. FA has an element of randomness and the demo sets the seed value for the random number generator to an arbitrary value of 0.

In the demo run in **Figure 1**, the best (smallest) error associated with the best position found so far was displayed every 100 epochs. After the algorithm finished, the best position found for any firefly was $x = (2.2033, 1.5711, 1.2793, 1.1134, 2.2216)$. This solution is close to, but not quite equal to, the optimal solution of $x = (2.2029, 1.5707, 1.2850, 1.9231, 1.7205)$. The value of the Michalewicz function at the solution found by FA was -4.45 , which is close to the true minimum value of -4.69 . The error of the FA solution is 0.0561.

This article assumes you have at least intermediate programming skills, but does not assume you know anything about numerical optimization or the firefly algorithm. The demo program is coded using C# but you shouldn't have too much difficulty refactoring the code to another language, such as JavaScript or Python.

More fireflies increase the chance of finding the true optimal solution at the expense of performance.

The complete demo code, with a few minor edits to save space, is presented in this article. The demo is also available in the code download that accompanies this article. The demo code has all normal error checking removed to keep the main ideas as clear as possible and the size of the code small.

Overall Program Structure

The overall program structure is presented in **Figure 3**. To create the demo, I launched Visual Studio and created a new C# console application named FireflyAlgorithm. The demo has no significant Microsoft .NET Framework dependencies, so any recent version of Visual Studio will work.

After the template code loaded into the Visual Studio editor, in the Solution Explorer window I renamed file Program.cs to the more descriptive FireflyProgram.cs and Visual Studio automatically

Code download available at msdn.microsoft.com/magazine/msdnmag0615.

```

file:///C:/FireflyAlgorithm/bin/Debug/FireflyAlgorithm... - [X]
Begin firefly algorithm optimization demo
Goal is to solve the Michalewicz benchmark function
The function has a known minimum value of -4.687658
x = 2.2029 1.5707 1.2850 1.9231 1.7205

Setting numFireflies      = 40
Setting problem dim       = 5
Setting maxEpochs        = 1000
Setting initialization seed = 0

Starting firefly algorithm

epoch =    0   error = 8.09399774623980
epoch =   100   error = 1.08059893484352
epoch =   200   error = 0.23030175299833
epoch =   300   error = 0.06083932613045
epoch =   400   error = 0.06083932613045
epoch =   500   error = 0.06083932613045
epoch =   600   error = 0.06062567496022
epoch =   700   error = 0.05610057924469
epoch =   800   error = 0.05610057924469
epoch =   900   error = 0.05610057924469

Finished

Best solution found:
x = 2.2033 1.5711 1.2793 1.1134 2.2216
Value of function at best position = -4.450802
Error at best position = 0.0561

End firefly demo

```

Figure 1 The Firefly Algorithm Optimization in Action

renamed class Program for me. At the top of the source code, I deleted all unnecessary using statements, leaving just the reference to System.

I coded the demo using a mostly static-method technique rather than using a full object-oriented programming approach. The demo has all the control logic in Main. The Main method begins by displaying the purpose of the demo:

```

Console.WriteLine("Goal is to solve the Michalewicz benchmark function");
Console.WriteLine("The function has a known minimum value of -4.687658");
Console.WriteLine("x = 2.2029 1.5707 1.2850 1.9231 1.7205");

```

Next, the parameters needed by FA are set:

```

int numFireflies = 40;
int dim = 5;
int maxEpochs = 1000;
int seed = 0;

```

The parameter values are displayed with these statements:

```

Console.WriteLine("Setting numFireflies = " + numFireflies);
Console.WriteLine("Setting problem dim = " + dim);
Console.WriteLine("Setting maxEpochs = " + maxEpochs);
Console.WriteLine("Setting initialization seed = " + seed);

```

The firefly algorithm is invoked like so:

```

Console.WriteLine("Starting firefly algorithm");
double[] bestPosition = Solve(numFireflies, dim, seed, maxEpochs);
Console.WriteLine("Finished");

```

The Main method concludes by displaying the FA results:

```

Console.WriteLine("Best solution found: ");
Console.WriteLine("x = ");
ShowVector(bestPosition, 4, true);
double z = Michalewicz(bestPosition);
Console.WriteLine("Value of function at best position = ");
Console.WriteLine(z.ToString("F6"));
double error = Error(bestPosition);
Console.WriteLine("Error at best position = ");
Console.WriteLine(error.ToString("F4"));

```

The firefly algorithm is really more of a meta-heuristic than a prescriptive algorithm. By that I mean FA is a set of design guidelines that can be adapted for many different types of optimization problems.

Understanding the Firefly Algorithm

The firefly algorithm presented in this article is based on the 2009 research paper, "Firefly Algorithms for Multimodal Optimization," by Xin-She Yang. The firefly algorithm process is illustrated in the graph in **Figure 4**. The graph represents a simplified dummy minimization problem in which there are just two input values, X and Y, and the global minimum value is at X = 0 and Y = 0. There are three fireflies. Firefly[0] is at (2, 1) and so is the closest to the correct solution. Firefly[1] is at (-4, -4). Firefly[2] is at (-8, 8) and is the farthest from the solution.

Real fireflies are flying insects that glow using bioluminescence, presumably to attract mates. Each firefly can glow with a different intensity. In FA, fireflies that are better, meaning a smaller error, have higher intensity. In **Figure 4**, then, firefly[0] has the highest intensity, firefly[1] has intermediate intensity, and firefly[2] has weak intensity.

The basic idea of FA is that a firefly will be attracted to any other firefly that has a higher intensity, and that attractiveness (the distance moved toward a more intense firefly) is stronger if the distance between the two fireflies is smaller. So, in **Figure 4**, firefly[0] has the highest intensity and will not move. Firefly[1] and firefly[2] will both be attracted to and move toward firefly[0]. Because firefly[1] is closer than firefly[2]

to firefly[0], firefly[1] will move a greater distance than firefly[2].

Expressed in very high-level pseudo-code, the firefly algorithm is presented in **Figure 5**. At first glance, the algorithm seems very simple; however, it's quite subtle, as you'll see when the code implementation is presented.

The first major issue is to define the intensity of a firefly. Because FA is a meta-heuristic, you are free to define intensity however you like, as long as a higher intensity is associated with a better solution/position. The next major issue is to define attraction so that closer fireflies will move toward a more-intense target more than distant fireflies will move.

The firefly algorithm is really more of a meta-heuristic than a prescriptive algorithm.

Implementing the Firefly Algorithm

The definition of method Solve begins as:

```

static double[] Solve(int numFireflies, int dim, int seed, int maxEpochs)
{
    Random rnd = new Random(seed);
    double minX = 0.0;
    double maxX = 3.2;
    double B0 = 1.0;
    double g = 1.0;
    double a = 0.20;
    int displayInterval = maxEpochs / 10;
    ...
}

```

Local variables minX and maxX establish boundaries for each firefly's position. The values used here, 0.0 and 3.2 (approximately Pi) are

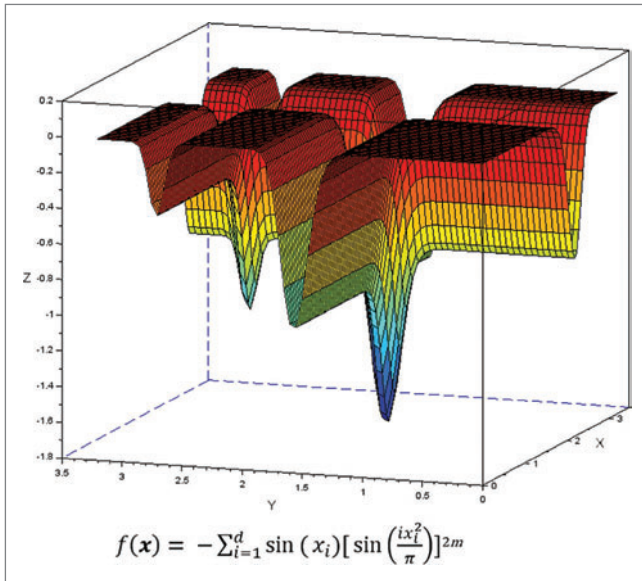


Figure 2 The Michalewicz Function of Two Variables

specific to the Michalewicz function. For machine learning optimization with normalized data, values of -10.0 and +10.0 are common.

Local variables B0 (base beta), g (gamma) and a (alpha) control the attractiveness of one firefly to another. The values used (1.0, 1.0, and 0.20) were recommended by the source research paper. Local variable displayInterval controls how often to display a progress message.

Next, an empty swarm of fireflies is created:

```
double bestError = double.MaxValue;
double[] bestPosition = new double[dim]; // Best ever
Firefly[] swarm = new Firefly[numFireflies]; // All null
```

A Firefly object is program-defined and encapsulates a position, an associated error and the corresponding intensity. Initially, all fireflies are null objects. The Firefly class definition will be presented in the next section of this article. Next, the swarm is instantiated and placed at random positions. For each firefly, the Firefly constructor is called:

```
for (int i = 0; i < numFireflies; ++i)
{
    swarm[i] = new Firefly(dim);
    for (int k = 0; k < dim; ++k) // Random position
        swarm[i].position[k] = (maxX - minX) * rnd.NextDouble() + minX;
    ...
}
```

A Firefly object is program-defined and encapsulates a position, an associated error and the corresponding intensity.

The constructor implicitly sets the position to (0.0, 0.0, 0.0, 0.0, 0.0) and the associated error and intensity to dummy values of 0.0. Then each component of the position array is set to a random value between minX and maxX (0.0 and 3.2). Next, the current firefly's error and intensity are calculated:

```
swarm[i].error = Error(swarm[i].position);
swarm[i].intensity = 1 / (swarm[i].error + 1);
...
```

The Error function will be presented shortly. Here, the intensity of the firefly is defined to be the inverse of the error so that small error values will have high intensity and large error values will have low intensity. The 1 in the denominator prevents division by zero when error is zero. Initialization concludes by checking the newly created firefly to see if it has the best position found:

```
...
if (swarm[i].error < bestError)
{
    bestError = swarm[i].error;
    for (int k = 0; k < dim; ++k)
        bestPosition[k] = swarm[i].position[k];
}
} // For each firefly
```

The main processing loop begins with these statements:

```
int epoch = 0;
while (epoch < maxEpochs)
{
    if (epoch % displayInterval == 0 && epoch < maxEpochs)
    {
        string sEpoch = epoch.ToString().PadLeft(6);
        Console.WriteLine("epoch = " + sEpoch);
        Console.WriteLine(" error = " + bestError.ToString("F14"));
    }
    ...
}
```

An alternative to a fixed number of iterations is to break when the value of bestError drops below some small threshold value (0.00001 is common). Each firefly is compared with all other fireflies using nested for loops:

```
for (int i = 0; i < numFireflies; ++i) // Each firefly
{
    for (int j = 0; j < numFireflies; ++j) // Others
    {
        if (swarm[i].intensity < swarm[j].intensity)
        {
            // Move firefly(i) toward firefly(j)
        }
    }
    ...
}
```

Figure 3 Firefly Demo Program Structure

```
using System;
namespace FireflyAlgorithm
{
    class FireflyProgram
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Begin firefly demo");
            // Code here
            Console.WriteLine("End firefly demo");
            Console.ReadLine();
        }

        static void ShowVector(double[] v, int dec, bool nl)
        {
            for (int i = 0; i < v.Length; ++i)
                Console.Write(v[i].ToString("F" + dec) + " ");
            if (nl == true)
                Console.WriteLine("");
        }

        static double[] Solve(int numFireflies, int dim,
            int seed, int maxEpochs) { . . . }

        static double Distance(double[] posA,
            double[] posB) { . . . }

        static double Michalewicz(double[] xValues) { . . . }

        static double Error(double[] xValues) { . . . }
    } // Program

    public class Firefly : IComparable<Firefly>
    {
        // Defined here
    }
}
```


TECHMENTOR

IN-DEPTH TRAINING FOR IT PROS

MICROSOFT HEADQUARTERS,
REDMOND, WA

AUGUST 3 - 7, 2015

ENGINEERED FOR YOU @ THE SOURCE

Join us August 3 – 7, 2015 for TechMentor 2015:
Datacenter Edition, focused entirely on making your datacenter
more modern, capable, and manageable through 5 days of
immediately usable IT education.

THE AGENDA FEATURES:

- ✦ 75-minute topic overview
breakout sessions
- ✦ 3 hour content-rich deep dives
- ✦ "Hands on" labs with your
own laptop

CONTENT AREAS INCLUDE:

- ✦ Windows PowerShell
- ✦ Infrastructure Foundations
- ✦ Runbook Automation
- ✦ Configuration and
Service Management
- ✦ Datacenter Provisioning
and Deployment

SAVE \$300!

WHEN YOU REGISTER BY JULY 1

USE PROMO CODE **TMJUN1**



Scan the QR code to register
or for more event details.

EVENT SPONSOR:



PLATINUM SPONSOR:



GOLD SPONSORS:



SUPPORTED BY:



TECHMENTOREVENTS.COM/REDMOND

PRODUCED BY:  1105 MEDIA

Notice that because each for loop index starts at 0, each pair of fireflies is compared twice in each iteration of the while loop. In order to move a firefly toward another firefly with higher intensity, first the attractiveness must be calculated:

```
double r = Distance(swarm[i].position, swarm[j].position);
double beta = B0 * Math.Exp(-g * r * r);
...
```

Variable beta defines attraction and will be used in a moment to move firefly[i]. Its value depends on the square of the distance between fireflies [i] and [j], which is calculated using helper method Distance. Method Distance returns the Euclidean distance between two positions. For example, if firefly[i] in two dimensions is at (3.0, 4.0) and firefly[j] is at (5.0, 9.0), the distance between them is $\sqrt{(5 - 3)^2 + (9 - 4)^2} = \sqrt{4 + 25} = \sqrt{29} = 5.4$. Notice that beta uses squared distance, which is the inverse of the square root operation, so the calculation of beta could be simplified, at the expense of flexibility, if you decided to use a different measure of distance.

The actual movement is accomplished with these statements:

```
for (int k = 0; k < dim; ++k)
{
    swarm[i].position[k] += beta *
        (swarm[j].position[k] - swarm[i].position[k]);
    swarm[i].position[k] += a * (rnd.NextDouble() - 0.5);
    if (swarm[i].position[k] < minX)
        swarm[i].position[k] = (maxX - minX) * rnd.NextDouble() + minX;
    if (swarm[i].position[k] > maxX)
        swarm[i].position[k] = (maxX - minX) * rnd.NextDouble() + minX;
}
...
```

The kth component of the position of firefly[i] is moved a beta-fraction of the distance between firefly[i] and firefly[j] toward firefly[j]. Then a small random term is added to each kth position component. This helps prevent the algorithm from getting stuck in non-optimal solutions. Each position component is checked to see if it went out of range, and if so, a random in-range value is assigned.

The nested loops movement code finishes by updating the error and intensity of the just-moved firefly:

```
swarm[i].error = Error(swarm[i].position);
swarm[i].intensity = 1 / (swarm[i].error + 1);
} // If firefly(i) < firefly(j)
} // j
} // i each firefly
...
```

Method Solve concludes with these statements:

```
...
Array.Sort(swarm); // low error to high
if (swarm[0].error < bestError)
{
    bestError = swarm[0].error;
    for (int k = 0; k < dim; ++k)
        bestPosition[k] = swarm[0].position[k];
}
++epoch;
} // While
return bestPosition;
} // Solve
```

After each pair of fireflies has been compared and less intense fireflies have moved toward more intense fireflies, the array of Firefly objects is sorted from low error to high error so that the best one is at swarm[0]. This object is checked to see if a new best solution has been found. Sorting the array of Firefly objects also has the important effect of changing their location within the array so that the objects are processed in a different order each time through the while loop.

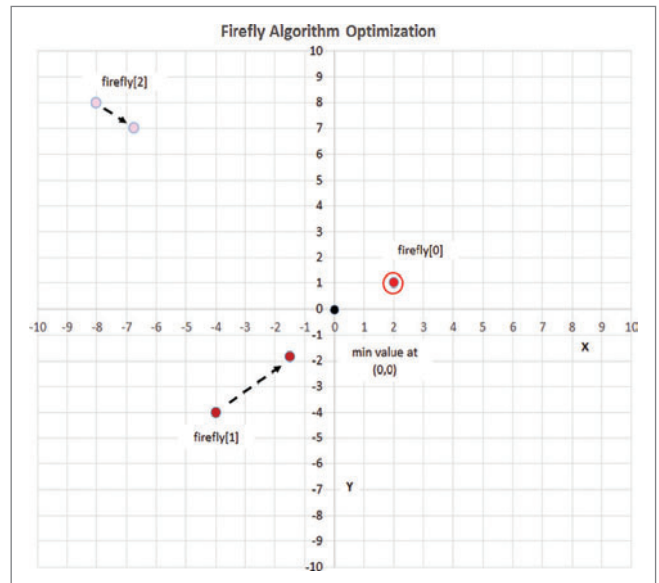


Figure 4 The Firefly Algorithm

The Helper Methods

Method Solve calls helper methods Distance and Error, which in turn calls helper method Michalewicz. Helper method Distance is defined as:

```
static double Distance(double[] posA, double[] posB)
{
    double ssd = 0.0; // sum squared differences
    for (int i = 0; i < posA.Length; ++i)
        ssd += (posA[i] - posB[i]) * (posA[i] - posB[i]);
    return Math.Sqrt(ssd);
}
```

Helper method Michalewicz is defined as:

```
static double Michalewicz(double[] xValues)
{
    double result = 0.0;
    for (int i = 0; i < xValues.Length; ++i) {
        double a = Math.Sin(xValues[i]);
        double b = Math.Sin(((i+1) * xValues[i] * xValues[i]) / Math.PI);
        double c = Math.Pow(b, 20);
        result += a * c;
    }
    return -1.0 * result;
}
```

If you refer to the math definition of the Michalewicz function at the bottom of **Figure 2**, you'll see that the function has an exponent of 2m. However, the value of m is usually set to 10, so in the code, a constant value of 20 is used. Helper method Error is defined as:

```
static double Error(double[] xValues)
{
    int dim = xValues.Length;
    double trueMin = 0.0;
    if (dim == 2)
        trueMin = -1.8013; // Approx.
    else if (dim == 5)
        trueMin = -4.687658; // Approx.
    double calculated = Michalewicz(xValues);
    return (trueMin - calculated) * (trueMin - calculated);
}
```

The error method just returns the product of the squared difference between the known minimum value of the Michalewicz function, and the calculated value. This dummy error function can be calculated very quickly, but in most machine learning scenarios, the error function can be very time consuming.

The Firefly Class

The Firefly class definition begins with:

```
public class Firefly : IComparable<Firefly>
{
    public double[] position;
    public double error;
    public double intensity;
    ...
}
```

The class inherits from the IComparable interface so that arrays and lists containing the object can be automatically sorted. The data fields are defined using public scope for simplicity. Because there's a one-to-one mapping between error and intensity, either of those two fields could be dropped. The class constructor is:

```
public Firefly(int dim)
{
    this.position = new double[dim];
    this.error = 0.0;
    this.intensity = 0.0;
}
```

There are many design alternatives you can consider. Here the constructor simply allocates space for the position array. The only other public method is CompareTo:

```
public int CompareTo(Firefly other)
{
    if (this.error < other.error) return -1;
    else if (this.error > other.error) return +1;
    else return 0;
}
// Class Firefly
```

The CompareTo method orders Firefly objects from low error to high. An equivalent alternative is to order from high intensity to low.

A Few Comments

The implementation of the firefly algorithm presented in this article is based on the seed 2009 paper. The original algorithm has spawned several variations. The research paper presents some data that suggests FA is superior to particle swarm optimization, at least on some dummy benchmark optimization problems. I'm somewhat skeptical. However, in my opinion, a scenario in which FA is very useful is when the objective function to be minimized has multiple solutions. Although it's not entirely obvious, as it turns

out, FA automatically self-organizes into sub-swarms that can find multiple solutions simultaneously. ■

DR. JAMES McCaffrey works for Microsoft Research in Redmond, Wash. He has worked on several Microsoft products including Internet Explorer and Bing. Dr. McCaffrey can be reached at jammc@microsoft.com.

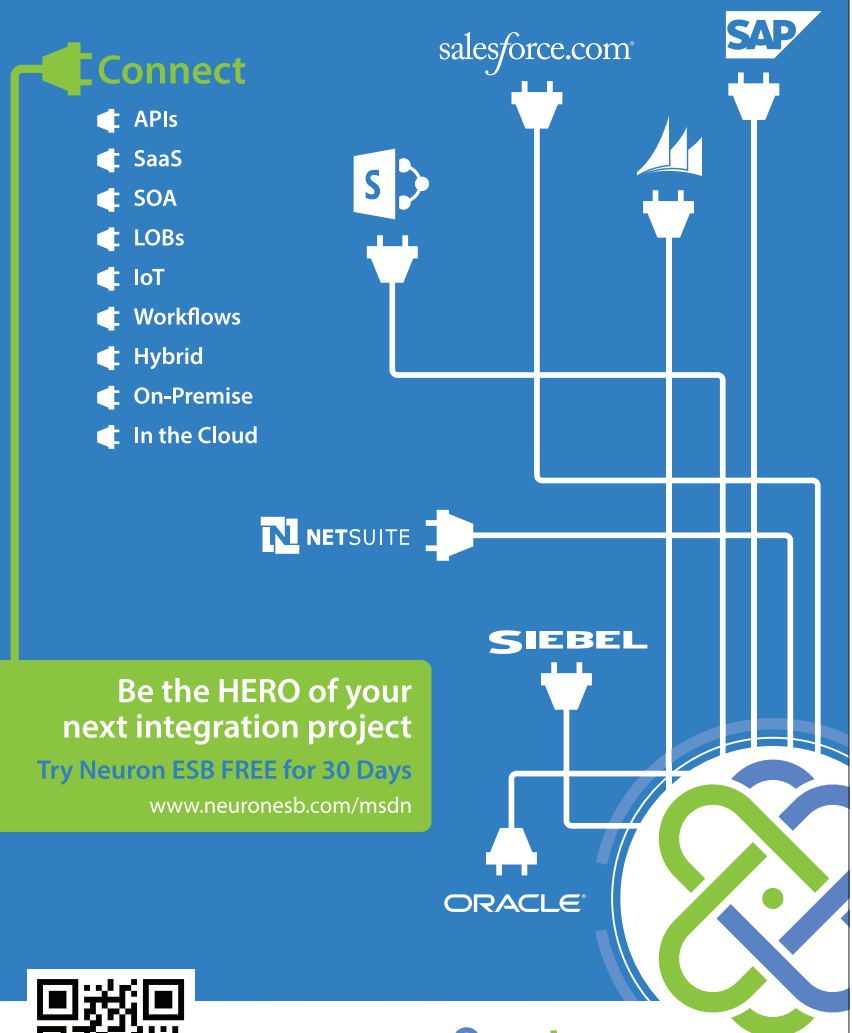
THANKS to the following Microsoft technical experts for reviewing this article: Todd Bello, Marciano Moreno Diaz Covarrubias and Alisson Sol

Figure 5 Firefly Algorithm

```
initialize n fireflies to random positions
loop maxEpochs times
    for i := 0 to n-1
        for j := 0 to n-1
            if intensity(i) < intensity(j)
                compute attractiveness
                move firefly(i) toward firefly(j)
                update firefly(i) intensity
            end for
        end for
    sort fireflies
end loop
return best position found
```

The Simple Solution to Your Complex Integration Challenges

Built by .NET Developers for .NET Developers



neuron  esb

Microsoft Partner
Gold Independent Software Vendor (ISV)



LIVE!
360

TECH EVENTS WITH PERSPECTIVE



2015 Orlando

ROYAL PACIFIC RESORT AT UNIVERSAL

November 16-20

LIVE360EVENTS.COM

The Ultimate Education Destination

Live! 360 is a unique conference where the IT and Developer community converge to test-drive leading edge technologies and fuel up on current ones. These five co-located events incorporate knowledge transfer and networking, along with finely tuned education and training, as you create your own custom conference, mixing and matching sessions and workshops to best suit your needs. All roads lead to Live! 360: **the ultimate education destination.**



EVENT PARTNERS



PLATINUM SPONSORS



GOLD SPONSOR



SUPPORTED BY



5
*Great
Conferences*
1
Great Price



Connect with Live! 360



twitter.com/live360
@live360



facebook.com
Search "Live 360"



linkedin.com
Join the "Live! 360" group!

REGISTER BY AUGUST
12 AND **SAVE \$500!**



Use promo code
L360JUN2

Scan the QR code to
register or for more
event details.



Take the Tour

Five events, 35+ tracks, and literally hundreds of sessions to choose from - mix and match sessions to create your own, custom event line-up - it's like no other conference available today.

Visual Studio **LIVE!**

CODE IN THE SUN and experience unbiased and practical development training on the Microsoft Platform.

SharePoint **LIVE!**

Chart the best **COLLABORATION COURSE** with SharePoint Live!, providing leading-edge SharePoint knowledge and training

SQL Server **LIVE!**

SQL Server Live! will leave you with the skills needed to **DRIVE DATA FORWARD**, whether you are a DBA, developer, IT Pro, or Analyst.

Modern Apps **LIVE!**

Presented in partnership with Magenit, this unique conference **LEADS THE WAY** to learning how to architect, design and build a complete Modern App.

TECHMENTOR

This is IT TRAINING WORTH TRAVELING **FOR**, focused entirely on making your datacenter more modern, more capable, and more manageable.



Natives, Immigrants and Symbionts

Marc Prensky coined the term “digital natives” in 2001 to describe people who grew from birth with digital technology in their lives. Those who were older when this technology appeared he called “digital immigrants.” My daughters, now 12 and 14, are digital natives. I am an immigrant, having gotten my first PC at age 26. No matter how hard I try, I will always speak geek with an accent.

Prensky was writing of the digital technology of that time, which meant a PC in the house, usually a desktop that you had to sit down within a particular place to use. That was half a human generation ago, an eternity in geek years. The whole tablet and mobile phone world hadn't happened yet. Now that it has, I see this native/immigrant divide cropping up again.

Today's young adults, digital natives by Prensky's original definition, love their smartphones. They keep them handy at all times, escape to them at even the slightest moment of boredom, worship them as the new fifth force of nature. (See my February 2012 column at msdn.microsoft.com/magazine/hh781031, especially the photo.)

As their digital immigrant parents did with PCs, today's mobile immigrants are layering this new technology onto their adult behavior patterns, formed years ago.

But native as they are to PCs, these post-millennials are immigrants to this constantly connected mobile world. As their digital immigrant parents did with PCs, today's mobile immigrants are layering this new technology onto their adult behavior patterns, formed years ago. They lack the native imprinting that will permeate the next generation, such as my 4-year-old niece, who could finger-swipe an iPad before she could walk. A child's impressionable brain, evolved to acquire language, cannot help but be sculpted by these new forces, into shapes never before seen.

The generation now being born will never experience the world without digital enhancement, not even for a minute. They'll wear digital baby monitor devices in their first cribs. Their slightest cry in the nursery will trigger an Amazon Echo to soothe them with a

melody. They'll play block-stacking games on their kiddie tablets instead of with actual wooden blocks.

As they become verbal, they'll start trying to control their world within its parameters: “OK Amazon, read me ‘Winnie-the-Pooh.’” Then they'll try to modify those parameters to show their power: “OK Amazon, read me ‘Winnie-the-Pooh,’ but this time make Piglet into bacon.”

They'll wear Google Glasses and Apple Watches and goodness knows what else to nursery school. I'm imagining kindergarten desks with charging ports. They will feel seriously impaired without their devices on, as you and I feel without our eyeglasses. Therefore, I hereby coin the name *digital symbionts* for this generation.

We will naturally need new digital assistants to raise these digital symbionts. Instead of Cortana and Siri, we'll have Mary Poppins or Supernanny—sort of a “Google is my co-parent” kind of thing. They'll be configurable for things like bedtimes, or the TV shows the kid is allowed to watch. Of course, the symbionts, with technical knowledge superior to their parents (some things never change), will adjust the settings to more permissive ones, and plead ignorance when they get caught.

I foresee huge dangers among the opportunities. We worry today about the National Security Agency (NSA) reading our e-mail; we'll start worrying about the NSA injecting subliminal conditioning messages (“Love your government!”) into the symbionts' subconscious minds through carrier signal modulation. Or a hacker could use information gleaned from social media: “Bobby, Fido misses you so much in doggie heaven. He'd like you to buy him a treat. Go into Daddy's wallet, get his Visa card and read me the number ...”

I can imagine parents buying apps that inculcate their chosen value systems into their digital symbiont children. The ultimate example is religion. The 2 year old calls out in the dark night: “Are you there, [insert deity of choice]?” And the Echo app replies, “Of course, Bobby, I'll always be with you.” After the hit I took when my girls caught me “helping” Santa Claus, I don't want to be around when the kid catches the parent feeding lines to the godhead.

Finally, consider this, and tremble: what will the digital symbionts' children be like? Perhaps *digital implantees*? And what about *their* children? ■

DAVID S. PLATT teaches programming .NET at Harvard University Extension School and at companies all over the world. He's the author of 11 programming books, including “Why Software Sucks” (Addison-Wesley Professional, 2006) and “Introducing Microsoft .NET” (Microsoft Press, 2002). Microsoft named him a Software Legend in 2002. He wonders whether he should tape down two of his daughter's fingers so she learns how to count in octal. You can contact him at rollthunder.com.

Of drivers who own family cars,

86%

would rather be *driving one of these.*



Move into the Fast Lane with MobileTogether®

In a little over two days, with one developer, we created a full-featured survey application and asked Android™, iOS®, and Windows® mobile users about their dream cars. 86% wanted to go faster.



ALTOVA®

MobileTogether®

Ditch the Minivan of Mobile Development

It's no longer practical to wait weeks or months for a cutting-edge mobile business solution. Your team needs to create, test, and deploy to all devices at the speed of business. With MobileTogether, we've removed the roadblocks.

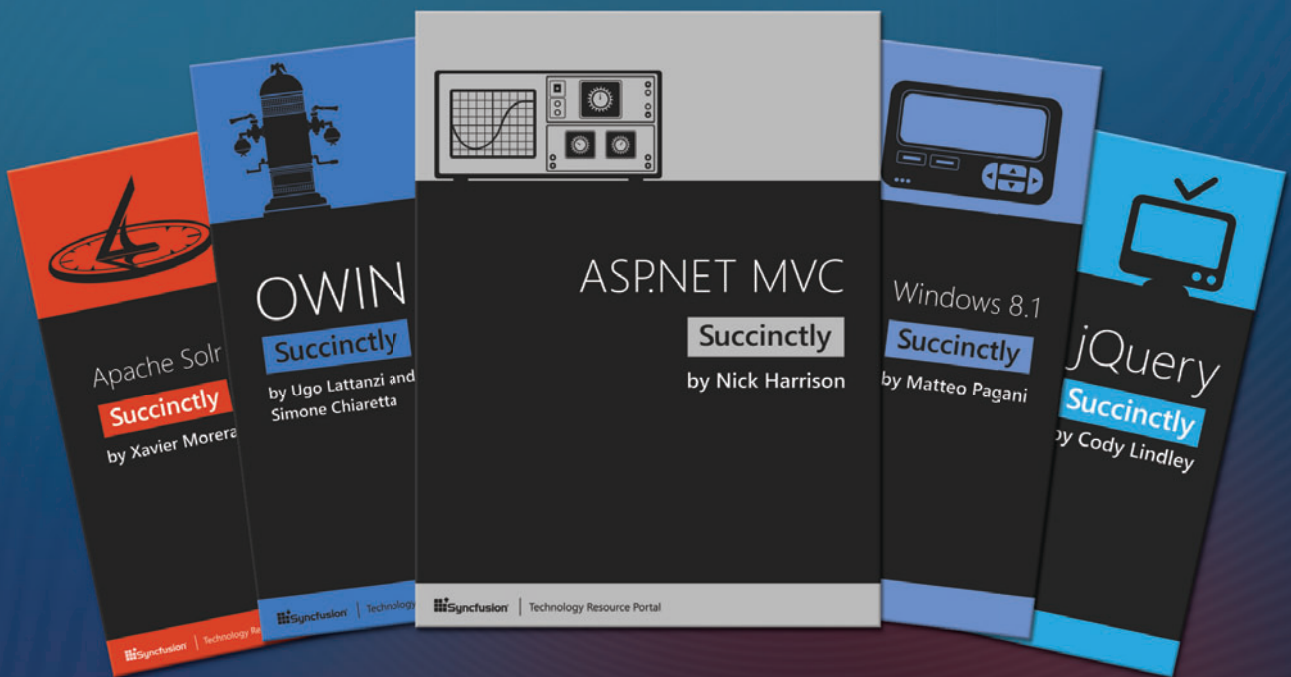
- Drag and drop UI design process
- Powerful visual & functional programming paradigm
- Native apps for all major mobile platforms
- Connectivity to SQL databases, XML, HTML, and more
- Deploy full-featured business app solutions in record time
- Pricing so affordable you might just have money left to start saving for that sports car



www.altova.com/MobileTogether

INTRODUCING
THE LATEST E-BOOK IN THE

SYNCFUSION SUCCINCTLY SERIES



60 titles and growing | Ad-free | 100 pages | PDF and Kindle formats

DOWNLOAD YOUR FREE COPY TODAY!

syncfusion.com/aspnetmvc



Powerful File APIs that are easy and intuitive to use

Native APIs for .NET, Java & Cloud

Adding File Conversion and Manipulation
to Business Systems

DOC, XLS, JPG, PNG, PDF,
BMP, MSG, PPT, VSD, XPS &
many other formats.

 www.aspose.com

Working with Files?



- ✓ CONVERT
- ✓ PRINT
- ✓ CREATE
- ✓ COMBINE
- ✓ MODIFY

100% Standalone - No Office Automation

Scan for
20% Savings!



ASPOSE

Your File Format APIs

ASPOSE.TOTAL

Every Aspose component combined in
ONE powerful suite!



Powerful

File Format APIs

- ▶ **Aspose.Words**
DOC, DOCX, RTF, HTML, PDF, XPS & other document formats.
 - ▶ **Aspose.Cells**
XLS, XLSX, XLSM, XLTX, CSV, SpreadsheetML & image formats.
 - ▶ **Aspose.BarCode**
JPG, PNG, BMP, GIF, TIF, WMF, ICON & other image formats.
 - ▶ **Aspose.Pdf**
PDF, XML, XLS-FO, HTML, BMP, JPG, PNG & other image formats.
 - ▶ **Aspose.Email**
MSG, EML, PST, EMLX & other formats.
 - ▶ **Aspose.Slides**
PPT, PPTX, POT, POTX, XPS, HTML, PNG, PDF & other formats.
 - ▶ **Aspose.Diagram**
VSD, VSDX, VSS, VST, VSX & other formats.
- ... and many others!*

Aspose.Total for .NET

Aspose.Total for Java

Aspose.Total for Cloud

Get your FREE evaluation copy at www.aspose.com

.NET

Java

Cloud

Aspose.Cells

Work with spreadsheets and data without depending on Microsoft Excel

- Solution for spreadsheet creation, manipulation and conversion.
- Import and export data.

ASPOSE.CELLS IS A

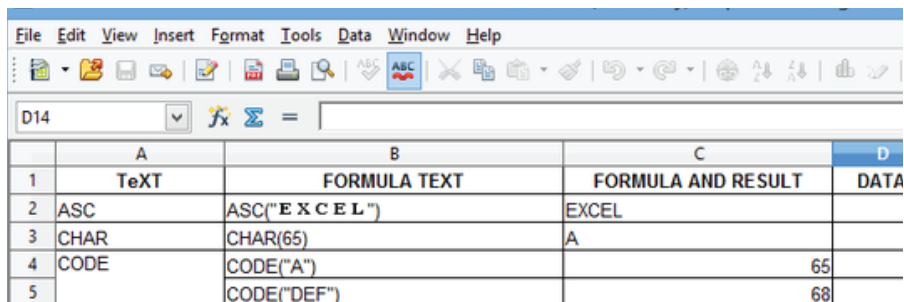
PROGRAMMING API that allows developers to create, manipulate and convert Microsoft Excel spreadsheet files from within their own applications. Its powerful features make it easy to convert worksheets and charts to graphics or save reports to PDF.

Aspose.Cells speeds up working with Microsoft Excel files. The

API is a flexible tool for simple tasks such as file conversion, as well as complex tasks like building models. Developers control page layout, formatting, charts and formulas. They can read and write spreadsheet files and save out to a wide variety of image and text file formats.

Fast and reliable, Aspose.Cells saves time and effort compared to using Microsoft Office Automation.

A flexible API for simple and complex spreadsheet programming.



	A	B	C	D
1	TeXT	FORMULA TEXT	FORMULA AND RESULT	DATA
2	ASC	ASC("EXCEL")	EXCEL	
3	CHAR	CHAR(65)	A	
4	CODE	CODE("A")		65
5		CODE("DEF")		68

Aspose.Cells lets developers work with data sources, formatting, even formulas.

Common Uses

- Building dynamic reports on the fly.
- Creating Excel dashboards with charts and pivot tables.
- Rendering and printing spreadsheets and graphics with high fidelity.
- Exporting data to, or importing from, Excel spreadsheets.
- Generating, manipulating and editing spreadsheets.
- Converting spreadsheets to images or other file formats.

Key Features

- A complete spreadsheet manipulation solution.
- Flexible data visualization and reporting.
- Powerful formula engine.
- Complete formatting control.

Supported File Formats

XLS, XLSX, XLSM, XMPS, XLTX, XLTM, ODS, SpreadsheetML, tab delim., CSV, TXT, PDF, HTML, and many image formats including TIFF, JPEG, PNG and GIF.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$999	\$1498	Site Small Business	\$4995	\$7490
Developer OEM	\$2997	\$4494	Site OEM	\$13986	\$20972

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926

Aspose.Cells for

.NET, Java, Cloud & more

File Formats

XLS XLSX TXT PDF HTML CSV TIFF PNG JPG BMP SpreadsheetML and many others.

Spreadsheet Manipulation

Aspose.Cells lets you create, import, and export spreadsheets and also allows you to manipulate contents, cell formatting, and file protection.

Creating Charts

Aspose.Cells comes with complete support for charting and supports all standard chart types. Also, you can convert charts to images.

Graphics Capabilities

Easily convert worksheets to images as well as adding images to worksheets at runtime.

Get your FREE Trial at
<http://www.aspose.com>

No Office Automation

Aspose.Cells does not require Microsoft Office to be installed on the machine in order to work.

Aspose.Words

Program with word processing documents independently of Microsoft Word

- Solution for document creation, manipulation and conversion.
- Advanced mail merge functionality.

ASPOSE.WORDS IS AN ADVANCED PROGRAMMING

API that lets developers perform a wide range of document processing tasks with their own applications. Aspose.Words makes it possible to generate, modify, convert, render and print documents without Microsoft Office Automation. It provides sophisticated and flexible access to, and control over,

Microsoft Word files.

Aspose.Words is powerful, user-friendly and

feature rich. It saves developers time and effort compared to using Microsoft Office Automation and makes gives them powerful document management tools.

Aspose.Words makes creating, changing and converting DOC and other word processing file formats fast and easy.

Generate, modify, convert, render and print documents without Microsoft Office Automation.

	Table			
	Column 1	Column 2	Column 3	Column 4
Row 1	Cell 1	Cell 2	Cell 3	Cell 4
Row 2	Cell 1	Cell 2	Cell 3	
Row 3	Cell 1	Cell 2		

Aspose.Words has sophisticated controls for formatting and managing tables and other content.

Common Uses

- Generating reports with complex mail merging; mail merging images.
- Populating tables and documents with data from a database.
- Inserting formatted text, paragraphs, tables and images into Microsoft Word documents.
- Adding barcodes to documents.
- Inserting diagrams and watermarks into Word documents.
- Formatting date and numeric fields.

Key Features

- A complete Microsoft Word document manipulation solution.
- Extensive mail merge features.
- Complete formatting control.
- High-fidelity conversion, rendering and printing.

Supported File Formats

DOC, DOCX, ODT, OOXML, XML, HTML, XHTML, MHTML, EPUB, PDF, XPS, RTF, and a number of image formats, including TIFF, JPEG, PNG and GIF.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$999	\$1498	Site Small Business	\$4995	\$7490
Developer OEM	\$2997	\$4494	Site OEM	\$13986	\$20972

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926

Case Study: Aspose.Words for .NET

ModulAcht e.K. - using Aspose.Words for .NET to convert from DOCX to PDF.

MODULACHT IS A SOFTWARE DEVELOPMENT TEAM

WHICH CREATES INDIVIDUAL SOFTWARE

for small businesses. Mostly we develop web applications including web UI and web-services, but we are also familiar with Windows Forms and Windows Services applications based on .NET.

Problem

For our main customer, we are developing the operating system they will use to administer the buying and selling of cars. With a need to generate documents easily, one of the main requirements was to have an easy-to-use template system.

"The really quick and competent support of Aspose helped us to solve some initial problems."

Looking for a Solution

We searched on the internet for DOCX to PDF converters, which is not as easy as it sounds. After filtering all the Interop wrappers only a handful of components remained to be tested. At the end only Aspose.Words for .NET created a result which really looks like the input DOCX. The really quick and competent support of Aspose helped us to solve some initial problems.

Implementation

Aspose.Words for .NET was the 4th component we tested. On our development machine, everything worked great, but after moving the code on to our test-server-machine, the resulting PDF did not look like the original DOCX file. Adjusting the settings didn't help so we decided to give the support team of Aspose a try.

After a short discussion in the live chat we started a new thread including a description, the input and the output file, in the Aspose.Words forum. Within less than 24 hours one of the support-team members told us that we would

have to check whether the font we used in the DOCX file was available on the server machine, which it was not. After changing the font, the

whole PDF looks exactly the same as the DOCX file.

Outcome

Choosing Aspose.Words for .NET meant an intuitive and easy to use software component and also getting a really friendly and straightforward software partner which is ready to help if you need help.

Next Steps

After getting our Test-Driver ready we will implement the template engine in our customer's software. Aspose.Words for .NET functionality will be used on many different places in this software to convert files into the PDF format.

This is an extract from a case study on our website. For the full version, go to: www.aspose.com/corporate/customers/case-studies.aspx



After converting, our PDF looks exactly the same as the DOCX file.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926



Open, Create, Convert, Print & Save Files

from within your *own* applications.

> ASPOSE.TOTAL

allows you to process these file formats:

- Word documents
- Excel spreadsheets
- PowerPoint presentations
- PDF documents
- Project documents
- Visio documents
- Outlook emails
- OneNote documents



**DOC XLS PPT PDF EML
PNG XML RTF HTML VSD
BMP & barcode images.**



ASPOSE

Your File Format APIs

Contact Us:

US: +1 888 277 6734

EU: +44 141 416 1112

AU: +61 2 8003 5926

sales@aspose.com

Helped over 11,000 companies and over 250,000 users work with documents in their applications.



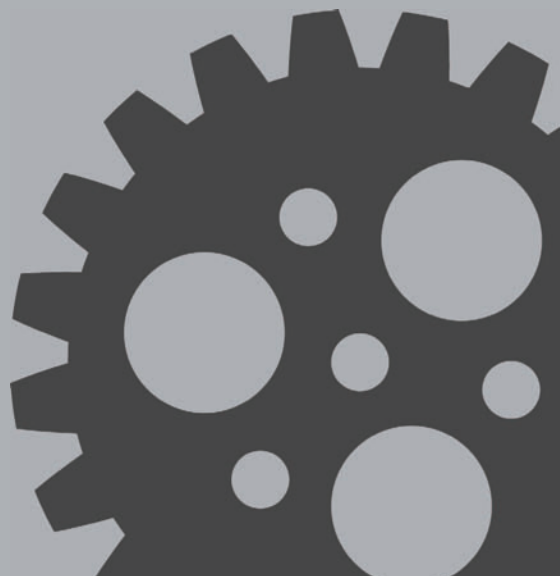
.NET, Java, and Cloud

Your File Format APIs



GET STARTED NOW

- Free Trial
- 30 Day Temp License
- Free Support
- Community Forums
- Live Chat
- Blogs
- Examples
- Video Demos



Adding File Conversion and Manipulation to Business Systems

How often do people in your organization complain that they can't get information in the file format and layout they want? Converting documents from one format to another without losing layout and formatting should be simple, but it can be frustrating for both users and developers.

EXTRACTING DATA FROM A DATABASE AND DELIVERING IT TO THE SALES TEAM AS A REPORT, complete with charts and corporate branding, is fine. Until the sales team says that they want it as a Microsoft Excel file, and could you add a dashboard?

Using information from online forms in letters that can be printed and posted is easy. But what if you also want to add tracking barcodes and archive a digital copy as a PDF?

Ensuring that your business system supports all the different Microsoft Office file formats your users want can be difficult. Sometimes the native file format support of your system lets you down. When that is the case, use tools that extend that capability. A good tool can save you time and effort.

Document Conversion Options

Building your own solution: Time-consuming and costly, this option is only sensible if the solution you develop is central to your business.

Using Microsoft Office

Automation: Microsoft Office

Automation lets you use Microsoft Office programs server-side. It is not how the Office products were designed to be used. It can work well but you might notice issues with the stability, security and speed of the system, as well as cost.

Aspose creates APIs that work independently of Microsoft Office Automation.

Using an API: The API market has lots of free and commercial solutions, some very focused, some feature-rich. An API integrates with your code and gives you access to a range of new features.

Look to Aspose

Aspose are API experts. We create APIs, components and extensions that work independently of Microsoft Automation to extend a platform's native file format manipulation capabilities.

Aspose have developed APIs for .NET, Java, Cloud and Android that lets developers convert, create and manipulate Microsoft Office files – Microsoft Word, Excel, PowerPoint, Visio and Project – and other popular business formats, from PDFs and images to emails. We also have APIs for working with images,

barcodes and OCR. The APIs are optimised for stability, speed and ease of use. Our APIs save users weeks, sometimes months, of effort.



Finding the Right Tool

To find the product that's right for you, take a systematic approach:

- List must-have and nice-to-have features.
- Research the market.
- Ask for recommendations.
- Select a few candidates .
- Run trials.
- Evaluate
 - ease of use,
 - support and documentation,
 - performance, and
 - current and future needs.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926

Aspose.BarCode

A complete toolkit for barcode generation and recognition

- Generate barcodes with customer defined size and color.
- Recognize a large number of barcode types from images.

ASPOSE.BARCODE IS A ROBUST AND RELIABLE BARCODE GENERATION AND RECOGNITION API that allows developers to add barcode generation and recognition functionality to their applications quickly and easily.

Aspose.BarCode supports most established barcode specifications. It can export generated barcodes to multiple image formats, including BMP, GIF, JPED, PNG and TIFF.

Aspose.BarCode gives you full control over every aspect of the barcode

Robust and reliable barcode generation and recognition.

image, from background and bar color, through image quality, rotation angle, X-dimension, captions, and resolution.

Aspose.BarCode can read and recognize most common 1D and 2D barcodes from any image and at any angle. Filters help developers



Aspose.BarCode offers a large number of symbologies and formatting options.

clean up difficult to read images to improve recognition.

Common Uses

- Generating and recognizing barcode images.
- Printing barcode labels.
- Enhancing workflow by adding barcode functionality.
- Using recognition functions to drive real-life work processes.

Key Features

- Barcode generation and recognition.
- Comprehensive support for 1D and 2D symbologies.
- Image processing for improved recognition.

Supported File Formats

JPG, TIFF, PNG, BMP, GIF, EMF, WMF,

EXIP and ICON.

Format support varies across platforms.

Supported Barcodes

Linear: EAN13, EAN8, UPCA, UPCE, Interleaved2of5, Standard2of5, MSI, Code11, Codabar, EAN14(SCC14), SSCC18, ITF14, Matrix 2 of 5, PZN, Code128, Code39 Extended, Code39 Standard, OPC, Code93 Extended, Code93 Standard, IATA 2 of 5, GS1Code128, ISBN, ISMN, ISSN, ITF6, Pharmacode, DatabarOmniDirectional, VIN, DatabarTruncated, DatabarLimited, DatabarExpanded, PatchCode, Supplement **2D:** PDF417, MacroPDF417, DataMatrix, Aztec, QR, Italian Post 25, Code16K, GS1DataMatrix **Postal:** Postnet, Planet, USPS OneCode, Australia Post, Deutsche Post Identcode, AustralianPosteParcel, Deutsche Post Leticode, RM4SCC, SingaporePost, SwissPostParcel

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$599	\$1098	Site Small Business	\$2995	\$5490
Developer OEM	\$1797	\$3294	Site OEM	\$8386	\$15372

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926









Aspose *for* Cloud

The easiest API to Create, Convert & Automate Documents in the cloud.



**Convert
Create
Render
Combine
Modify**

without installing anything!

Aspose.Words for Cloud  Create and convert docs Manipulate text Render documents Annotate	Aspose.Cells for Cloud  Create spreadsheets Convert spreadsheets Manipulate cells and formulas Render spreadsheets
Aspose.Slides for Cloud  Create presentations Manage slides Edit text and images Read and convert	Aspose.Pdf for Cloud  Create and convert PDFs Manipulate text, images Add pages, split, encrypt Manage stamps
Aspose.Email for Cloud  Create, update, and convert messages Extract attachments Use with any language	Aspose.BarCode for Cloud  Generate barcodes Read barcodes Set attributes Multiple image formats

Free Evaluation at www.aspose.com

Aspose.Email

Work with emails and calendars without Microsoft Outlook

- Complete email processing solution.
- Message file format support.

ASPOSE.EMAIL IS AN EMAIL PROGRAMMING API that allows developers to access and work with PST, EML, MSG and MHT files. It also offers an advanced API for interacting with enterprise mail systems like Exchange and Gmail.

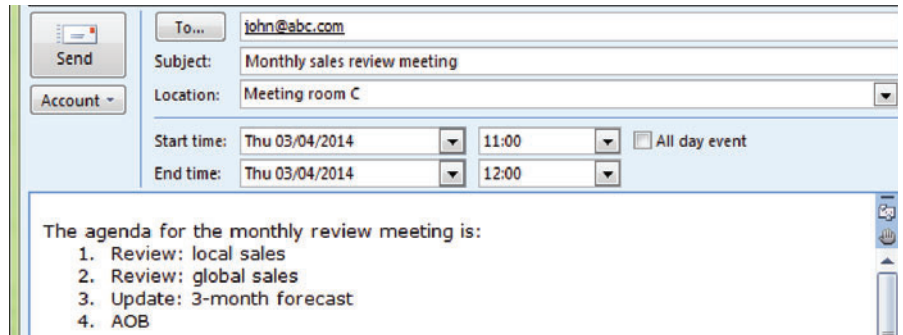
Aspose.Email can work with HTML and plain text emails, attachments and embedded OLE objects. It

allows developers to work against SMTP, POP, FTP and Microsoft Exchange servers. It supports mail

merge and iCalendar features, customized header and body, searching archives and has many other useful features.

Aspose.Email allows developers to focus on managing email without getting into the core of email and network programming. It gives you the controls you need.

Aspose.
Email works
with HTML
and plain
text emails,
attachments
and embedded
OLE objects.



Aspose.Email lets your applications work with emails, attachments, notes and calendars.

Common Uses

- Sending email with HTML formatting and attachments.
- Mail merging and sending mass mail.
- Connecting to POP3 and IMAP mail servers to list and download messages.
- Connecting to Microsoft Exchange Servers to list, download and send messages.
- Create and update tasks using iCalendar.
- Load from and save messages to file or stream (EML, MSG or MHT formats).

Key Features

- A complete email processing solution.
- Support for MSG and PST formats.
- Microsoft Exchange Server support.
- Complete recurrence pattern solution.

Supported File Formats

MSG, MHT, OST, PST, EMLX, TNEF, and EML.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$599	\$1059	Site Small Business	\$2995	\$5490
Developer OEM	\$1797	\$3294	Site OEM	\$8386	\$15372

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926

Aspose.Pdf

Create PDF documents without using Adobe Acrobat

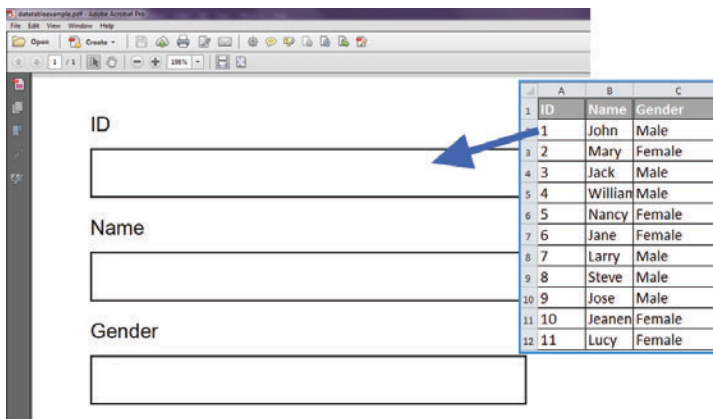
- A complete solution for programming with PDF files.
- Work with PDF forms and form fields.

ASPOSE.PDF IS A PDF DOCUMENT CREATION AND MANIPULATION API that developers use to read, write and manipulate PDF documents without using Adobe Acrobat. Aspose.Pdf is a sophisticated product that integrates with your application to add PDF capabilities.

Aspose.Pdf offers a wealth of features that lets developers compress files, create tables, work with links, add and remove security, handle custom fonts, integrate with external data sources, manage bookmarks, create table of contents, create forms and manage form fields.

Read, write and manipulate PDF documents independently of Adobe Acrobat.

It helps developers add, work with attachments, annotations and PDF form data, add, replace or remove text and images, split, concatenate,



Aspose.Pdf can be used to automatically complete PDF forms with external data.

extract or inset pages, and print PDF documents.

Common Uses

- Creating and editing PDF files.
- Inserting, extracting, appending, concatenating and splitting PDFs.
- Working with text, images, tables, images, headers, and footers.
- Applying security, passwords and signatures.
- Working with forms and form fields.

Key Features

- PDF creation from XML or XLS-FO documents.
- PDF form and field support.
- Advanced security and encryption.
- High-fidelity printing and conversion.
- Supported File Formats
- PDF, PDF/A, PDF/A_1b, PCL, XLS-FO, LaTeX, HTML, XPS, TXT and a range of image formats.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$799	\$1298	Site Small Business	\$3995	\$6490
Developer OEM	\$2397	\$3894	Site OEM	\$11186	\$18172

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926

Aspose.Pdf

.Net, Java & Cloud

File Formats

PDF XPS ePUB HTML XML XLS TXT DOC XSL-FO & other image file formats.

Create and Manipulate PDFs

Create new or edit/manipualte existing PDFs.

Form Field Features

Add form fields to your PDFs. Import and export form fields data from select file formats.

Table Features

Add tables to your PDFs with formatting such as table border style, margin and padding info, column width and spanning options, and more.

Get started today at www.aspose.com



Conversion is Fast And High-Fidelity



Aspose.Note for .NET

Aspose.Note for .NET is an API that lets developers convert Microsoft OneNote pages to a variety of file formats, and extract the text and document information.

Conversion is fast and high-fidelity. The output looks like the OneNote page, no matter how complex the formatting or layout.

Aspose.Note works independently of Office Automation and does not require Microsoft Office or OneNote to be installed.

Product	Benefit	Supported Platforms
Aspose.Note for .NET	Modify, convert, render and extract text and images from Microsoft OneNote files without relying on OneNote or other libraries.	.NET Framework 2.0, 3.0, 3.5, 4.0, 4.0 CP

Features

File Formats and Conversion		Rendering and Printing	Document Management
Microsoft OneNote 2010, 2010 SP1, 2013	Load, Save	Save as Image (BMP, GIF, JPG, PNG)	<ul style="list-style-type: none">Extract textGet the number of pages in a document.Get page information.Extract images.Get image information from a document.Replace text in document.
PDF	Save	Save as PDF	
Images (BMP, GIF, JPG, PNG)	Save		

Aspose.Imaging

Create Images from scratch.

- Load existing images for editing purposes.
- Render to multiple file formats.

ASPOSE.IMAGING IS A CLASS

LIBRARY that facilitates the developer to create Image files from scratch or load existing ones for editing purpose. Also, Aspose.Imaging provides the means to save the created or edited Image to a variety of formats. All of the above mentioned can be achieved without the need of an Image Editor. It works independent of other applications and although Aspose.Imaging allows you to save to Adobe PhotoShop® format (PSD), you do not need PhotoShop installed on the machine.

Aspose.Imaging is flexible, stable and powerful. It's many features and image processing routines should meet most imaging requirements. Like all Aspose file format components, Aspose.

Imaging introduces support for an advanced set of drawing features along with the core functionality. Developers can

Create images from scratch. or load existing ones...



Aspose.Imaging allows creation and manipulation of images.

draw on Image surface either by manipulating the bitmap information or by using the advanced functionality like Graphics and Paths.

Common Uses

- Create images from scratch.
- Load and Edit existing images.
- Export images to a variety of formats.
- Adding watermark to images.
- Export CAD drawings to PDF & raster image formats.
- Crop, resize & RotateFlip images.
- Extract frames from multipage TIFF image.

Key Features

- Create, edit, and save images
- Multiple file formats
- Drawing features
- Export images

Supported File Formats

BMP, JPG, TIFF, GIF, PNG, PSD, DXF, DWG, and PDF.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$399	\$898	Site Small Business	\$1995	\$4490
Developer OEM	\$1197	\$2694	Site OEM	\$5586	\$12572

The pricing info above is for .NET.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926

Aspose.Slides

Work with presentations without using Microsoft PowerPoint

- Complete solution for working with presentation files.
- Export presentations and slides to portable or image formats.

ASPOSE.SLIDES IS A FLEXIBLE PRESENTATION MANAGEMENT API that helps developers read, write and manipulate Microsoft PowerPoint documents. Slides and presentations can be saved to PDF, HTML and image file formats without Microsoft Office Automation.

Aspose.Slides offers a number of advanced features that make it easy to perform tasks such as rendering slides, exporting presentations, exporting slides to SVG and printing. Developers use Aspose.Slides to build customizable slide decks, add or remove standard graphics and automatically publish presentations to other formats. Aspose.Slides gives developers the tools they need to work with presentation files. It integrates quickly and saves time and money.

Aspose.Slides gives you the tools you need to work with presentation files.



Aspose.Slides has advanced features for working with every aspect of a presentation.

Common Uses

- Creating new slides and cloning existing slides from templates.
- Handling text and shape formatting.
- Applying and removing protection.
- Exporting presentations to images and PDF.
- Embedding Excel charts as OLE objects.
- Generate presentations from database.

Key Features

- A complete presentation development solution.
- Control over text, formatting and slide elements.
- OLE integration for embedding

external content.

- Wide support for input and output file formats.

Supported File Formats

PPT, POT, PPS, PPTX, POTX, PPSX, ODP, PresentationML, XPS, PDF and image formats including TIFF and JPG.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$799	\$1298	Site Small Business	\$3995	\$6490
Developer OEM	\$2397	\$3894	Site OEM	\$11186	\$18172

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926

Support Services

Get the assistance you need, when you need it, from the people who know our products best.

- Use experienced Aspose developers for your projects
- Get the level of support that suits you and your team

NO ONE KNOWS OUR PRODUCTS AS WELL AS WE DO.

We develop them, support them and use them. Our experience is available to you, whether you want us to develop a solution for you, or you just need a little help to solve a particular problem.

Consulting

Aspose's developers are expert users of Aspose APIs. They understand how to use our products and have hands-on experience of using them for software development. Aspose's developers are skilled not just with Aspose tools but in a wide range of programming languages, tools and techniques.

When you need help to get a project off the ground, Aspose's developers can help.

Aspose's file format experts are here to help you with a project or your support questions



Work with the most experienced Aspose developers in the world.

Consulting Benefits

- Use Aspose engineers to work on your products
- Get peace of mind from a fully managed development process
- Get a custom-built solution that meets your exact needs

Support Options

Free

Everyone who uses Aspose products have access to our free support. Our software developers are on stand-by to help you succeed with your project, from the evaluation to roll-out of your solution.

Priority

If you want to know when you'll hear back from us on an issue, and know that your issue is prioritized, Priority Support is for you. It provides a more formal support structure and has its own forum that is monitored by our software engineers.

Enterprise

Enterprise customers often have very specific needs. Our Enterprise Support option gives them access to the product development team and influence over the roadmap. Enterprise Support customers have their own, dedicated issue tracking system.



Pricing Info

Each consulting project is evaluated individually; no two projects have exactly the same requirements.

To see the Priority and Enterprise support rates, refer to the product price list, or contact our sales team.

www.aspose.com

EU: +44 141 416 1112

US: +1 888 277 6734
sales@aspose.com

Oceania: +61 2 8003 5926

We're Here to Help You

Aspose has 4 Support Services to best suit your needs

Free Support

Support Forums with no Charge

Priority Support

24 hour response time in the week,
issue escalation, dedicated forum

Enterprise Support

Communicate with product
managers, influence the roadmap

Sponsored Support

Get the feature you need built now

Technical Support is an issue that Aspose takes very seriously. Software must work quickly and dependably. When problems arise, developers need answers in a hurry. We ensure that our clients receive useful answers and solutions quickly.

Email • Live Chat • Forums

Contact Us

US Sales: +1 888 277 6734
sales@aspose.com

EU Sales: +44 141 416 1112

AU Sales: +61 2 8003 5926

