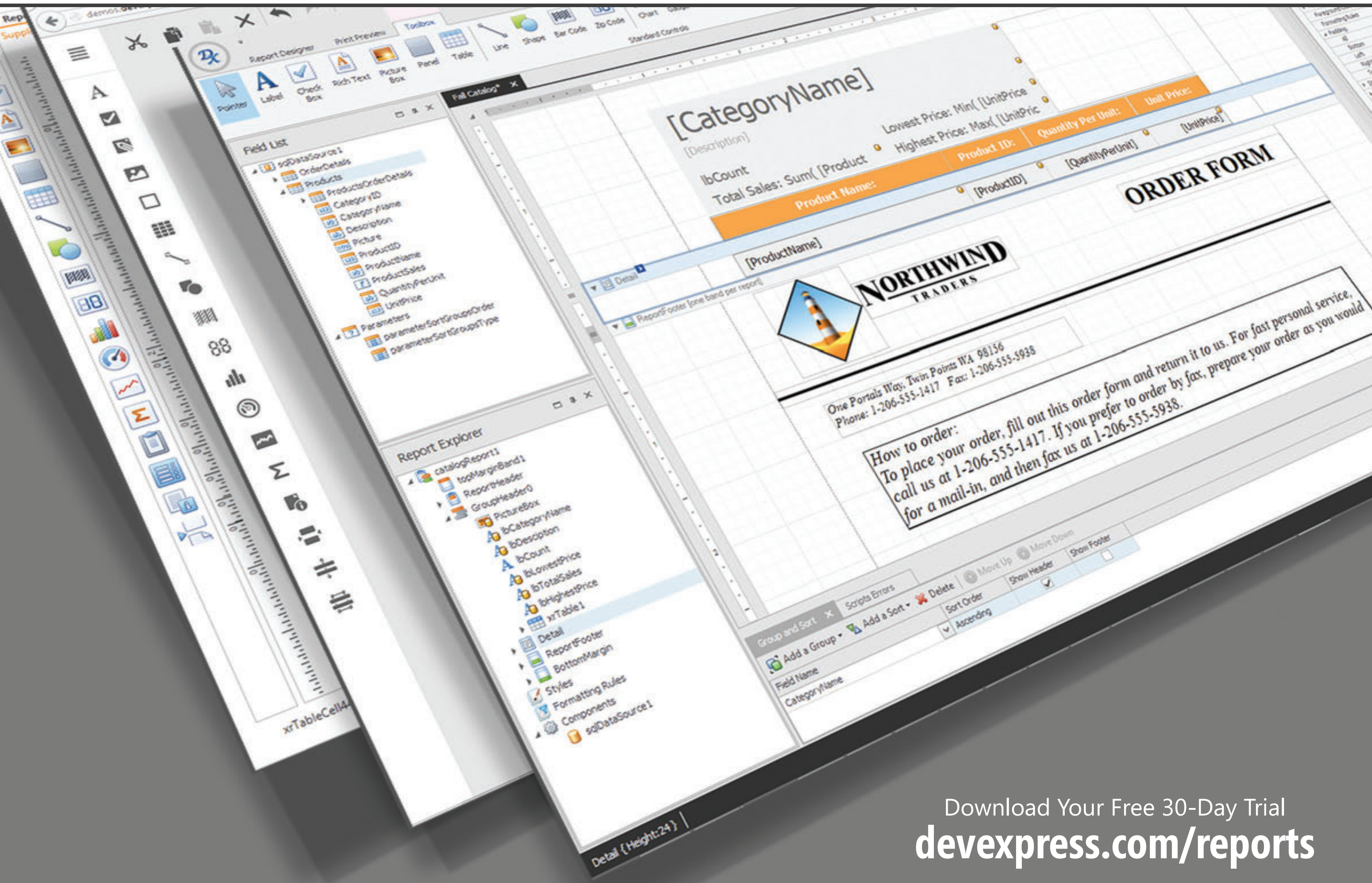
THE MICROSOFT JOURNAL FOR DEVELOPERS

AUGUST 2015 VOL 30 NO 8

msdn magazine



One Portals Way, Twin Points WA 98156
Phone: 1-206-555-1417 Fax: 1-206-555-5938

How to order:
To place your order, fill out this order form and return it to us. For fast personal service, call us at 1-206-555-1417. If you prefer to order by fax, prepare your order as you would for a mail-in, and then fax us at 1-206-555-5938.

THE MICROSOFT JOURNAL FOR DEVELOPERS

AUGUST 2015 VOL 30 NO 8

msdn magazine

Drag and Drop for Windows Universal Apps...20

COLUMNS

Microsoft

XLS

msdn magazine

AUGUST 2015 VOLUME 30 NUMBER 8

General Manager Jeff Sandquist
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

ENTERPRISE COMPUTINGGROUP

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**

1105 MEDIA INC

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.

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: jlong@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: FirstinitialLastname@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 contentsherein do not necessarily express those of the publisher.

Microsoft

LEADTOOLS®
THE WORLD LEADER IN IMAGING SDKs
PDF
MERGE
SPLIT
INSERT
TIFF
150 + FORMATS
FORMS
3D
DICOM
OCR
DWG
SVG
PACS
DOC/DOCX

Editor's Note

MICHAEL DESMOND

The Importance of Play

Back in March, Michael Oneppo kicked off a series of articles focused on game development. Since that first installment ("A Web Game in an Hour" [msdn.microsoft.com/magazine/dn913185]), he's published two additional features: "2D Drawing Techniques and Libraries for Web Games" (msdn.microsoft.com/magazine/dn948109) in the April issue and "2D Game Engines for the Web" (msdn.microsoft.com/magazine/dn973016) in May. This month, he explores 3D game development with his feature, "Introduction to 3D for Web Games."

Oneppo's series is hardly the first time we've covered game development in *MSDN Magazine*. Adam Tuliper last year published an outstanding series on cross-platform game development on the Unity framework, and our Azure Insider authors Bruno Terkaly and Ricardo Villalobos the year before showed how Microsoft Azure can support back-end development for gaming. (msdn.microsoft.com/magazine/dn532200).

Heck, Oneppo's article isn't even the first coverage of game development in this issue. That honor belongs to this month's Upstart column author Michael Thompson, a Microsoft content developer on the Visual Studio team who focuses on C++, graphics and gaming. He provides an insightful look at popular game dev frameworks—including Unity, Unreal Engine 4 and the open source Cocos2D—and reveals how they're enabling a new generation of game developers.

The common thread in all this coverage is the way game development intersects mainstream programming concerns and disciplines. From back-end development in Azure to cross-platform development in Unity and Cordova, game developers are facing—and solving—many of the same challenges that vex their business-minded counterparts.

Oneppo is a six-year veteran in the DirectX team at Microsoft. He's also enrolled as a graduate student in the Interactive Telecommunications Program at NYU, where he says the experience of working alongside artists and creative technologists on game projects has informed his view of software development. He describes game development as an on-ramp to advanced concepts that are "needed in everyday software engineering."

"For example, the art of making a game fun is deeply rooted in psychology and provides the foundation for user experience design .... Another example is physics and simulation. If you want to make a projectile fly across the screen in a realistic way, that's basic calculus in a much easier to swallow package," Oneppo says. "The list goes on and on: Role-playing games are all about statistics, computer graphics is multi-variable calculus, enemy AI is machine learning, etc."

> The common thread in all this coverage is the way game development intersects mainstream programming concerns and disciplines.

Maturing platforms like Unity and Unreal Engine 4 abstract many of the thorniest challenges of game programming, and make the discipline accessible to coders who might otherwise never get started in the field. Of course, plenty of opportunities exist for what Thompson calls "ninja-level game developers," and there are ample lessons to be learned there, as well, as Oneppo notes.

"It's also worth mentioning that advanced game development requires a deep understanding of how the machine is natively executing your code, to get the best possible performance out of the system," Oneppo says. "So game development is a great place to learn C++, OS architecture, low-level networking and optimization."

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

/n software

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 ...

Microsoft .NET | Microsoft BizTalk Server | Microsoft SQL Server | Windows Mobile | activex | Microsoft SharePoint 2010 | CODE GEAR DELPHI | CODE GEAR C++BUILDER | Java | php

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

Microsoft .NET, | Java | ActiveX | C++ | Delphi | C++ Builder | PHP | Objective-C | Windows | Windows Mobile | Mac OS X | Linux/Unix | iPhone

Committing to Cross Platform: The ABCs of iOS

A recurring theme at conferences like Build this year has been that Microsoft is serious about cross-platform development. To that end, Visual Studio 2015, Visual Studio Online (VSO) and Visual Studio Code now allow developers to access a powerful, first-class experience for creating great iOS apps.

This is big news. In addition to Visual Studio 2015 supporting cross-platform development in C#/Xamarin and C++, developers can now use Apache Cordova to develop, build, debug and deploy iOS apps using pure, standards-compliant JavaScript. Better yet, the Cordova project structure in Visual Studio 2015 mirrors the file system, which means that any Cordova project you create can be edited and maintained using both Visual Studio on Windows or Visual Studio Code on a Mac. Same project, same code. You can even mix in popular command-line utilities like those for the powerful Ionic or Monaca JavaScript frameworks.

Three Paths to iOS App Development

Visual Studio 2015 Tools for Apache Cordova (aka.ms/cordova) enables you to build, debug, run and simulate an iOS version of your app through an integrated remote agent running on OS X. You can install this agent on your own machine, use a "pay-as-you-go" or managed plan in MacInCloud, or rely on dedicated cloud capacity in MacInCloud or MacStadium. These cloud providers not only enable you to build for iOS, they give you access to Xcode and all the other iOS development tools you may want to access.

To get started, after installing and configuring the remote agent on your OS X machine, select either the Remote Device or Simulator -* debug targets in Visual Studio to automatically build, deploy, launch, and even debug your app on an attached iOS device or simulator.

VSO also adds a rich set of engineering features that support iOS development. In addition to collaboration features, VSO (and Team Foundation Server [TFS] 2015) supports a new cross-platform agent that enables anyone to build an iOS app directly on OS X. With it, you can build Cordova-based iOS apps (aka.ms/cordova-vso) or even native Xcode projects.

The VSO cross-platform agent is Node.js-based and uses a simple HTTPS connection to your TFS 2015 server or VSO to fetch work so your OS X machine only needs to have outbound HTTP access to a TFS/VSO instance. The upshot: Cloud providers like MacInCloud or MacStadium can be directly integrated with VSO. You can sign up today to try out these new cross-platform and iOS build capabilities for free at aka.ms/get-vso.

Finally, if you missed the announcement, Visual Studio Code is the new Microsoft editor targeting open source and multi-platform developers. Currently in preview, it runs on OS X, Linux, and Windows, and is available for download at code.visualstudio.com. JavaScript is one of the languages Visual Studio Code supports in the initial release, meaning you can use it to write code for Cordova apps today.

Like Visual Studio 2015, the Visual Studio Code interface reflects the file system directly, so editing Cordova projects is as simple as clicking File | Open Folder and selecting the Cordova project folder. This file system mirroring results in fantastic interoperability between Visual Studio and Visual Studio Code. For example, the Visual Studio Task Runner Explorer allows developers to use the powerful Gulp task runner in Visual Studio, and developers can execute these same tasks from the command line when using Visual Studio Code.

For iOS, being able to edit Cordova projects in Visual Studio Code on OS X means developers can troubleshoot particularly sticky problems by opening up a native iOS project generated by the "cordova prepare ios" Cordova CLI command (aka.ms/cordova-cli) from the platforms/ios folder in Xcode. After troubleshooting, developers can then make edits in Visual Studio Code, build using the Cordova CLI or other command-line tools, and commit changes back into source control. Developers using Visual Studio on Windows can directly pull down these changes, as well.

Only the Beginning

By now you should be sensing a trend. iOS development is a big priority for us here at Visual Studio, as we strive to make our IDE the premier multi-platform development suite of tools in the market. Across the company, Microsoft is embracing iOS development and you'll see continued improvements across Visual Studio, Xamarin, Visual Studio Online, and Visual Studio Code, all designed to bring the flexibility and power of cross-platform development to everyone.

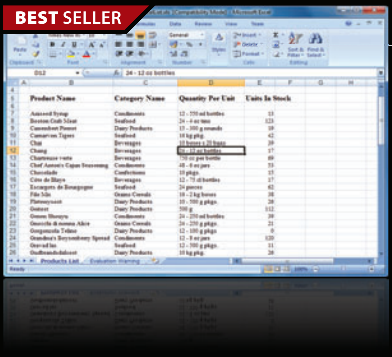
Check out Visual Studio 2015, VSO and Visual Studio Code now and let us know what you think. ■

CHUCK LANTZ *is senior program manager in the Visual Studio Client Tools Team at Microsoft and spent more than 15 years as a developer, advocate and architect in a variety of mid- and large-scale enterprise IT shops. He brought his passion for app development to Microsoft in 2012 and is currently focused on cross-platform mobile app development. Follow him at twitter.com/chuxel.*

ComponentSource®

/update/2015/08

Experience the brand new look at **www.componentsource.com**

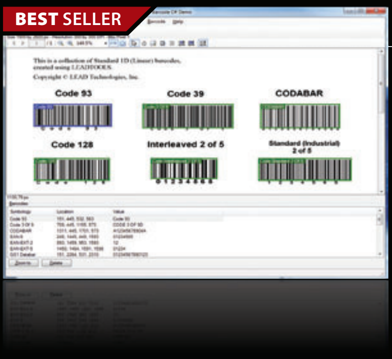

BEST SELLER

Aspose.Total for .NET | from $2,449.02

ASPOSE
Your File Format APIs

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


BEST SELLER

LEADTOOLS Barcode Pro SDK V19 | from $1,295.00 SRP

LEAD TECHNOLOGIES

Add powerful Barcode detection, reading/writing functionality to desktop, tablet & mobile apps.

- Supports 100+ 1D linear barcode types including UPC, EAN, Code 128, 2 of 5 & GS1 DataBar
- Supports 2D Barcode types including QR Code, Data Matrix, Aztec, PDF417 & MicroPDF417
- Load, save, view & convert 150+ raster image formats including JPEG, JPEG 2000, PNG & TIFF
- Over 200 image processing functions for enhanced recognition accuracy
- Includes native libraries for .NET, C/C++, WinRT, iOS, OS X, Android, Linux, Silverlight & more

BEST SELLER

ActiveReports 9 | from $1,567.02

GrapeCity.

Award-winning .NET reporting platform for HTML5, WPF, WinForms, ASP.NET & Windows Azure.

- Sophisticated, fast, and powerful reports
- Visual Studio-integrated report designer
- Extensible optional report server with built-in scalability
- Responsive HTML5 Report Portal
- Royalty-free redistribution

BEST SELLER

Help & Manual Professional | from $583.10

ec software

Help and documentation for .NET and mobile applications.

- 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 responsive HTML, CHM, PDF, MS Word, ePUB, Kindle or print
- Styles and Templates give you full design control

We accept purchase orders.
Contact us to apply for a credit account.

US Headquarters
ComponentSource
650 Claremore Prof Way
Suite 100
Woodstock
GA 30188-5188
USA

European Headquarters
ComponentSource
2 New Century Place
East Street
Reading, Berkshire
RG1 4ET
United Kingdom

Asia / Pacific Headquarters
ComponentSource
3F Kojimachi Square Bldg
3-3 Kojimachi Chiyoda-ku
Tokyo
Japan
102-0083

Sales Hotline - US & Canada:
(888) 850-9911
www.componentsource.com

MasterCard VISA DISCOVER NETWORK

GSA Schedule
Contract GS-35F-0188R

Games Revolution

I was born in 1983, just as the early video game market famously crashed and left many to believe it was a fad that had run its course. But demand for quality games proved resilient and by the time I entered grade school the Nintendo Entertainment System was a favorite topic of playground debate. Since then, the games industry has become larger and more mainstream each year. In 2014, the sector topped U.S. $80 billion in worldwide revenues; double that of the film industry.

It's no surprise, then, that many of us who grew up with games—and especially those of us who became professional geeks—have dreamt of creating games of our own. And more of us might, if the dream of becoming a ninja-level game developer didn't seem so unattainable.

But the perception—and reality—of game development is changing. Once an exclusive clubhouse for a special caste of programmers, recent years have seen new offerings in hobbyist-friendly game development toolsets throw the doors open. For little or no up-front cost, these toolsets grant you the opportunity to benefit from the experience of elite game-engine programmers, conveniently exposed through WYSIWYG editors, comprehensive APIs, and beginner-friendly languages like C#, JavaScript, and Lua. Many of these toolsets also integrate marketplaces where users can buy, sell, and share game assets such as 3D models, animations, sound effects, and useful scripts. And while games are the primary focus, these toolsets have also lent their talents to advertising, education and multimedia art installations.

The most popular of these toolsets, Unity, combines a cutting-edge, cross-platform game engine (it supports 21 platforms to date), an intuitive GUI for crafting game worlds, scripting in C# and UnityScript (which is similar to JavaScript), and a vibrant marketplace for content, scripts and editor extensions. Using Unity doesn't cost a thing unless you're working as part of an organization with more than $100,000 in revenue or funding per year. On top of that, Unity offers extensive documentation, tutorials and sample projects, as well as benefiting from a rich user community. With so much to offer and so few barriers to entry—technical or financial—it's easy to see why Unity is so popular.

But Unity isn't the only game in town. A popular alternative is Unreal Engine 4 (UE4) from Epic Games, which in previous incarnations was marketed to large game studios (and priced accordingly). UE4 is the first to embrace smaller studios and hobbyist developers by becoming open source and charging only a small royalty on game revenue. The UE4 toolset and workflow are largely similar to Unity's, though it's sometimes said to be more capable and less friendly due to its professional lineage.

Another popular alternative is the 100 percent free-to-use and open source Cocos2D, which has traditionally focused on 2D games and favored code over GUI editors. It is further distinguished by offering a family of similar APIs for different languages—C++, JavaScript, Swift and others—rather than a monolithic platform. While still not as feature-rich as Unity or UE4, Cocos2D has recently gained 3D support and the peripheral Cocos Studio provides some of the functionality of competing editors.

Because these toolsets take care of the technology—rendering, sound, animation, input and more—you're free to concentrate on the creative work of building your game. You don't need to know how to write optimized C++ code—the toolset provides an optimized engine. You don't need to know Direct3D or OpenGL—the toolset provides rendering that you can extend with custom shaders. You don't need to know how to import character models from 3DStudioMax—the toolset can import game assets of all kinds in just about any popular format.

Unleashing creativity with one of these toolsets doesn't require an exotic skillset. In fact, it doesn't take much more than a willingness to learn how its components work together, an understanding of high-school math and modest programming skills. Even if you've never written a game before you'll find a lot of familiar ground. Most of what you do to create a game using these tools is a matter of configuring properties or writing code in the right place. Adding a bad guy to your game is not unlike dropping a button into a Windows Forms app and writing code to handle its OnClick event.

Ryder Donahue penned the first Upstart column in the May issue of *MSDN Magazine* (msdn.microsoft.com/magazine/dn973009), and was quoted in the issue's Editor's Note column as saying: "The tools that are given to developers, and the resources available, empower anyone to really make their dream app a reality." Toolsets like Unity, UE4, Cocos2D and others prove that this observation rings true for aspiring game developers and for other creative developers, as well. More than three decades after the crash of 1983, it's clear that the game dev space is more vibrant—and more accessible—than ever before. ■

MICHAEL THOMPSON *is a content developer at Microsoft who writes about C++, graphics and gaming for the Visual Studio team. He's a DigiPen alumni and a top member at gamedev.net.*

DOMAINS | MAIL | HOSTING | eCOMMERCE | SERVERS

NEXT GENERATION 1&1 CLOUD SERVER

TRY IT FOR 1 MONTH - FREE!

Then starting at $9.99 per month*

Powered by intel® Cloud Technology

Top performer CLOUD SPECTATOR

EASY TO USE – READY TO GO

The new 1&1 Cloud Server offers all the advantages of dedicated hardware performance combined with the flexibility of the cloud!

FLEXIBLE & AFFORDABLE

Customized configuration

- SSD, RAM and CPU can be adjusted independently, flexibly and precisely
- **NEW:** Pre-configured packages available

Windows

Transparent costs

- Billing by the minute
- Clearly structured cost overview enables efficient planning and management
- No minimum contract term

EASY & SECURE

1&1 Cloud Panel

- Innovative, user-friendly interface with smart administration

Security

- Built-in firewall to protect your server from online threats
- Backups and snapshots to prevent accidental data loss
- High-performance 1&1 Data Centers are among the safest in the US

ALL-INCLUSIVE

Best performance

- Unlimited traffic
- Premium SSD with the highest performance
- Private networks, professional API, load balancers, firewalls and more-all easy to configure
- Ready-to-use applications including WordPress, Drupal™ and Magento®

1 TRIAL TRY FOR 30 DAYS ★★★★★

1 MONTH FLEXIBLE PAYMENT OPTIONS ★★★★★

1 CALL SPEAK WITH AN EXPERT 24/7 ★★★★★

☎ 1 (877) 461-2631

1&1®

1and1.com

*1&1 Cloud Server is available free for one month, after which regular price of $9.99/month applies. No setup fee is required. Visit 1and1.com for full offer details, terms and conditions. 1&1 Cloud Server powered by Intel® Xeon® Processor E5-2683 (35 Cache, 2.00 Ghz). Intel, the Intel Logo, Intel Inside, the Intel Inside logo, Intel. Experience What's Inside are trademarks of Intel Corporation in the U.S. and/or other countries. 1&1 and the 1&1 logo are trademarks of 1&1 Internet, all other trademarks are property of their respective owners. ©2015 1&1 Internet. All rights reserved.

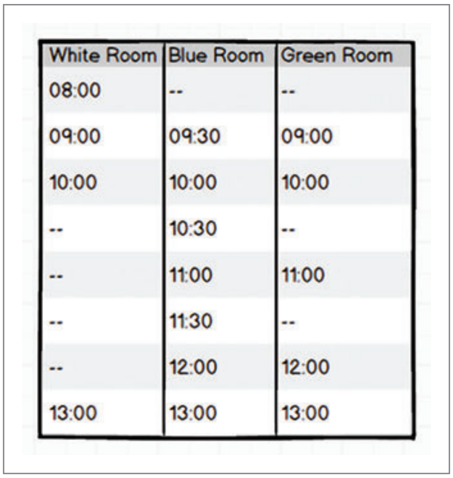
Cutting Edge

DINO ESPOSITO

CQRS and Events: A Powerful Duo

As with many aspects of modern software, nothing is really new. Things are often relabeled and presented with new and compelling buzzwords. Command Query Responsibility Segregation (CQRS) is a great example. Domain events that describe observable changes in the business domain are another.

While defining the Eiffel programming language back in the late 1980s, Bertrand Meyer formalized the principle at the foundation of what is today called CQRS. In software, any basic action can be either a command or a query—but never both. If it's a command, it's expected to alter the state of the system. If it's a query, it's expected to report about the state of the system without altering it in any way.



White Room	Blue Room	Green Room
08:00	--	--
09:00	09:30	09:00
10:00	10:00	10:00
--	10:30	--
--	11:00	11:00
--	11:30	--
--	12:00	12:00
13:00	13:00	13:00

Figure 1 **A Mockup for the Rooms Timesheet**

In other words, asking a question shouldn't change the answer. This is sometimes referred to as the Command Query Separation (CQS) principle. That's fine for commands and queries, but what about domain events?

Events in Business Applications

Since the early days of software engineering, architects have designed line-of-business applications able to survive changes in business and track those changes. To support business intelligence and statistical analysis, architects have also managed to make sequences of actions repeatable to some extent. They didn't call them "domain events" or coin cool terms such as "event sourcing" right away, yet the concept of domain events has been used extensively for years in business systems. This is especially true for accounting, finance and banking systems.

Then we had the Domain Driven Design (DDD) revolution and lost in this "new" universal software model, I think everyone missed some perspective on software architecture. Developers mostly focused on layers and disregarded vertical segments of systems such as command and query stacks.

In my view, the difference between the CQS principle formalized by Bertrand Meyer and today's CQRS principle is all in where it's applied. CQS is more of a universal principle for software development. CQRS touches on the system architecture. When you apply CQS at the architecture level, you have CQRS.

I first presented the entry-level version of CQRS in the article, "CQRS for the Common Application" (msdn.microsoft.com/magazine/mt147237). I went a bit deeper in the subsequent article, "CQRS and Message-Based Applications" (msdn.microsoft.com/magazine/mt238399).

These articles described the command stack based on messages being exchanged between the presentation and business logic via the application layer. I briefly touched on business events you can save to an ad hoc store to log relevant business items, such as a product order or cancellation.

The focus was more on using messages to implement business workflows triggered by commands. In that context, events were just cross-handler notifications of recent occurrences to which handlers may need to react. This is the first step of a longer evolution aimed at transitioning software architects from the idea of "models-to-persist" to the idea of "events-to-log." I see CQRS as the starting point of a change that will have a profound impact on system architecture.

I see CQRS as the starting point of a change that will have a profound impact on system architecture.

The Apple of Sir Isaac Newton and I

You've heard the story of the apple that fell on Sir Isaac Newton's head that led him to formulate the Universal Law of Gravity. It's probably more legend than fact, but it's true that many discoveries start with a curious event. I recently had my own version of an apple falling on my head that made me think seriously about CQRS and events. I was doing some preliminary analysis for rebuilding a system a customer has been using for about five years.

DynamicPDF Converter
DynamicPDF Rasterizer
DynamicPDF Core Suite
DynamicPDF Core Suite

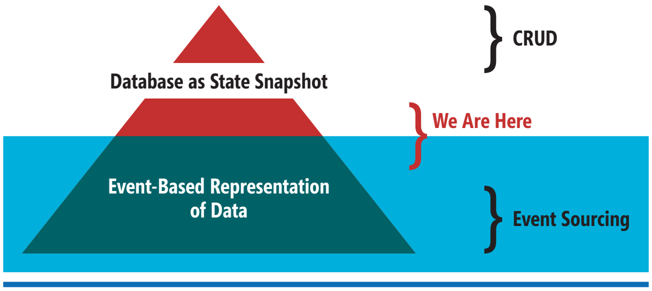

CRUD
Database as State Snapshot
We Are Here
Event-Based Representation of Data
Event Sourcing

Figure 2 **The Underwater Iceberg of Events**

The customer's system was an ASP.NET MVC 2 Web site with a plain SQL Server back-end database. The data access layer was centered on a relational database consumed via Entity Framework. Inferred entity classes were extended with additional, business-specific methods using the partial class mechanism of the Microsoft .NET Framework. Cross-entities behavior was delegated to domain services. Although I wouldn't call it a reference implementation of the Domain Model pattern, it had bits and pieces of DDD facts here and there. Generally, I'd call it a plain Create, Read, Update, Delete (CRUD) system with some amount of business logic to sit as a buffer between the back end and the rest of the system.

In software, a CRUD system is an application built around a bunch of database tables. The layers surrounding the database validate and lay the groundwork for the core CRUD operations against the database. That was the system.

Another aspect of a CRUD system that often goes unnoticed, or simply doesn't get enough attention, is it uses the database as a snapshot. At any time, the database behind a CRUD system stores the last known good system state. Sometimes this is enough, sometimes not.

The apple that fell on my head made me realize a snapshot database approach was OK for version one. With version two, the customer wants more. Now the customer wants to be able to track history and repeat sequences of steps under different conditions to determine the outcome. When it comes to this, the snapshot database as the sole (or primary) form of storage is no longer appropriate.

A Sample Booking Application

Consider an application that lets users book a meeting room. At some point, the presentation layer will display a screen similar to **Figure 1**.

Imagine each room has its own business rules concerning booking. For example, users can book the White room for one hour from 8 to 10 and then from 13 onward. The Blue room has 30-minute time slots available from 9:30 in the morning. Finally, the Green room is available starting at 9 a.m. for one-hour increments. These details are saved and matched to existing bookings. You can generate a grid where you can gray out slots already taken and make clickable slots available.

The rules governing room availability aren't set in stone and may indeed change. No big deal, you say. If it changes, let the admin edit the rules accordingly. This approach works nicely as long as you're using the mockup of **Figure 1** to present the latest system state.

The moment the customer asks to be able to navigate between the current grid and the grid three weeks before, you're lost. More precisely, you're lost if any changes to the booking business rules happen within the past three weeks. You know the current slots and times for each room, but you've lost the details of what it was three weeks before.

When the customer brought this up, I thought fixing it was simply a matter of bringing some missing information to the table. All I have to do is save some rules and read them back when needed. As simple as it may sound, properly addressing such a small piece of missing information requires a deep architectural change. It's like the tip of the iceberg—and the iceberg is switching from a snapshot database to event-based data representation.

Event sourcing refers to event-based data representation. I won't cover event sourcing in-depth here, but that's precisely the topic for the next column. For the rest of this article, I'll discuss a mixed approach that weds database snapshots and business event logs. It's probably a good first step to transition existing classic architectures to the future. More than any fancy new technology, framework, release and runtime, the big thing of the next few years is primarily architectural—event-based data as the primary data source instead of data snapshots (as shown in **Figure 2**).

Beyond Data Snapshots

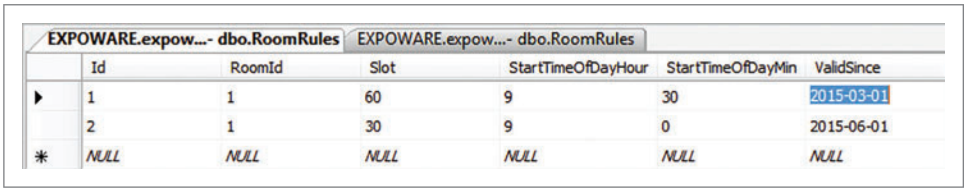
For decades, most of us built systems concerned only with the latest current system state. In this scenario, databases stored entity snapshots (or aggregates if you feel more comfortable with the DDD terminology) and it worked beautifully. If you've never felt the need to log business events as they occur, then you probably won't.

The moment the customer requests a feature for which the application clock must be moved backward to a specific time, though, you're lost. The only way out is to re-architect the system to save business events. Suppose now you have a table similar to the one in **Figure 3**.

This table indicates you can book room 1 according to different rules between March 1 and June 1. It also means you must redesign the **Figure 1** grid according to the date to which it refers.

The record in the RoomRules table is logically equivalent to a business event, in that it indicates an event occurred to change the booking rule. In terms of implementation, though, it's a plain record in an additional table and requires a bunch of extra queries (one per room) before populating the grid of **Figure 1**.

A new booking rule just requires a new record to the table. Diligent developers, however, would probably already have some sort of rules table. They would provide one row per room. In this context, the only difference seems to be an extra column called ValidSince in **Figure 3**.


EXPOWARE.expow...- dbo.RoomRules
EXPOWARE.expow...- dbo.RoomRules

	Id	RoomId	Slot	StartTimeOfDayHour	StartTimeOfDayMin	ValidSince
▶	1	1	60	9	30	2015-03-01
	2	1	30	9	0	2015-06-01
*	NULL	NULL	NULL	NULL	NULL	NULL

Figure 3 **A Sample Table Storing Booking Rules for Rooms**

Coming from a classic snapshot storage vision, using the word "business event" to define a bunch of additional records in an additional table might seem like overkill. Yet this is precisely the perspective you get when you're right on the waterline with snapshots above you and the entire world of event sourcing just beneath.

In more specific terms, the rules for booking rooms are events and logging them is tracking the history of the system. Getting the rules to apply to a given room is "replaying" the events for that room that have occurred since the system started and up to a given point. You may not have the current status saved somewhere, but you have all events logged. To get the system state, just replay events and build a valid instance of the Room entity.

Does this sound weird? This is how any banking software calculates your current "balance." The balance never comes out of a plain read. More likely, it's a snapshot persisted to a given date and replaying within all subsequent events.

Events and CQRS Together

Events empower software beyond imagination. Even plain CRUD systems are touched as soon as customers request simple forms of business intelligence and statistical analysis. Events are immutable. Because of that, you can easily replicate event databases to new instances of software. This is a key statement of scalability. At the same time, having command and queries separated (and thus distinct stacks for processing commands and executing queries) lets you focus on events in the command stack and have snapshot tables ready for quick reads in the query stack.

If you have performance concerns, you should save events when you run a command and update all projections of data you need for queries. For example, when you add a new booking rule, from now on that's the default. You can update an additional current settings table you can quickly query without replaying events. You can then go back and replay events as required.

Wrapping Up

With events, you don't miss a thing. You still have your business run and start teaching yourself a more powerful type of architecture. By storing data as events, you take one step down the data abstraction level—meaning you can use the same amount of lower-level data to build any number of projections and snapshots for many business purposes such as plain queries, business intelligence, what-if analysis, statistics, usage analytics and anything else you can imagine. ■

Dino Esposito *is the coauthor 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 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*

Windows with C++

KENNY KERR

Windows Runtime Components with MIDL

In my July 2015 column (msdn.microsoft.com/magazine/mt238401), I introduced the concept of Windows Runtime (WinRT) components as the evolution of the COM programming paradigm. Whereas Win32 put COM on the side, the Windows Runtime puts COM front and center. The Windows Runtime is the successor to Win32, the latter being an umbrella term for the Windows API as it encompasses many disparate technologies and programming models. The Windows Runtime provides a consistent and unified programming model, but in order for it to succeed developers both inside and outside of Microsoft need better tools to develop WinRT components and to use those components from within apps.

The primary tool provided by the Windows SDK to serve this need is the MIDL compiler. In the July column, I showed how the MIDL compiler could produce the Windows Runtime Metadata (WINMD) file most language projections require to consume WinRT components. Of course, any longtime developer on the Windows platform will be aware that the MIDL compiler also produces code that a C or C++ compiler can consume directly. In fact, MIDL itself knows nothing about the WINMD file format. It's principally about parsing IDL files and producing code for C and C++ compilers in order to support COM and remote procedure call (RPC) development and the production of proxy DLLs. The MIDL compiler is such a historically critical piece of machinery that the engineers who developed the Windows Runtime chose not to take the risk of breaking it and instead developed a "sub compiler" responsible only for the Windows Runtime. Developers are not usually aware of this subterfuge—and need not be—but it helps explain the way the MIDL compiler works in practice.

Let's look at some IDL source code and see what's really going on with the MIDL compiler. Here's an IDL source file that defines a classic COM interface:

```
C:\Sample>type Sample.idl
import "unknwn.idl";

[uuid(e21df825-937d-4b0b-862e-e411b57e280e)]
interface IHen : IUnknown
{
  HRESULT Cluck();
}
```

Classic COM doesn't have a strong notion of namespaces so the IHen interface is simply defined at file scope. The definition of IUnknown must also be imported prior to use. I can then just pass this file through the MIDL compiler to produce a number of artifacts:

```
C:\Sample>midl Sample.idl

C:\Sample>dir /b
dlldata.c
Sample.h
Sample.idl
Sample_i.c
Sample_p.c
```

The dlldata.c source file contains a few macros that implement the necessary exports for a proxy DLL. The Sample_i.c contains the IHen interface's GUID, should you be using a 25-year-old compiler that lacks support for the uuid __declspec that attaches GUIDs to types. Then there's Sample_p.c, which contains the marshaling instructions for the proxy DLL. I'll gloss over these for the moment and instead focus on Sample.h, which contains something quite handy. If you overlook all of the horrible macros intended to help C developers use COM (the horror!) you'll find this:

```
MIDL_INTERFACE("e21df825-937d-4b0b-862e-e411b57e280e")
IHen : public IUnknown
{
public:
  virtual HRESULT STDMETHODCALLTYPE Cluck( void) = 0;

};
```

It's not elegant C++ but, after preprocessing, it amounts to a C++ class that inherits from IUnknown and adds a pure virtual function of its own. This is handy because it means you don't have to write this by hand, possibly introducing a mismatch between the C++ definition of the interface and the original IDL definition that other tools and languages might consume. So that's the essence of what the MIDL compiler provides for C++ developers, producing a translation of the IDL source code in such a way that a C++ compiler can consume those types directly.

Now let's return to the Windows Runtime. I'll update the IDL source code just slightly to comply with the more stringent requirements for WinRT types:

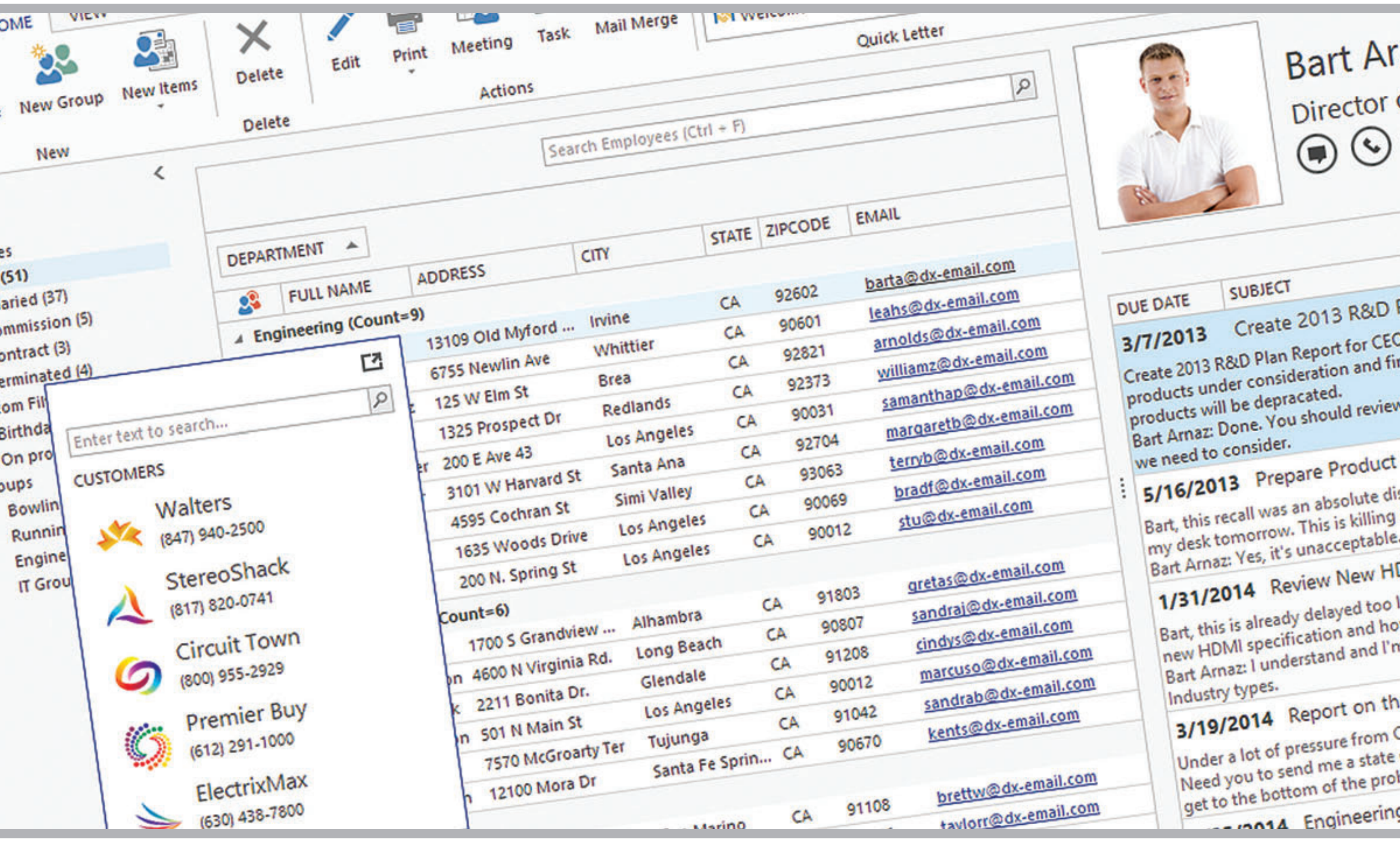
```
C:\Sample>type Sample.idl
import "inspectable.idl";

namespace Sample
{
  [uuid(e21df825-937d-4b0b-862e-e411b57e280e)]
  [version(1)]
  interface IHen : IInspectable
  {
    HRESULT Cluck();
  }
}
```

WinRT interfaces must inherit directly from IInspectable, and a namespace is used in part to associate the types with the implementing component. If I try to compile it as before, I run into a problem:

```
.\Sample.idl(3) : error MIDL2025 : syntax error : expecting an
interface name or DispatchInterfaceName or CoclassName or ModuleName or
LibraryName or ContractName or a type specification near "namespace"
```

The MIDL compiler doesn't recognize the namespace keyword and gives up. That's what the /winrt command-line option is for. It tells the MIDL compiler to pass the command line directly to the MIDLRT compiler to preprocess the IDL source file. It is this second compiler—MIDLRT—that expects the /metadata_dir command-line option I mentioned in the July column:

```
C:\Sample>midl /winrt Sample.idl /metadata_dir
  "C:\Program Files (x86)\Windows Kits ..."
```



es
(51)
aried (37)
ommission (5)
ontract (3)
erminated (4)
om Fil
Birthda
On pro
oups
Bowlin
Runnin
Engine
IT Grou

Engineering (Count=9)

ADDRESS	CITY	STATE	ZIPCODE	EMAIL
13109 Old Myford ...	Irvine	CA	92602	barta@dx-email.com
6755 Newlin Ave	Whittier	CA	90601	leahs@dx-email.com
125 W Elm St	Brea	CA	92821	arnolds@dx-email.com
1325 Prospect Dr	Redlands	CA	92373	williamz@dx-email.com
200 E Ave 43	Los Angeles	CA	90031	samanthap@dx-email.com
3101 W Harvard St	Santa Ana	CA	92704	margaretb@dx-email.com
4595 Cochran St	Simi Valley	CA	93063	terryb@dx-email.com
1635 Woods Drive	Los Angeles	CA	90069	bradf@dx-email.com
200 N. Spring St	Los Angeles	CA	90012	stu@dx-email.com
Count=6)				
1700 S Grandview ...	Alhambra	CA	91803	gretas@dx-email.com
4600 N Virginia Rd.	Long Beach	CA	90807	sandraj@dx-email.com
2211 Bonita Dr.	Glendale	CA	91208	cindys@dx-email.com
501 N Main St	Los Angeles	CA	90012	marcuso@dx-email.com
7570 McGroarty Ter	Tujunga	CA	91042	sandrab@dx-email.com
12100 Mora Dr	Santa Fe Sprin...	CA	90670	kents@dx-email.com
	...Marino	CA	91108	brettw@dx-email.com
				taylorr@dx-email.com

CUSTOMERS	
Walters	(847) 940-2500
StereoShack	(817) 820-0741
Circuit Town	(800) 955-2929
Premier Buy	(612) 291-1000
ElectrixMax	(630) 438-7800

Bart Ar
Director

DUE DATE	SUBJECT
3/7/2013	Create 2013 R&D
5/16/2013	Prepare Product
1/31/2014	Review New HD
3/19/2014	Report on th
	Engineering

Create 2013 R&D Plan Report for CEC products under consideration and fi products will be depracated. Bart Arnaz: Done. You should review we need to consider.

Bart, this recall was an absolute dis my desk tomorrow. This is killing Bart Arnaz: Yes, it's unacceptable.

Bart, this is already delayed too l new HDMI specification and ho Bart Arnaz: I understand and I'n Industry types.

Under a lot of pressure from C Need you to send me a state get to the bottom of the pro

As further evidence of this, take a closer look at the MIDL compiler output and you'll see what I mean:

```
C:\Sample>midl /winrt Sample.idl /metadata_dir "..."
Microsoft (R) 32b/64b MIDLRT Compiler Engine Version 8.00.0168
Copyright (c) Microsoft Corporation. All rights reserved.
MIDLRT Processing .\Sample.idl
.
.
.
Microsoft (R) 32b/64b MIDL Compiler Version 8.00.0603
Copyright (c) Microsoft Corporation. All rights reserved.
Processing C:\Users\Kenny\AppData\Local\Temp\Sample.idl-34587aaa
.
.
.
```

I've removed some of the processing of dependencies in order to highlight the key points. Calling the MIDL executable with the /winrt options blindly passes the command line to the MIDLRT executable before exiting. MIDLRT parses the IDL first to generate the WINMD file, but it also produces another temporary IDL file. This temporary IDL file is a translation of the original with all of the WinRT-specific keywords, such as namespaces, replaced in such a way that the original MIDL compiler will accept it. MIDLRT then calls the MIDL executable again but without the /winrt option and with the location of the temporary IDL file so it can produce the original set of C and C++ headers and source files as before.

The namespace in the original IDL file is removed and the name of the IHen interface is decorated in the temporary IDL file as follows:

```
interface __x_Sample_CIHen : IInspectable
.
.
.
```

This is actually an encoded form of the type name that's interpreted by the MIDL compiler given the /gen_namespace command-line option that MIDLRT uses when calling MIDL with the preprocessed output. The original MIDL compiler can then process this directly without specific knowledge of the Windows Runtime. This is just one example but it gives you an idea of how the new tooling makes the most of existing technology to get the job done. If you're curious to see how this works, you might poke around the temporary folder the MIDL compiler traces out, only to find that those files—Sample.idl-34587aaa in the previous example—are missing. The MIDLRT executable is careful to clean up after itself, but if you include the /savePP command-line option, MIDL won't delete these temporary preprocessor files. Anyway, a bit more preprocessing is thrown in and the resulting Sample.h now has something that even a C++ compiler will recognize as a namespace:

```
namespace Sample {

  MIDL_INTERFACE("e21df825-937d-4b0b-862e-e411b57e280e")
  IHen : public IInspectable
  {
  public:
    virtual HRESULT STDMETHODCALLTYPE Cluck( void) = 0;

  };

}
```

I can then implement this interface as before, confident that the compiler will pick up any discrepancies between my implementation and the original definitions that I coded in IDL. On the other hand, if you only need MIDL to produce the WINMD file and don't need all of the source files for a C or C++ compiler, you can avoid all of the extra build artifacts with the /nomidl command-line option. This option is passed, along with all the rest, by the MIDL executable to the MIDLRT executable. MIDLRT then skips the last step of calling MIDL again after it finishes producing the WINMD file. It's also customary, when using a Windows Runtime ABI produced by MIDL, to include the /ns_prefix command-line option so that resulting types and namespaces are enclosed within the "ABI" namespace as follows:

```
namespace ABI {
  namespace Sample {

    MIDL_INTERFACE("e21df825-937d-4b0b-862e-e411b57e280e")
    IHen : public IInspectable
    {
    public:
      virtual HRESULT STDMETHODCALLTYPE Cluck( void) = 0;

    };

  }
}
```

Finally, I should mention that neither MIDL nor MIDLRT is enough to produce a self-contained WINMD file that sufficiently describes a component's types. If you happen to reference external types, usually other types defined by the OS, the WINMD file, produced by the process described thus far, must still be merged with the main metadata file for the version of Windows you're targeting. Let me illustrate the problem.

I'll start with an IDL namespace describing both an IHen interface and an activatable Hen class that implements this interface, as shown in **Figure 1**.

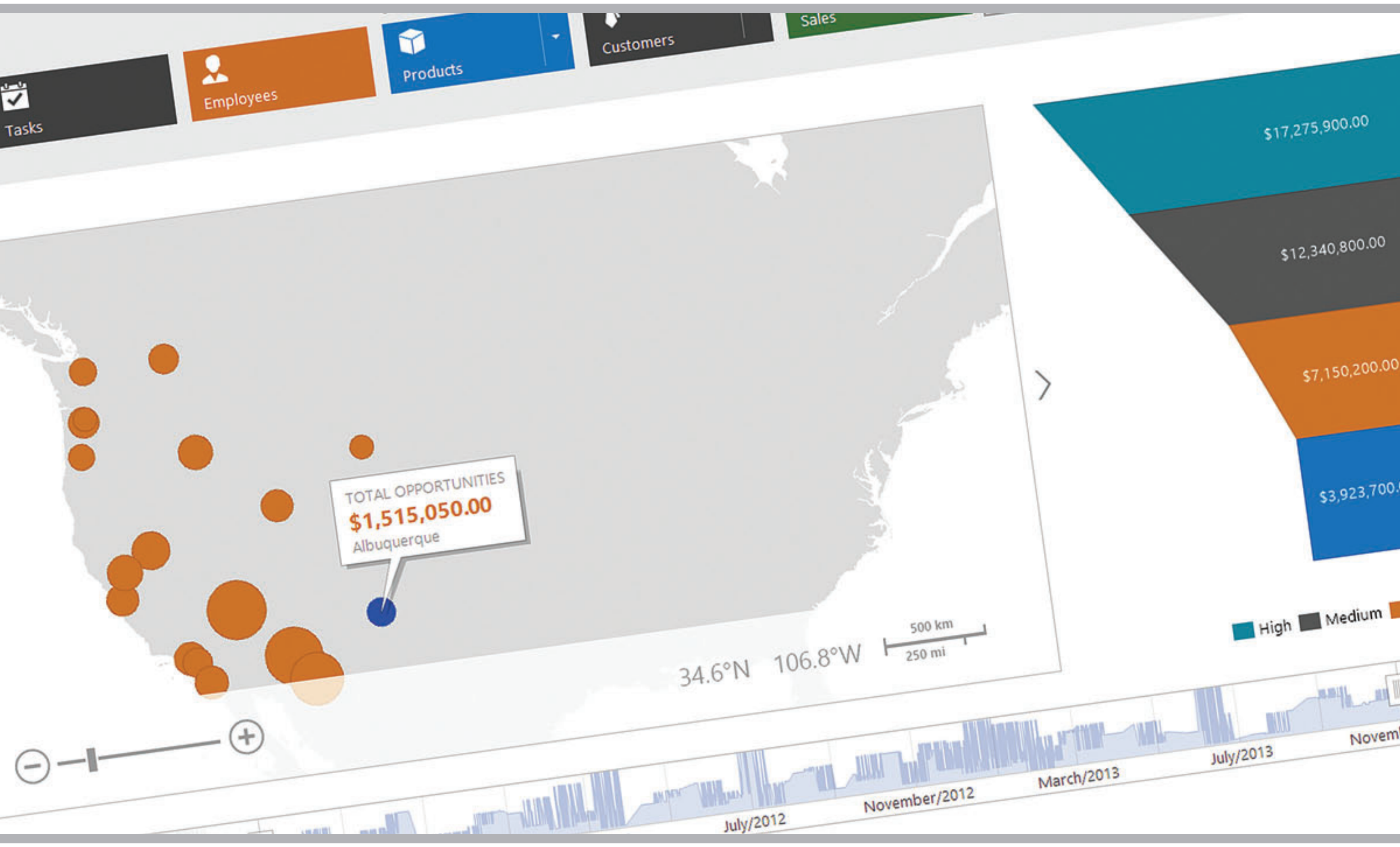
I'll then implement it using the same technique I described in the July column, except that now I can rely on the definition of IHen as provided by the MIDL compiler. Now, inside a WinRT app, I can simply create a Hen object and call the Cluck method. I'll use C# to illustrate the app side of the equation:

```
public void SetWindow(CoreWindow window)
{
  Sample.IHen hen = new Sample.Hen();
  hen.Cluck();
}
```

The SetWindow method is part of the IFrameworkView implementation provided by the C# app. (I described IFrameworkView in my August 2013 column, which you can read at msdn.microsoft.com/magazine/jj883951.) And, of course, this works. C# is utterly dependent on the WINMD metadata describing the component. On the other hand, it sure makes sharing native C++ code with C# clients a breeze. Most of the time, anyway. One problem that arises is if you reference external types, as I alluded to a moment ago. Let's update the Cluck

Figure 1 **The Hen Class in IDL**

```
namespace Sample
{
  [uuid(e21df825-937d-4b0b-862e-e411b57e280e)]
  [version(1)]
  interface IHen : IInspectable
  {
    HRESULT Cluck();
  }

  [version(1)]
  [activatable(1)]
  runtimeclass Hen
  {
    [default] interface IHen;
  }
}
```


Tasks
Employees
Products
Customers
Sales
TOTAL OPPORTUNITIES
$1,515,050.00
Albuquerque
34.6°N 106.8°W
500 km
250 mi
$17,275,900.00
$12,340,800.00
$7,150,200.00
$3,923,700.
High
Medium
July/2012
November/2012
March/2013
July/2013
Novem

msdn magazine VP

MSDN MAGAZINE VENDOR PROFILE

RAVENDB

RavenDB OLTP Made Simple

RavenDB is the premier NoSQL database for .NET. Open source, speed-obsessed, and a joy to use.

What is Hibernating Rhinos?

Hibernating Rhinos LTD is the corporate alter ego of Oren Eini (AKA Ayende@Rahien), its founder, recognized as a leading global authority in open source NoSQL and document databases' architecture and design. He and his team are the creators of RavenDB, a leading OLTP document database offering.

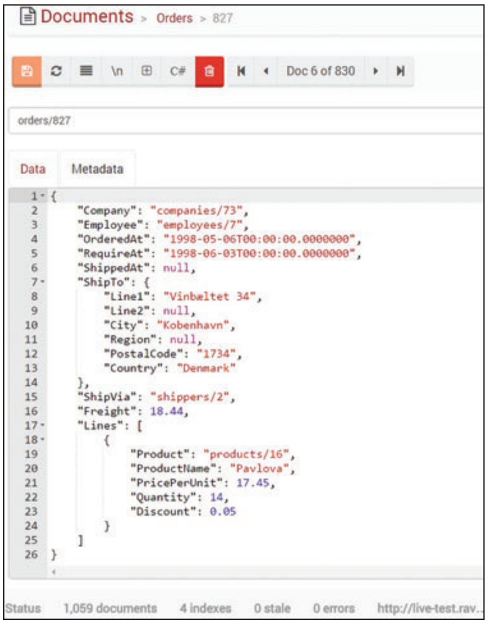
Why RavenDB?

This outstanding database solution provides high availability, high performance, zero administration and self-optimization, resulting in an open source database system trusted and employed by a global client base of Fortune 500 Companies, supported by a global network of RavenDB Certified Affiliates.

What Makes RavenDB Unique

Featuring one of the most robust replication mechanisms in the industry, a RavenDB cluster is comprised of multiple nodes, whether running in a single data center, or geo-distributed around the globe, guaranteeing all data is replicated in all nodes enabling instantaneous switching to a secondary node in the event of primary node failure. This affords unprecedented zero downtime performance.

All data transactions are ACID (Atomicity, Consistency, Isolation, and Durability); this in contrast to competing database offerings which often lack one or more of these attributes. Your data is absolutely safe with RavenDB.



```
{
    "Company": "companies/73",
    "Employee": "employees/7",
    "OrderedAt": "1998-05-06T00:00:00.0000000",
    "RequireAt": "1998-06-03T00:00:00.0000000",
    "ShippedAt": null,
    "ShipTo": {
        "Line1": "Vinbæltet 34",
        "Line2": null,
        "City": "Kobenhavn",
        "Region": null,
        "PostalCode": "1734",
        "Country": "Denmark"
    },
    "ShipVia": "shippers/2",
    "Freight": 18.44,
    "Lines": [
        {
            "Product": "products/16",
            "ProductName": "Pavlova",
            "PricePerUnit": 17.45,
            "Quantity": 14,
            "Discount": 0.05
        }
    ]
}
```

RavenDB's astonishingly low administrative overhead is owed to its built in ability to assess and know what current load patterns are. Using an advanced set of automated heuristics, it automatically and transparently configures itself, in real time, based on actual production loads and conditions. It will recognize high load demand conditions and shift resources between background processes and client requests, balances loads and latencies in real time to maximize overall performance, without requiring human intervention.

RavenDB reduces the total cost of ownership by allowing users to make the most of client hardware and personnel resources, while maximizing automated operational efficiency.

Zero Barrier to Implementation

Initial install takes five minutes; moving from development to production is easy, fast and painless, RavenDB will auto configure itself based on your cluster topology and configuration on the fly. There is no need for major complex production release procedures.

Don't take our word for it- go to our live instance at **http://live-test.ravendb.net** and just start playing. Give RavenDB a shot. We promise you will be amazed.

For More Information → http://ravendb.net/

method to require a CoreWindow as an argument. CoreWindow is defined by the OS so I can't simply define it inside my IDL source file.

First, I'll update the IDL to take a dependency on the ICoreWindow interface. I'll simply import the definition as follows:

```
import "windows.ui.core.idl";
```

And then I'll add an ICoreWindow parameter to the Cluck method:

```
HRESULT Cluck([in] Windows.UI.Core.ICoreWindow * window);
```

The MIDL compiler will turn that "import" into a #include of windows.ui.core.h inside the header that it generates so all I need to do is update my Hen class implementation:

```
virtual HRESULT __stdcall Cluck(ABI::Windows::UI::Core::ICoreWindow *)
  noexcept override
{
  return S_OK;
}
```

I can now compile the component as before and ship it to the app developer. The C# app developer dutifully updates the Cluck method call with a reference to the app's CoreWindow as follows:

```
public void SetWindow(CoreWindow window)
{
  Sample.IHen hen = new Sample.Hen();
  hen.Cluck(window);
}
```

Unfortunately, the C# compiler now complains:

```
error CS0012: The type 'ICoreWindow' is defined in an assembly
  that is not referenced.
```

You see, the C# compiler doesn't recognize the interfaces as being the same. The C# compiler isn't satisfied with just a matching type name and can't make the connection with the Windows type of the same name. Unlike C++, C# is very much dependent on binary type information to connect the dots. To solve this problem, I can employ another tool provided by the Windows SDK that will compose or merge the metadata from the Windows OS along with my component's metadata, correctly resolving the ICoreWindow to the main metadata file for the OS. This tool is called MDMERGE:

```
c:\Sample>mdmerge /i . /o output /partial /metadata_dir "..."
```

The MIDLRT and MDMERGE executables are quite particular about their command-line arguments. You need to get them just right in order for it to work. In this case, I can't simply update Sample.winmd in place by pointing the /i (input) and /o (output) options to the same folder because MDMERGE actually deletes the input WINMD file upon completion. The /partial option tells MDMERGE to look for the unresolved ICoreWindow interface in the metadata provided by the /metadata_dir option. This is called the reference metadata. MDMERGE can thus be used to merge multiple WINMD files but in this case, I'm just using it to resolve references for OS types.

At this point, the resulting Sample.winmd correctly points to the metadata from the Windows OS when referring to the ICoreWindow interface and the C# compiler is satisfied and will compile the app as written. Join me next month as I continue to explore the Windows Runtime from C++. ■

Kenny Kerr *is a computer programmer based in Canada, as well as an author for Pluralsight and a Microsoft MVP. He blogs at kennykerr.ca and you can follow him on Twitter at twitter.com/kennykerr.*

Thanks to the following Microsoft technical expert for reviewing this article: *Larry Osterman*

Modern Drag and Drop for Universal Windows Apps

Anna Pai and Alain Zanchetta

Drag and drop is an intuitive way to transfer data within an application or between applications on the Windows desktop. It made its debut in Windows 3.1 with the File Manager and was then extended to all applications supporting OLE 2, such as Microsoft Office. When Windows 8 was released, Microsoft introduced a new type of Windows application, called a Windows Store app, which was designed for tablets and always displayed in full-screen mode. Because only one application was visible at any given time, cross-application drag and drop didn't make sense and other ways of sharing data, such as the Share Charm, were developed. In Windows 10, however, applications on a desktop PC are once again running in windowed mode, which means multiple windows are displayed on the screen, and therefore drag and drop makes a come-back as a Universal Windows app API, with the addition of new features that enhance the UX.

This article is based on the public preview of Windows 10 and Visual Studio 2015.

This article discusses:

- Drag-and-drop concepts
- Implementing drag sources and drop targets
- Customizing visual feedback
- Using asynchronous operations

Technologies discussed:

Windows 10, XAML, Universal Windows Apps

Code download available at:

msdn.microsoft.com/magazine/msdnmag0815

Drag-and-Drop Concepts

Drag and drop lets the user transfer data between applications or within an application using a standard gesture (press-hold-and-pan with the finger or press-and-pan with a mouse or a stylus).

The drag source, which is the application or area where the drag gesture is triggered, provides the data to be transferred by filling a data package object that can contain standard data formats, including text, RTF, HTML, bitmaps, storage items or custom data formats. The source also indicates the kind of operations it supports: copy, move or link. When the pointer is released, drop occurs. The drop target, which is the application or area underneath the pointer, processes the data package and returns the type of operation it performed.

During drag and drop, the drag UI provides a visual indication of the type of drag-and-drop operation that's taking place. This visual feedback is initially provided by the source but can be changed by the targets as the pointer moves over them.

Modern drag and drop is available on desktop, tablet and phone. It allows data transfer between or within any kind of application, including Classic Windows apps, although this article focuses on the XAML API for modern drag and drop.

Implementing Drag and Drop

The drag source and drop target play different roles. An application can have UI components that are only drag sources, only drop targets or both, such as the sample Photo Booth application shown in **Figure 1**.

A drag-and-drop operation is driven entirely by user input, so its implementation is almost exclusively event-based, as shown in **Figure 2**.

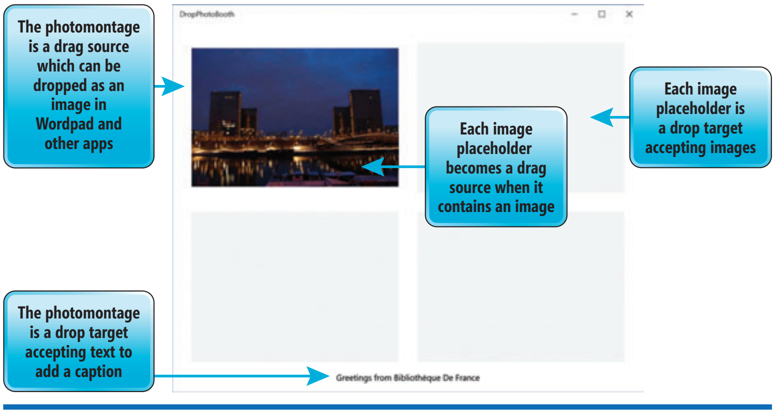

The photomontage is a drag source which can be dropped as an image in Wordpad and other apps
DropPhotoBooth
Each image placeholder becomes a drag source when it contains an image
Each image placeholder is a drop target accepting images
The photomontage is a drop target accepting text to add a caption
Greetings from Bibliothèque De France

Figure 1 **Drag Sources and Drop Targets**

Implementing a Drag Source In Windows 8.1, a ListView can be the source of an in-app drag-and-drop operation if its CanDragItems property is set to true:

```
<ListView CanDragItems="True"
          DragItemsStarting="ListView_DragItemsStarting"
          ItemsSource="{Binding Pictures}"
          ItemTemplate="{StaticResource PictureTemplate}"
          />
```

The application can handle the DragItemsStarting event on the source. This is still supported in Windows 10 with the addition of a DragItemsCompleted event, which wasn't needed in Windows 8.1 applications where target and source must belong to the same process.

The main drag source for modern drag and drop is the UIElement, which gives access to all modern drag-and-drop features and is the main focus of this article.

One way to make UIElement draggable is to set its CanDrag property to true. This can be done in the markup or in the codebehind. The XAML framework handles gesture recognition and fires the DragStarting event to indicate the start of a drag operation. The application must configure the DataPackage by filling its content and indicating which operations are supported. The source application can put different formats in the DataPackage, which will make it compatible with more targets, as shown in **Figure 3**. The supported operations are defined in the DataPackageOperation type and can be Copy, Move, Link or any combination of these.

You can cancel the Drag operation in the event handler by setting the Cancel property of the DragStartingEventArgs parameter; for example, in our sample application, an image placeholder will start a Drag operation only if it has received an image or a file.

The DragStartingEvent handler is also the place where the source application can customize the drag UI, which is explained later in this article.

In some cases, the application might want to use a special gesture to start a drag-and-drop operation, or want to allow dragging a control whose normal interactions interfere with the drag-and-drop gesture, for example, the TextBox, which already reacts to pointer-down events by changing its selection. In these cases, the application can implement its own gesture detection and then initiate a drag-and-drop operation by calling the StartDragAsync method. Note that this method expects a pointer identifier and, therefore, you can't start a drag-and-drop operation with non-standard devices, such as a Kinect sensor. Once StartDragAsync is called, the rest of the drag and drop operation follows the same pattern as if CanDrag=True had been used, including the DragStarting event.

Once the user has released the pointer, the drag-and-drop operation is complete and the source is notified through the DropCompleted event, which contains the DataPackageOperation returned by the target on which the user released the pointer, or DataPackageOperation.None if the pointer was released on a target that doesn't accept the data or if Cancel was pressed:

```
private void DropGrid_DropCompleted(UIElement sender,
DropCompletedEventArgs args)
{
  if (args.DropResult == DataPackageOperation.Move)
  {
    // Move means that we should clear our own image
    Picture = null;
    _bitmapSource = null;
    _fileSource = null;
  }
}
```

StartDragAsync returns an IAsyncOperation<DataPackageOperation>; the source application can handle the end of the operation either by waiting on the IAsyncOperation or by handling the DropCompleted event. Programmatically canceling after the DragStarting event is possible through the IAsyncOperation interface, but might be disturbing for the user.

Note that although both the ListView drag and drop and the UIElement drag and drop are implemented on the same system services and are fully compatible, they don't raise the same events on the source side. That is, if a ListView has its CanDragItems property set to true, only DragItemsStarting and DragItemsCompleted are raised. DragStarting and DropCompleted are events related to the UIElement's CanDrag property.

Implementing a Drop Target Any UIElement can be a drop target provided its AllowDrop property is set to true. During a drag-and-drop operation, the following events can be raised on a target: DragEnter, DragOver, DragLeave and Drop. These events already exist in Windows 8.1 but the DragEventArgs class has been

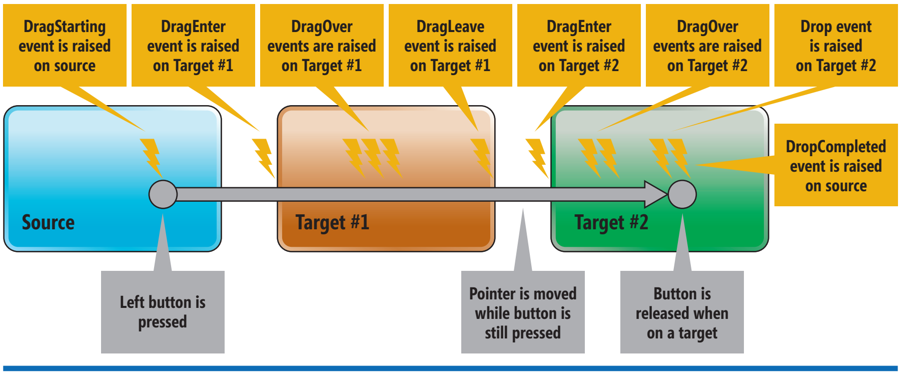

DragStarting event is raised on source
DragEnter event is raised on Target #1
DragOver events are raised on Target #1
DragLeave event is raised on Target #1
DragEnter event is raised on Target #2
DragOver events are raised on Target #2
Drop event is raised on Target #2
DropCompleted event is raised on source
Source
Target #1
Target #2
Left button is pressed
Pointer is moved while button is still pressed
Button is released when on a target

Figure 2 **Drag and Drop Events**

msdn magazine VP

MSDN MAGAZINE VENDOR PROFILE

ASPOSE

Your File Format APIs

File APIs for Developers: DOC, XLS, PPT, PDF, EML and more

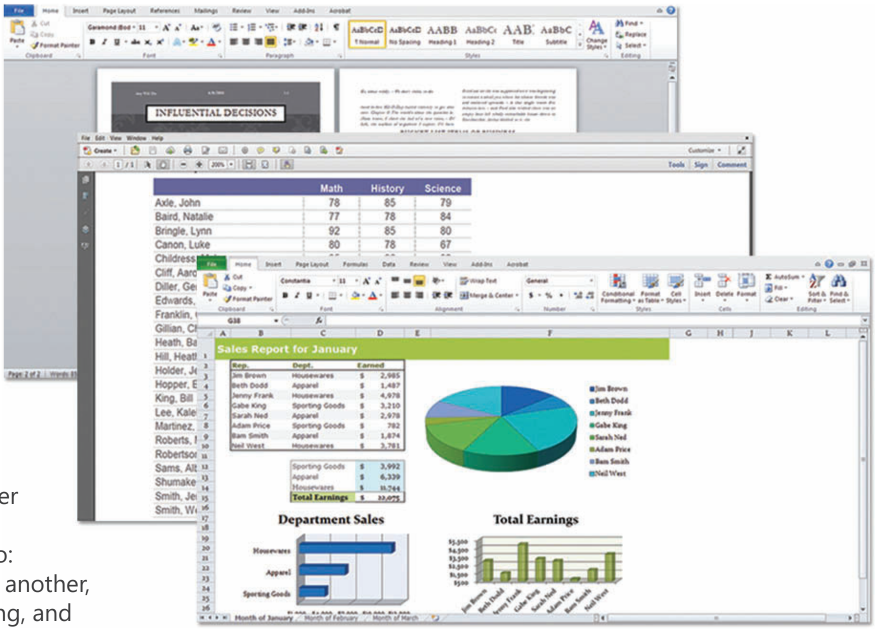
A Q&A with Justin Anderson, Associate Marketing Director at Aspose

Q Who is Aspose, and why should developers care?

A Aspose makes developer APIs for working with business-centric files, like: .DOC, .XLS, .PPT, .PDF, .EML and many more. Visual Studio does not offer native support for working with business files, and no other company offers the comprehensive support that Aspose does. Aspose's core focus is to offer the most complete and powerful set of file management products on the market.

Q What types of things can be done with files?

A Developers use files in a number of ways: importing and exporting data, reporting, user collaboration and data visualization, just to name a few. Aspose APIs allow developers to: create files, convert files from one format to another, combine files, edit file content and formatting, and print files.



INFLUENTIAL DECISIONS

	Math	History	Science
Axle, John	78	85	79
Baird, Natalie	77	78	84
Bringle, Lynn	92	85	80
Canon, Luke	80	78	67

Sales Report for January

Rep.	Dept.	Earned
Jim Brown	Housewares	$ 2,985
Beth Dodd	Apparel	$ 1,487
Jenny Frank	Housewares	$ 4,978
Gabe King	Sporting Goods	$ 3,210
Sarah Ned	Apparel	$ 2,978
Adam Price	Sporting Goods	$ 782
Bam Smith	Apparel	$ 1,874
Neil West	Housewares	$ 3,781

Sporting Goods	$ 3,992
Apparel	$ 6,339
Housewares	$ 11,744
Total Earnings	$ 22,075

Q What makes Aspose special?

A Quality and comprehensiveness; no one else supports files like Aspose. Whether you look at the number of supported file formats, file features or platforms, Aspose is the leader. We offer native APIs for almost every major platform.

Q Does Aspose use Microsoft automation, or have third-party dependencies?

A No. Aspose APIs are 100% standalone. They are a pure .NET alternative to the Microsoft Office Object Model.

Q Who uses Aspose?

A A lot of folks; our customers range from small shops, to the largest enterprise companies. Aspose has over 16,000 customers in 112 countries, and we serve more than 70% of the Fortune 100.

Q Is it hard or costly to get started with Aspose?

A Not at all; Aspose offers one of the best "try before you buy" models around. Free evaluation versions are available for all products. Aspose also provides free support, even before a purchase is made. Free evaluation licenses are available in case developers need to showcase how full versions work.

Aspose believes in the quality of their products, and they want developers to ensure their APIs are a perfect fit, risk free.

Q What about the cloud; does Aspose have a cloud solution?

A Absolutely! Aspose products work just fine in the cloud. Aspose also offers a completely hosted solution, which is pay as you go. Developers can setup a free account and begin testing within minutes.

For more information, please visit → www.aspose.com extended in Windows 10 to give applications access to all the features of modern drag and drop. When handling a drag-and-drop event, the target application should first inspect the content of the DataPackage through the DataView property of the event argument; in most cases, checking for the presence of a data type is enough and it can be done synchronously. In some cases, such as with files, the application might have to check the type of the available files before accepting or ignoring the DataPackage. This is an asynchronous operation and requires the target application to take a deferral and later complete it (this pattern is detailed later in this article).

Once the target has determined if it can process the data, it must set the AcceptedOperation property of the DragEventArgs instance to allow the system to provide the right feedback to the user.

Note that if the application returns DataTransferOperation.None—or an operation not accepted by the source—from an event handler, drop won't take place even if the user releases the pointer over the target; the DragLeave event will be raised instead.

The application can handle either DragEnter or DragOver; the AcceptedOperation returned by DragEnter is kept if DragOver isn't handled. As DragEnter is called only once, it should be preferred to DragOver for performance reasons. However, in the case of nested targets, it's necessary to return the correct value from DragOver in case a parent target might override it (setting Handled to true prevents the event from bubbling up to the parent). In the sample application, each photo placeholder checks for images in the DataPackage and routes the event to the parent grid only if no image is available, which allows the grid to accept Text even if it's physically dropped on a placeholder (see **Figure 4**).

Advanced Concepts

Customizing the visual feedback The only feedback that OLE 2 drag and drop provided was to change the mouse cursor according to the target's answer to the DragOver event. Modern drag and drop allows more advanced scenarios as the visual feedback provided to the user is richer. The drag UI consists of three parts: visual content, glyph and caption.

The visual content represents the data being dragged. It might be the dragged UIElement (if the source is a XAML application); a standard icon chosen by the system based on the DataPackage's content; or a custom image set by the application.

The glyph reflects the type of operation accepted by the target. It can take four different shapes corresponding to the values of the DataPackageOperation type. It can't be customized by applications, but it can be hidden.

Figure 3 **Handling DragStarting and Filling the DataPackage**

```
private void DropGrid_DragStarting(UIElement sender,
  DragStartingEventArgs args)
{
  if (Picture == null)
  {
    args.Cancel = true;
  }
  else if (_fileSource != null)
  {
    args.Data.RequestedOperation =
      DataPackageOperation.Copy | DataPackageOperation.Move;
    args.Data.SetStorageItems(new IStorageItem[] { _fileSource });
  ...
}
```

The caption is a description given by the target. Depending on the target application, a Copy operation might, for example, be the addition of a song to a playlist, the upload of a file to OneDrive or a plain file copy. The caption allows more precise feedback than the glyph and plays a role very similar to a tooltip.

The table in **Figure 5** shows how the source and target can customize these different parts. When the pointer isn't over a drop target, the drag UI is exactly what the source has configured. When the pointer is over a drop target, some parts of the visual might be overridden by the target; all overrides are cleared when the pointer leaves the target.

When a drag-and-drop operation starts, if the source application doesn't try to customize the drag UI in the DragStarting event handler, a snapshot of the dragged UIElement is taken by XAML and used as the drag UI content. The initial UIElement is still displayed in its original position, which is a different behavior from ListView, where the dragged ListViewItems are hidden from their initial position. Because the snapshot of the dragged UIElement is taken after the DragStarting event has been raised, it's possible to trigger a visual state change during this event to alter the snapshot. (Note that the UIElement's state is also altered and, even if it's restored, a light flicker might happen.)

Figure 4 **Handling DragEnter and Inspecting DataPackage**

```
private async void DropGrid_DragEnter(object sender, DragEventArgs e)
{
  if (!App.IsSource(e.DataView))
  {
    bool forceMove = ((e.Modifiers & DragDropModifiers.Shift) ==
      DragDropModifiers.Shift);
    if (e.DataView.Contains(StandardDataFormats.Bitmap))
    {
      _acceptData = true;
      e.AcceptedOperation = (forceMove ? DataPackageOperation.Move :
        DataPackageOperation.Copy);
      e.DragUIOverride.Caption = "Drop the image to show it in this area";
      e.Handled = true;
    }
    else if (e.DataView.Contains(StandardDataFormats.StorageItems))
    {
      // Notify XAML that the end of DropGrid_Enter does
      // not mean that we have finished to handle the event
      var def = e.GetDeferral();
      _acceptData = false;
      e.AcceptedOperation = DataPackageOperation.None;
      var items = await e.DataView.GetStorageItemsAsync();
      foreach (var item in items)
      {
        try
        {
          StorageFile file = item as StorageFile;
          if ((file != null) && file.ContentType.StartsWith("image/"))
          {
            _acceptData = true;
            e.AcceptedOperation = (forceMove ? DataPackageOperation.Move :
              DataPackageOperation.Copy);
            e.DragUIOverride.Caption = "Drop the image to show it in this area";
            break;
          }
        }
        catch (Exception ex)
        {
          Debug.WriteLine(ex.Message);
        }
      }
      e.Handled = true;
      // Notify XAML that now we are done
      def.Complete();
    }
  }
  // Else we let the event bubble on a possible parent target
}
```

Figure 5 **Customizations Available to the Source and Target**

	Source	Target
Visual Content	Default = dragged element Can use system-generated content based on DataPackage Can use any bitmap	Default = what has been set by source Can't use system-generated content Can use any bitmap Can show or hide
Glyph	No access	Aspect based on AcceptedOperation Can show or hide
Caption	No access	Can use any string Can show or hide

When handling a DragStarting event, the drag source can customize the visual feedback through the DragUI property of the DragStartingEventArgs class. For example, asking the system to use the DataPackage's content to generate the visual content is done through SetContentFromDataPackage, as shown in **Figure 6**.

You can set a custom bitmap as the content of the drag UI using two different classes: the well-known XAML BitmapImage class or a new Windows 10 class, SoftwareBitmap. If this bitmap is a resource of the application, it's easier to use a BitmapImage and initialize it with the URI of the resource:

```
private void SampleBorder_DragStarting(UIElement sender,
DragStartingEventArgs args)
{
  args.Data.SetText(SourceTextBox.SelectedText);
  args.DragUI.SetContentFromBitmapImage(new BitmapImage(new Uri(
    "ms-appx:///Assets/cube.png", UriKind.RelativeOrAbsolute)));
}
```

If the bitmap must be generated on the fly when the drag operation starts or when the pointer enters a drop target, a SoftwareBitmap can be created from the buffer generated by the XAML RenderTargetBitmap class, which generates a bitmap containing the visual representation of a UIElement, as shown in **Figure 7**. This UIElement must be in the XAML visual tree but doesn't need to be on the visible part of the page. Because RenderTargetBitmap does this rendering asynchronously, it's necessary to take a deferral here so XAML knows if the bitmap isn't ready when the event handler is finished and waits for the deferral to be completed to update the content of the Drag UI. (We'll explain the deferral mechanism in more detail in the next section of this article.)

Figure 6 **Using SetContentFromDataPackage to Generate Visual Content**

```
private void DropGrid_DragStarting(UIElement sender, DragStartingEventArgs args)
{
  ...
  if (_fileSource != null)
  {
    args.Data.RequestedOperation =
      DataPackageOperation.Copy | DataPackageOperation.Move;
    args.Data.SetStorageItems(new IStorageItem[] { _fileSource });
    args.DragUI.SetContentFromDataPackage();
  }
  else if (_bitmapSource != null)
  {
    args.Data.RequestedOperation =
      DataPackageOperation.Copy | DataPackageOperation.Move;
    args.Data.SetBitmap(_bitmapSource);
    args.DragUI.SetContentFromDataPackage();
  }
}
```

Of course, if the SoftwareBitmap has already been generated—and it could be cached for subsequent drag-and-drop operations—no deferral is needed.

For both SetContentFromBitmapImage and SetContentFromSoftwareBitmap, you can specify an anchor point that indicates how to position the drag UI relative to the pointer position. If you use the overload without anchor point parameter, the top-left corner of the custom bitmap will follow the pointer. The DragStartingEventArgs GetPosition method returns the position of the pointer relative to any UIElement, which can be used to set the start position of the drag UI exactly where the dragged UIElement is located.

On the target side, the different parts of the dragged visual can be customized either in DragEnter or DragOver event handlers. Customization is made through the DragUIOverride property of DragEventArgs class, which exposes four SetContentFrom methods identical to DragUI on the source side, as well as four properties that let you hide different parts of the DragUI and change the caption. Finally, DragUIOverride also exposes a Clear method that resets all overrides of the DragUI made by the target.

Asynchronous Operations The Windows Universal Applications API enforces an asynchronous pattern for all operations that can take more than a few milliseconds. This is particularly critical in the case of drag and drop as such operations are completely driven by the user. Because of its richness, drag and drop uses three different asynchronous patterns: asynchronous calls, deferrals and callbacks.

Asynchronous calls are used when the application calls a system API that might take some time to complete. This pattern is now well-known by Windows developers and is made simple by the async and await keywords in C# (or create_task and then in C++). All methods that retrieve data from DataPackage follow this

Figure 7 **Customizing Drag UI Content with RenderTargetBitmap and SoftwareBitmap**

```
private async void PhotoStripGrid_DragStarting(UIElement sender,
DragStartingEventArgs args)
{
  if ((Picture1.Picture == null) || (Picture2.Picture == null)
    || (Picture3.Picture == null) || (Picture4.Picture == null))
  {
    // Photo Montage is not ready
    args.Cancel = true;
  }
  else

  {
    args.Data.RequestedOperation = DataPackageOperation.Copy;
    args.Data.SetDataProvider(StandardDataFormats.Bitmap,
      ProvideContentAsBitmap);
    App.SetSource(args.Data);
    var deferral = args.GetDeferral();
    var rtb = new RenderTargetBitmap();
    const int width = 200;
    int height = (int)(.5 + PhotoStripGrid.ActualHeight / PhotoStripGrid.ActualWidth
                     * (double)width);
    await rtb.RenderAsync(PhotoStripGrid, width, height);
    var buffer = await rtb.GetPixelsAsync();
    var bitmap = SoftwareBitmap.CreateCopyFromBuffer(buffer, BitmapPixelFormat.Bgra8,
                                                    width, height,
                                                    BitmapAlphaMode.Premultiplied);
    args.DragUI.SetContentFromSoftwareBitmap(bitmap);
    deferral.Complete();
  }
}
```

msdn magazine VP

MSDN MAGAZINE VENDOR PROFILE

Syncfusion's Unique Big Data Solution for Windows

Q&A with Daniel Jebaraj, Vice President of Syncfusion

Syncfusion, Inc. is a leading provider of .NET and Javascript components, leveraging over 12 years of experience with Windows platforms and mobile devices.

Introduction

Syncfusion is a leading enterprise software vendor delivering a broad range of web, mobile, and desktop controls coupled with a service-oriented approach throughout the entire application lifecycle. With over 10,000 customers, including many Fortune 500 companies, Syncfusion creates powerful frameworks that can answer any challenge, from trading systems to managing vast oil fields.

How can Syncfusion help developers with Big Data?

- The Syncfusion Big Data Platform takes the guesswork out of Apache Hadoop, providing development-time support and tools for those working on Windows.
- We provide the missing pieces to integrate big data solutions with .NET, using our tools on top of open source tools to simplify development.
- We provide a complete production environment for deployment to local or cloud clusters.
- Easy-to-use installers are provided to enable one-click installation of a development-time Hadoop stack on any Windows machine. Check it out at syncfusion.com/bigdata15.

What kind of effort is involved in getting started with big data?

- 15 minutes is all it takes to get started. It is as simple as downloading and running a Windows installer.
- There are no additional dependencies to worry about, no virtual machines. Everything is on Windows.
- With our Big Data Platform, you can seriously cut down on startup time and focus on building your big data solution.

What is the cost of the Syncfusion Big Data platform?

- The platform is completely free for any use, including commercial.
- We offer optional paid commercial support for the platform.
- Samples, patches, and workarounds are provided where applicable, all delivered to our SLA.

What is coming next?

- We are particularly excited about our July 2015 release, which will be the first to include support for Apache Spark.
- Apache Spark allows for processing huge volumes of data in a scalable manner without being tied to the Map Reduce model.
- This has major implications when working with problems in several domains, most notably machine learning.
- With Apache Spark, you have a machine learning solution that can truly scale with your needs. This offers companies of all sizes truly astounding possibilities for benefitting from data.

Why choose Syncfusion?

- Syncfusion has over a decade of experience creating solutions for companies big and small.
- We have the expertise needed to make big data and predictive modeling work for customers on the Windows platform.
- No deployment fees whatsoever—save millions over comparable solutions.

If you are not currently a Syncfusion customer, contact Syncfusion to find out more. Call 1-888-9DOTNET or email sales@syncfusion.com today!

To learn more, please visit our website → **www.syncfusion.com/bigdata15**

pattern, such as GetBitmapAsync, which our sample application uses to retrieve an image stream reference, as shown in **Figure 8**.

Deferrals are used when the XAML framework calls back into an application's code that might itself issue an asynchronous call before returning a value awaited by the framework. This pattern was not widely used in previous versions of XAML, so let's take the time to analyze it. When a button generates a Click event, it's a one-way call in the sense that no value needs to be returned by the application. If an asynchronous call is made by the application, its result will be available after the completion of the Click event handler, but this is perfectly fine because this handler doesn't return a value.

On the other hand, when XAML raises a DragEnter or a DragOver event, it expects the application to set the AcceptedOperation property of the event arguments to indicate whether the content of the DataPackage can be handled. If the application only cares about the available data type inside the DataPackage, this can still be done synchronously, for example:

```
private void DropTextBox_DragOver(object sender, DragEventArgs e)
{
  bool hasText = e.DataView.Contains(StandardDataFormats.Text);
  e.AcceptedOperation = hasText ? DataPackageOperation.Copy :
    DataPackageOperation.None;
}
```

Figure 8 **Using Asynchronous Calls to Read the DataPackage**

```
private async void DropGrid_Drop(object sender, DragEventArgs e)
{
  if (!App.IsSource(e.DataView))
  {
    bool forceMove = ((e.Modifiers & DragDropModifiers.Shift) ==
      DragDropModifiers.Shift);

    if (e.DataView.Contains(StandardDataFormats.Bitmap))
    {
      // We need to take a deferral as reading the data is asynchronous
      var def = e.GetDeferral();
      // Get the data
      _bitmapSource = await e.DataView.GetBitmapAsync();
      var imageStream = await _bitmapSource.OpenReadAsync();
      var bitmapImage = new BitmapImage();
      await bitmapImage.SetSourceAsync(imageStream);
      // Display it
      Picture = bitmapImage;
      // Notify the source
      e.AcceptedOperation = forceMove ? DataPackageOperation.Move :
        DataPackageOperation.Copy;
      e.Handled = true;
      def.Complete();

    }
...
}
```

Figure 9 **Deferring the Data Until It Is Actually Read**

```
private void DeferredData_DragStarting(UIElement sender,
  DragStartingEventArgs args)
{
  args.Data.SetDataProvider(StandardDataFormats.Text, ProvideDeferredText);
}

async void ProvideDeferredText(DataProviderRequest request)
{
  var deferral = request.GetDeferral();
  var file = await KnownFolders.DocumentsLibrary.GetFileAsync(fileName);
  var content = await FileIO.ReadTextAsync(file);
  request.SetData(content);
  deferral.Complete();
}
```

However, if, for example, the application can accept only some file types, it must not only check the data types inside the DataPackage, it must access the data, as well, which can only be done asynchronously. This means that the execution of the code is suspended until the data has been read and that the DragEnter (or DragOver) event handler will be completed before the application knows whether it can accept the data. This scenario is exactly the purpose of the deferral: by getting a deferral from the DragEventArg object, the application tells XAML that it will defer some of its processing, and by completing the deferral, the application notifies XAML that this processing is done and the output properties of the DragEventArgs instance have been set. Refer back to **Figure 4** to see how our sample application checks for StorageItems after having got a deferral.

The deferral can also be used when the customization of the drag UI content on the target side requires asynchronous operations such as the RenderTargetBitmap's RenderAsync method.

On the source side of a drag-and-drop operation, DragStartingEventArgs exposes a deferral, too, whose purpose is to allow the operation to start as soon as the event handler terminates (even if the deferral hasn't been completed) in order to provide the fastest feedback to the user even if creating a bitmap to customize the Drag UI takes some time.

Callbacks are used in the DataPackage to defer supplying the data until it's really needed. With this mechanism, the source application can advertise several formats in the DataPackage, but only the data actually read by a target will have to be prepared and transferred. In many cases, the callback will never be called—for example, if no target can understand the corresponding data format—which is a nice performance optimization.

Note that, in many cases, providing the real data will require an asynchronous call and, therefore, the DataProviderRequest parameter of this callback exposes a deferral so applications can notify that they need more time to provide the data and then that the data is available, as shown in **Figure 9**.

Wrapping Up

When writing an application that manipulates standard data formats such as files, images or text, you should consider implementing drag and drop as this is an operation both natural and well-known to users. The basics of drag and drop are already familiar to those who have programmed with Windows Forms and Windows Presentation Foundation, which reduces the learning curve of this rich feature, with its specific concepts such as drag UI customization and the relatively unused patterns such as the deferral pattern. If you just want to support basic drag-and-drop scenarios, you can rely on your previous experience and have a straightforward implementation or, if you prefer, you can fully exploit the new features to provide a tailored experience to the user. ■

Anna Pai *is a software engineer on the Xbox Team. She previously worked on Silverlight, Silverlight for Windows Phone and then XAML for Windows and Windows Phone.*

Alain Zanchetta *is a software engineer on the Windows XAML Team. Previously he was an architect in the consulting division of Microsoft France.*

Thanks to the following Microsoft technical expert for reviewing this article: *Clément Fauchère*

TEXT CONTROL

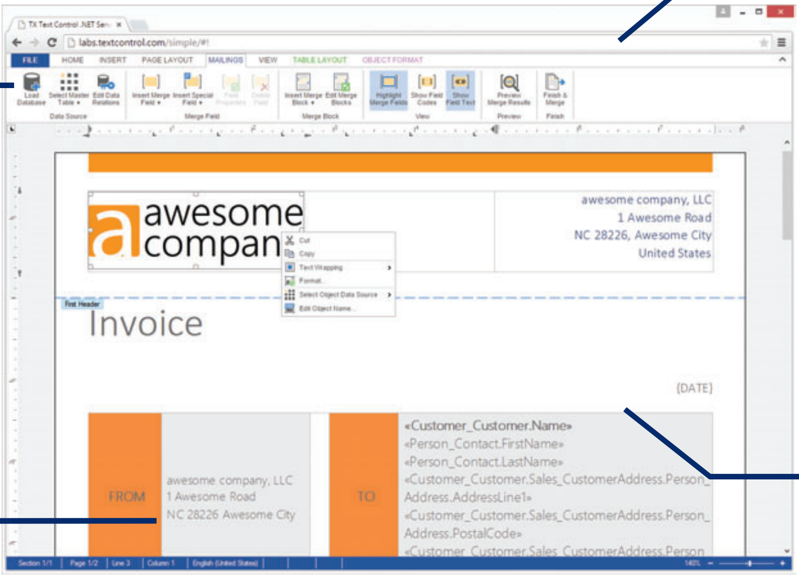
Reporting!

Combine powerful reporting with easy-to-use word processing

The **Text Control Reporting Framework** combines powerful reporting features with an easy-to-use, MS Word compatible word processor. Users create documents and templates using ordinary MS Word skills. TX Text Control is completely independent from MS Word or any other third-party application and can be completely integrated into your business application.

ASP.NET ▪ Windows Forms ▪ WPF

Database support for ADO.NET, ODBC, DataSet, DataTable and all IEnumerable business objects



awesome company

awesome company, LLC
1 Awesome Road
NC 28226, Awesome City
United States

Invoice

[DATE]

FROM
awesome company, LLC
1 Awesome Road
NC 28226 Awesome City

TO
«Customer_Customer.Name»
«Person_Contact.FirstName»
«Person_Contact.LastName»
«Customer_Customer.Sales_CustomerAddress.Person_Address.AddressLine1»
«Customer_Customer.Sales_CustomerAddress.Person_Address.PostalCode»
«Customer_Customer.Sales_CustomerAddress.Person

Cross-browser, cross-platform document and template editing

Create Adobe PDF and PDF/A documents

MS Word compatible templates and MS Word inspired UI

Live demos and 30-day trial version download at:

www.textcontrol.com/reporting

Visual Studio Partner

HTML

TEXT CONTROL

Create a Web Service with Azure Web Apps and WebJobs

Rick Anderson, Kraig Brockschmidt, Tom Dykstra, Erik Reitan and Mike Wasson

Today, many mobile apps are connected to one or more Web services that provide valuable and interesting data. When designing and developing such apps, the easiest approach to take is to make direct REST API calls to those services and then process the response in the client. However, this approach has a number of drawbacks. For one, every network call and every bit of client-side processing consumes precious battery power and bandwidth. Furthermore, extensive client-side processing can take a while to perform, especially on lower-end hardware, making the app less responsive. And different Web services might impose throttling limitations, which means that a purely client-side solution will not readily scale to a larger number of users.

This article discusses:

- Using Azure to power a cloud back end for a mobile app
- Using Azure WebJobs to run continual background tasks
- Automating builds and deployment with Visual Studio Online
- Designing REST APIs for mobile clients with ASP.NET Web API

Technologies discussed:

Microsoft Azure, Entity Framework, ASP.NET, Visual Studio, Visual Studio Online, Windows PowerShell, Xamarin, JavaScript

Code download available at:

aka.ms/altostratusproject

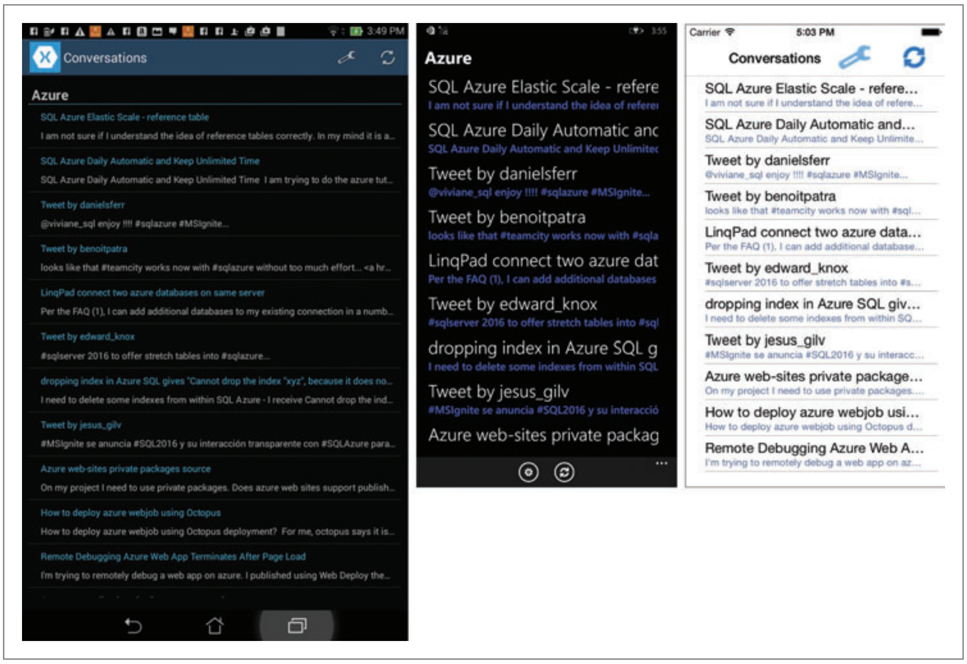
As a result, it makes sense in many scenarios—especially ones that pull data from multiple sources—to create your own back end, to which you can offload certain tasks. As part of our work at Microsoft as a team that develops content for ASP.NET, Microsoft Azure and cross-platform development tools in Visual Studio, we created a specific example of this approach.

In this two-part article, we discuss our approach, some of the challenges we encountered and some of the lessons we learned while developing our application. This application, which we named "Altostratus" (an interesting type of cloud), searches Stack Overflow and Twitter for specific topics that we call "conversations." We chose these two providers because they both have good Web APIs. The application has two main components:

- A cloud back end, hosted on Azure. The back end periodically makes requests to the providers and aggregates the data into the form that's best suited for the client. This avoids throttling concerns, reduces any concerns about latency in the providers, minimizes client-side processing, and reduces the number of network requests from the client. One tradeoff is that the WebJob runs every few hours, so you don't get real-time data.
- A lightweight mobile client app, created with Xamarin to run on Windows, Android and iOS (see **Figure 1**). The mobile client fetches the aggregated data from the back end and presents it to the user. It also keeps a synchronized cache of the data in a local database for a good offline experience and faster startup times.

Users can optionally sign in to the mobile client using social providers (Google or Facebook). When signed in, users can set preferences that further optimize communications with the client. Specifically, they can select which subject areas to receive and the maximum number of conversations within each subject area. User preferences are stored on the back end, so a user can sign in from any client and get the same experience. To demonstrate this idea, we also created a simple Web client that talks to the same back end.

As part of this project, we also wanted to use the application lifecycle management (ALM) tools built into Visual Studio and Visual Studio Online to manage sprints and the work backlog, and to perform automated unit testing, continuous integration (CI) and continuous deployment (CD).



3:49 PM

Conversations

Azure

SQL Azure Elastic Scale - reference table
I am not sure if I understand the idea of reference tables correctly. In my mind it is a...

SQL Azure Daily Automatic and Keep Unlimited Time
SQL Azure Daily Automatic and Keep Unlimited Time I am trying to do the azure tut...

Tweet by danielsferr
@viviane_sql enjoy !!!! #sqlazure #MSIgnite...

Tweet by benoitpatra
looks like that #teamcity works now with #sqlazure without too much effort... <a hr...

LinqPad connect two azure databases on same server
Per the FAQ (1), I can add additional databases to my existing connection in a numb...

Tweet by edward_knox
#sqlserver 2016 to offer stretch tables into #sqlazure...

dropping index in Azure SQL gives "Cannot drop the index "xyz", because it does no...
I need to delete some indexes from within SQL Azure - I receive Cannot drop the ind...

Tweet by jesus_gilv
#MSIgnite se anuncia #SQL2016 y su interacción transparente con #SQLAzure para...

Azure web-sites private packages source
On my project I need to use private packages. Does azure web sites support publish...

How to deploy azure webjob using Octopus
How to deploy azure webjob using Octopus deployment? For me, octopus says it is...

Remote Debugging Azure Web App Terminates After Page Load
I'm trying to remotely debug a web app on azure. I published using Web Deploy the...

3:55

Azure

SQL Azure Elastic Scale - refere
I am not sure if I understand the idea of referei

SQL Azure Daily Automatic anc
SQL Azure Daily Automatic and Keep Unlimitec

Tweet by danielsferr
@viviane_sql enjoy !!!! #sqlazure #MSIgnite...

Tweet by benoitpatra
looks like that #teamcity works now with #sqla

LinqPad connect two azure dat
Per the FAQ (1), I can add additional databases

Tweet by edward_knox
#sqlserver 2016 to offer stretch tables into #sql

dropping index in Azure SQL g
I need to delete some indexes from within SQL

Tweet by jesus_gilv
#MSIgnite se anuncia #SQL2016 y su interacció

Azure web-sites private packag

Carrier 5:03 PM

Conversations

SQL Azure Elastic Scale - refere...
I am not sure if I understand the idea of refere...

SQL Azure Daily Automatic and...
SQL Azure Daily Automatic and Keep Unlimite...

Tweet by danielsferr
@viviane_sql enjoy !!!! #sqlazure #MSIgnite...

Tweet by benoitpatra
looks like that #teamcity works now with #sql...

LinqPad connect two azure data...
Per the FAQ (1), I can add additional database...

Tweet by edward_knox
#sqlserver 2016 to offer stretch tables into #s...

dropping index in Azure SQL giv...
I need to delete some indexes from within SQ...

Tweet by jesus_gilv
#MSIgnite se anuncia #SQL2016 y su interacc...

Azure web-sites private package...
On my project I need to use private packages....

How to deploy azure webjob usi...
How to deploy azure webjob using Octopus d...

Remote Debugging Azure Web A...
I'm trying to remotely debug a web app on az...

Figure 1 **The Xamarin Mobile Client Running on an Android Tablet (Left), a Windows Phone (Middle) and an iPhone (Right)**

This two-part article explores the details of our project. Part 1 focuses on the back end and our use of ALM tools (DevOps). We'll also talk about some of the challenges we encountered and some of the lessons learned, such as:

- How to securely automate the deployment of passwords to non-Web apps.
- How to handle Azure automation time-outs.
- Efficient and informative background processing.
- CI/CD limitations and workarounds.

Part 2 will discuss how we used Xamarin to target multiple mobile client platforms, including authentication and maintaining a synchronized client-side cache of the data.

Architecture

Figure 2 shows the high-level architecture for the Altostratus solution.

- On the back end, we use Azure App Service to host the Web app, and Azure SQL Database to store data in a relational database. We use Entity Framework (EF) for data access.
- We use Azure WebJobs to run a scheduled background task that aggregates data from Stack Overflow and Twitter and writes it to the database. The WebJob can easily be extended to aggregate data from additional sources.
- The mobile client is created using Xamarin and communicates with the back end using a simple REST API.
- The REST API is implemented using ASP.NET Web API, which is a framework for creating HTTP services in the Microsoft .NET Framework.
- The Web client is a relatively simple JavaScript app. We used the KnockoutJS library for data binding and jQuery for AJAX calls.

The Database Schema

We use EF Code First to define the database schema and manage the back-end SQL database. As **Figure 3** shows, the database stores the following entities:

- Provider: A data source, such as Twitter or Stack Overflow.
- Conversation: An item from a provider. For Stack Overflow, this corresponds to a question with answers. For Twitter, it corresponds to a tweet. (It could also be a tweet with replies, but we didn't implement that feature.)
- Category: The subject for a conversation, such as "Azure" or "ASP.NET."
- Tag: A search string for a specific category and provider. These correspond to tags in Stack Overflow ("azure-web-sites") and hash tags in Twitter ("#azurewebsites"). The back end uses these to query the providers. The end user doesn't see them.
- UserPreference: Stores per-user preferences.
- UserCategory: Defines a join table for UserPreference and Category.

In general, the Create, Read, Update, Delete (CRUD) code in the Altostratus application is typical for EF Code First. One special consideration has to do with handling database seeding and migrations. Code First creates and seeds a new database if no database exists the first time a program tries to access data. Because the first attempt to access data could happen in the WebJob, we have code in the WebJob's Main method to make EF use the MigrateDatabaseToLatestVersion initializer:

```
static void Main()
{
  Task task;
  try
  {
    Database.SetInitializer<ApplicationDbContext>(
      new MigrateDatabaseToLatestVersion<ApplicationDbContext,
        Altostratus.DAL.Migrations.Configuration>());
```

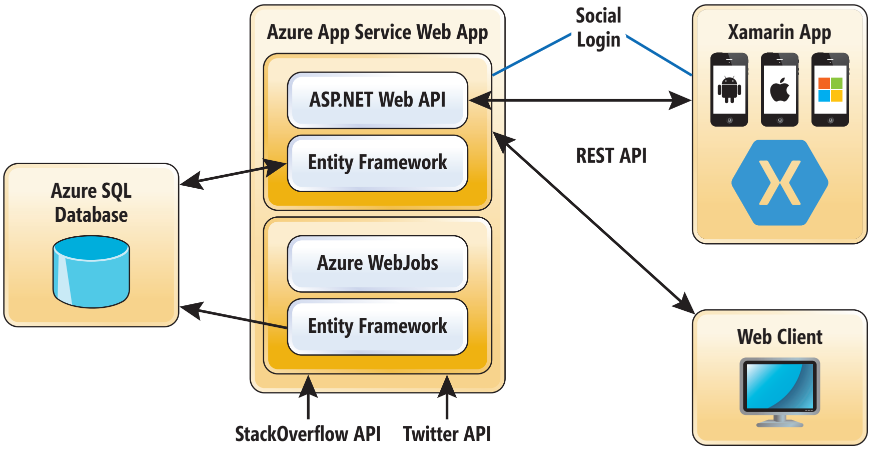


Figure 2 **Altostratus Architecture**

Without this code, the WebJob would use the CreateDatabaseIfNotExists initializer by default, which doesn't seed the database. The result would be an empty database and errors when the application tries to read data from empty tables.

Designing the WebJob

WebJobs provides an ideal solution for running background tasks, which is work that previously would have required a dedicated Azure Worker Role to perform. You can run WebJobs on an Azure Web app with no additional cost. You can read about the advantages WebJobs provide over worker roles at Troy Hunts blog post, "Azure WebJobs Are Awesome and You Should Start Using Them Right Now!" (bit.ly/1c28yAk).

The WebJob for our solution periodically runs three functions: get Twitter data, get Stack Overflow data and purge old data. These functions are independent, but they must be run sequentially because they share the same EF context. After a WebJob completes, the Azure Portal shows the status of each function (see **Figure 3**). If the function completes, it's marked with a green Success message, and if an exception is thrown, it's marked with a red Failed message.

Failures are not infrequent in Altostratus because we use the free Stack Overflow and Twitter provider APIs. Queries, in particular, are limited: If you exceed the query limit, the providers return a throttling error. This was a key reason for creating the back end in the first place. Although it would be straightforward for a mobile app to make requests to these providers directly, a growing number of users could quickly reach the throttling limit. Instead, the back end can make just a few periodic requests to collect and aggregate the data.

One issue we encountered was around WebJob error handling. Normally, if any function in the WebJob throws an exception, the WebJob instance is terminated, the remaining functions don't run and the entire WebJob run is marked as failed. In order to run all the tasks and show a failure at the function level, the WebJob must catch exceptions. If any function in Main throws, we log the last exception and re-throw, so the WebJob gets marked as Failed. The pseudo code in **Figure 4** shows this approach. (The project download contains the complete code.)

In **Figure 4**, only the last exception is shown at the WebJob level. At the function level, every exception is logged, so no exceptions are lost. **Figure 5** shows the dashboard for a WebJob that ran successfully. You can drill down into each function to see diagnostic output.

Designing the REST API

The mobile app communicates to the back end through a simple REST API, which we implemented using ASP.NET Web API 2. (Note that in ASP.NET 5, Web API is merged into the MVC 6 framework, making it simpler to incorporate both into a Web app.)

Figure 6 summarizes our REST API.

All responses are in JSON format. For example, **Figure 7** shows an HTTP response for conversations.

For conversations, we don't require the request to be authenticated. That means the mobile app can always show meaningful data without requiring the user to log in first. But we also wanted to demonstrate having the back end perform additional optimizations when a user logs in. Our client app lets the user select which categories of conversations to display, along with a conversation limit. So if a request is authenticated (meaning the user logged into the app), the back end automatically filters the response based on those preferences. This limits the amount of data that must be processed by the client, which, over time, reduces demands on bandwidth, memory, storage and battery power.

The conversations API also takes an optional "from" parameter in the query string. If specified, this filters the results to include only conversations updated after that date:

```
GET api/conversations?from=2015-04-20T03:59Z
```

This minimizes the size of the response. Our mobile client uses this parameter to ask for only the data it needs to synchronize its cache.

We could have designed the API to use query strings to communicate a user's preferences on a per-request basis, meaning those preferences would be maintained only in the client. That would have avoided the need for authentication. While this approach would work fine for basic scenarios, we wanted to provide an example that could be extended to more complicated situations where query strings would be insufficient. Storing preferences on the back end also means they're automatically applied on every client where the same user logs in.

Data Transfer Objects (DTOs)

The REST API is the boundary between the database schema and the wire representation. We didn't want to serialize the EF models directly:

- They contain information the client doesn't need, like foreign keys.
- They can make the API vulnerable to over-posting. (Over-posting is when a client updates database fields you didn't indent to expose for updates. It can occur when you convert an HTTP request payload directly into an EF model without validating the input sufficiently. See bit.ly/1It1wl2 for more information.)

NEW YORK
Code Trip
NAVIGATE THE .NET HIGHWAY

- The “shape” of the EF models are designed for creating database tables, and aren’t optimal for the client.

Therefore, we created a set of data transfer objects (DTOs), which are just C# classes that define the format for the REST API responses. For example, here’s our EF model for categories:

```
public class Category
{
  public int CategoryID { get; set; }
  [StringLength(100)]
  public string Name { get; set; }  // e.g: Azure, ASP.NET

  public ICollection<Tag> Tags { get; set; }
  public ICollection<Conversation> Conversations { get; set; }
}
```

The category entity has a primary key (CategoryID) and navigation properties (Tags and Conversations). The navigation properties make it easy to follow relations in EF—for example, to find all the tags for a category.

When the client asks for the categories, it just needs a list of category names:

```
[ "Azure", "ASP.NET" ]
```

This conversion is easy to perform using a LINQ Select statement in the Web API controller method:

```
public IEnumerable<string> GetCategories()
{
  return db.Categories.Select(x => x.Name);
}
```

The UserPreference entity is a bit more complicated:

```
public class UserPreference
{
  // FK to AspNetUser table Id
  [Key, DatabaseGenerated(DatabaseGeneratedOption.None)]
  public string ApplicationUser_Id { get; set; }
  public int ConversationLimit { get; set; }
  public int SortOrder { get; set; }
  public ICollection<UserCategory> UserCategory { get; set; }

  [ForeignKey("ApplicationUser_Id")]
  public ApplicationUser AppUser { get; set; }
}
```

ApplicationUser_Id is a foreign key to the user table. UserCategory points to a junction table, which creates a many-to-many relation between user preferences and categories.

Here’s how we want this to look to the client:

```
public class UserPreferenceDTO
{
  public int ConversationLimit { get; set; }
  public int SortOrder { get; set; }
  public ICollection<string> Categories { get; set; }
}
```

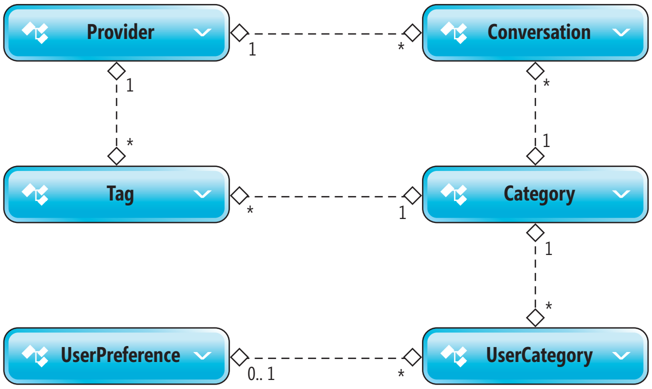


Figure 3 **The Altostratus Data Model**

This hides the things that are implementation details of the database schema, like foreign keys and junction tables, and flattens category names into a list of strings.

The LINQ expression to convert UserPreference to UserPreferenceDTO is fairly complex, so we used AutoMapper instead. AutoMapper is a library that maps object types. The idea is to define a mapping once, and then use AutoMapper to do the mapping for you.

We configure AutoMapper when the app starts:

```
Mapper.CreateMap<Conversation, ConversationDTO>();

Mapper.CreateMap<UserPreference, UserPreferenceDTO>()
  .ForMember(dest => dest.Categories,
             opts => opts.MapFrom(
               src => src.UserCategory.Select(
                 x => x.Category.Name).ToList()));
              // This clause maps ICollection<UserCategory> to a flat
              // list of category names.
```

The first call to CreateMap maps Conversation to ConversationDTO, using AutoMapper’s default mapping conventions. For UserPreference, the mapping is less straightforward, so there’s some extra configuration.

Once AutoMapper is configured, mapping objects is easy:

```
var prefs = await db.UserPreferences
  .Include(x => x.UserCategory.Select(y => y.Category))
  .SingleOrDefaultAsync(x => x.ApplicationUser_Id == userId);

var results = AutoMapper.Mapper.Map<UserPreference,
  UserPreferenceDTO>(prefs);
```

AutoMapper is smart enough to convert this into a LINQ statement, so the SELECTs happen on the database.

Authentication and Authorization

We use a social login (Google and Facebook) to authenticate users. (The authentication process will be described in detail in Part 2

Figure 4 **Catching and Re-Throwing Exceptions**

```
static void Main()
{
  Task task;
  try
  {
    Exception _lastException = null;
    try
    {
      task = host.CallAsync("Twitter");
      task.Wait();
    }
    catch (Exception ex)
    {
      _lastException = ex;
    }
    try
    {
      task = host.CallAsync("StackOverflow");
      task.Wait();
    }
    catch (Exception ex)
    {
      _lastException = ex;
    }
    task = host.CallAsync("Purge Old Data");
    task.Wait();
    if (_lastException != null)
    {
      throw _lastException;
    }
  }
  catch (Exception ex)
  {

  }
}
```

of this article.) After Web API authenticates the request, the Web API controller can use this information to authorize requests and to look up the user in the user database.

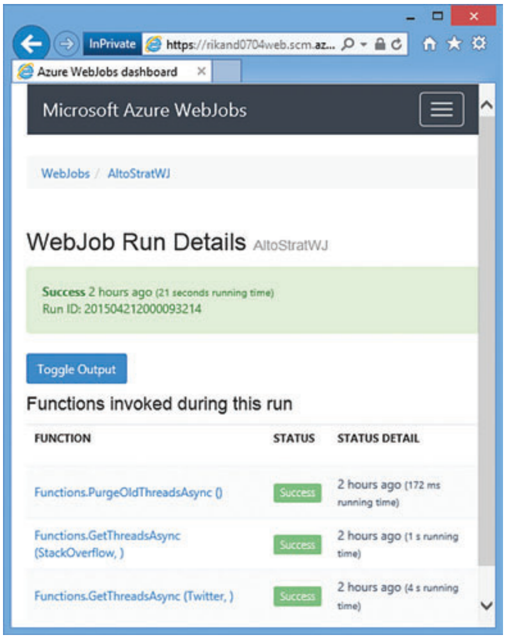
To restrict a REST API to authorized users, we decorate the controller class with the [Authorize] attribute:

```
[Authorize]
public class UserPreferencesController :
ApiController
```

Now if a request to api/userpreferences isn't authorized, Web API automatically returns a 401 error:

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json; charset=utf-8
WWW-Authenticate: Bearer
Date: Tue, 21 Apr 2015 23:55:47 GMT
Content-Length: 68

{
  "Message": "Authorization has been
denied for this request."
}
```

Notice that Web API added a WWW-Authenticate header to the response. This tells the client what type of authentication scheme is supported—in this case, OAuth2 bearer tokens.



WebJob Run Details AltoStratWJ

Success 2 hours ago (21 seconds running time)
Run ID: 201504212000093214

Functions invoked during this run

FUNCTION	STATUS	STATUS DETAIL
Functions.PurgeOldThreadsAsync ()	Success	2 hours ago (172 ms running time)
Functions.GetThreadsAsync (StackOverflow,)	Success	2 hours ago (1 s running time)
Functions.GetThreadsAsync (Twitter,)	Success	2 hours ago (4 s running time)

Figure 5 **Successful WebJob Run**

By default, [Authorize] assumes that every authenticated user is also authorized, and only anonymous requests are blocked. You can also limit authorization to users in specified roles (such as Admins), or implement a custom authorization filter for more complex authorization scenarios.

The conversations API is a more interesting case: We allow anonymous requests, but apply extra logic when the request is authenticated. The following code checks whether the current request is authenticated and, if so, gets the user preferences from the database:

```
if (User.Identity.IsAuthenticated)
{
  string userId = User.Identity.GetUserId();

  prefs = await db.UserPreferences
    .Include(x => x.UserCategory.Select(y => y.Category))
    .Where(x => x.ApplicationUser_Id == userId).SingleOrDefaultAsync();
}
```

If prefs is non-null, we use the preferences to shape the EF query. For anonymous requests, we just run a default query.

Data Feeds

As mentioned earlier, our app pulls data from Stack Overflow and Twitter. One of our design principles was to take data from multiple, diverse sources and aggregate it into a single, normalized source. This simplifies the interaction between clients and the back end, because clients don't need to know the data format for any particular provider. In the back end, we implemented a provider model that makes it easy to aggregate additional sources, without requiring any changes in the Web APIs or clients. Providers expose a consistent interface:

```
interface IProviderAPI
{
   Task<IEnumerable<Conversation>> GetConversationsAsync(
     Provider provider, Category category,
     IEnumerable<Tag> tags, DateTime from, int maxResults, TextWriter logger);
}
```

The Stack Overflow and Twitter APIs provide plenty of capability, but with that capability comes complexity. We found that the StacMan and LINQtoTwitter NuGet packages made working with the APIs much easier. LINQtoTwitter and StacMan are well-documented, actively supported, open source and are easy to use with C#.

Handling Passwords and Other Secrets

We followed the article, "Best Practices for Deploying Passwords and Other Sensitive Data to ASP.NET and Azure App Service" (bit.ly/1zlNiQI), which mandates never checking passwords in source code. We store secrets only in auxiliary config files on local development machines. To deploy the app to Azure, we use Windows PowerShell or the Azure Portal.

This approach works well for the Web app. We moved the secrets out of the web.config file with the following markup:

```
<appSettings file="..\..\AppSettingsSecrets.config">
</appSettings>
```

Moving the file up two levels from the source directory means it's completely out of the solution directory and won't get added to source control.

The app.config file used by a console app (WebJobs) doesn't support relative paths, but it does support absolute paths. You can use an absolute path to move your secrets out of your project

Figure 6 **REST API Summary**

GET api/categories	Gets the categories.
GET api/ conversations?from=iso-8601-date	Gets the conversations.
GET api/userpreferences	Gets the user's preferences.
PUT api/userpreferences	Updates the user's preferences.

Figure 7 **An HTTP Response for Conversations**

```
HTTP/1.1 200 OK
Content-Length: 93449
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/8.0
Date: Tue, 21 Apr 2015 22:38:47 GMT

[
  {
    "Url": "http://twitter.com/815911142/status/590317675412262912",
    "LastUpdated": "2015-04-21T00:54:36",
    "Title": "Tweet by rickraineytx",
    "Body": "Everything you need to know about #AzureWebJobs is here.
      <a href=\"http://t.co/t2bywUQoft\"">http://t.co/t2bywUQoft</a>",
    "ProviderName": "Twitter",
    "CategoryName": "Azure"
  },

  // ... Some results deleted for space

]
```

Boost ASP.NET Performance
Read our Whitepaper online

directory. The following markup adds the secrets in the C:\secrets\AppSettingsSecrets.config file, and non-sensitive data in the app.config file:

```
<configuration>
  <appSettings file="C:\secrets\AppSettingsSecrets.config">
    <add key="TwitterMaxThreads" value="24" />
    <add key="StackOverflowMaxThreads" value="24" />
    <add key="MaxDaysForPurge" value="30" />
  </appSettings>
</configuration>
```

To handle the secrets in our Windows PowerShell scripts we use the Export-CliXml cmdlet to export the encrypted secrets to disk and Import-CliXml to read the secrets.

Automate Everything

A DevOps best practice is to automate everything. We originally wrote Windows PowerShell scripts to create the Azure resources our app requires (Web app, database, storage) and hook up all the resources, such as setting the connection string and the app settings secrets.

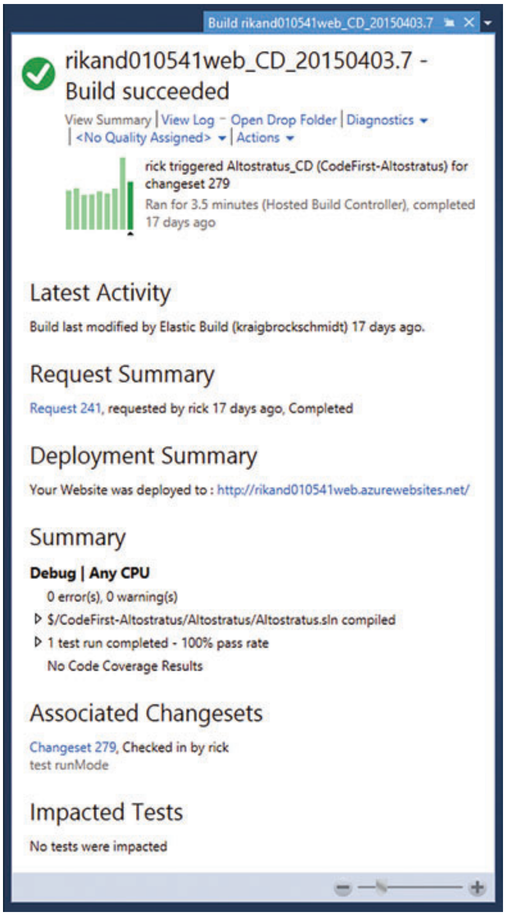
The Visual Studio Deployment Wizard does a good job of automating deployment to Azure, but there are still several manual steps to deploy and configure the app:

- Entering the password of the administrator account on the Azure SQL Database.
- Entering the app settings secrets for the Web app and WebJob.
- Entering the WebJob storage account strings to hook up WebJob monitoring.
- Updating the deployment URL in the Facebook and Google developer consoles, to enable OAuth social logins.

A new deployment URL requires you to update your OAuth provider authentication URL, so there's no way to automate the last step without using a custom domain name. Our Windows PowerShell scripts create all the Azure resources we need and hook them up, so everything that can be automated is automated.

The first time you deploy a WebJob from Visual Studio, you're prompted to set up a schedule for running the WebJob. (We run the WebJob every three hours.) At the time of this writing, there's no way in Windows PowerShell to set a WebJob schedule. After running the Windows PowerShell creation and deployment scripts, you need the one-time additional step of deploying the WebJob from Visual Studio to set up the schedule, or you can set up a schedule on the portal.

We soon discovered that the Windows PowerShell scripts would sometimes time out while attempting to create a resource. Using the traditional Azure PowerShell approach, there's no simple way to deal with a random resource-creation failure. Deleting any resources that were successfully created requires a non-trivial script that could possibly delete resources you didn't intend to remove. Even when your script is successful, after you finish testing, you need to delete all the resources the script created. Keeping track of the resources to delete after a test run is non-trivial and error-prone.



Build rikand010541web_CD_20150403.7

rikand010541web_CD_20150403.7 - Build succeeded

View Summary | View Log - Open Drop Folder | Diagnostics
| <No Quality Assigned> | Actions

rick triggered Altostratus_CD (CodeFirst-Altostratus) for changeset 279

Ran for 3.5 minutes (Hosted Build Controller), completed 17 days ago

Latest Activity

Build last modified by Elastic Build (kraigbrockschmidt) 17 days ago.

Request Summary

Request 241, requested by rick 17 days ago, Completed

Deployment Summary

Your Website was deployed to : http://rikand010541web.azurewebsites.net/

Summary

Debug | Any CPU

0 error(s), 0 warning(s)

▷ $/CodeFirst-Altostratus/Altostratus/Altostratus.sln compiled

▷ 1 test run completed - 100% pass rate

No Code Coverage Results

Associated Changesets

Changeset 279, Checked in by rick
test runMode

Impacted Tests

No tests were impacted

Figure 8 **A Build Summary**

Note that resource-creation timeout is not a flaw with Azure. Remotely creating complex resources, such as a data server or Web app, is inherently time-consuming. Cloud apps must be architected from the beginning to deal with timeouts and failure.

Azure Resource Manager (ARM) to the Rescue

ARM allows you to create resources as a group. This lets you easily create all your resources and handle transient faults, so if your script fails to create a resource, you can try again. Additionally, cleanup is easy. You simply delete your resource group, and all the dependent objects are automatically deleted.

Transient faults occur infrequently, and typically only one retry is necessary for the operation to succeed. The following snippet from a Windows PowerShell script shows a simple approach to implement retry logic with linear backoff when using an ARM template:

```
$cnt = 0
$SleepSeconds = 30
$ProvisioningState = 'Failed'
while ( [string]::Compare($ProvisioningState, 'Failed', $True) -eq 0
-and ($cnt -lt 4 ) ){
   My-New-AzureResourceGroup -RGname $RGname `
    -WebSiteName $WebSiteName -HostingPlanName $HostingPlanName
  $RGD = Get-AzureResourceGroupDeployment -ResourceGroupName $RGname
  $ProvisioningState = $RGD.ProvisioningState
  Start-Sleep -s ($SleepSeconds * $cnt)
  $cnt++
}
```

My-New-AzureResourceGroup is a Windows PowerShell function we wrote that wraps a call to the cmdlet New-AzureResourceGroup, which uses an ARM template to create Azure resources. The New-AzureResourceGroup call will almost always succeed, but the creation of resources specified by the template can time out.

If any resource wasn't created, the provisioning state is Failed, and the script will sleep and try again. During a retry, resources that were already successfully created aren't recreated. The preceding script attempts three retries. (Four failures in a row almost certainly indicate a non-transient error.)

The idempotence that ARM provides is extremely useful in scripts that create many resources. We're not suggesting you need this retry logic in all your deployments, but ARM gives you that option when it's beneficial. See "Using Azure PowerShell with Azure Resource Manager" (bit.ly/1GyaMzv).

msdn
magazine

MSDN MAGAZINE VENDOR PROFILE

dtSearch®

The Smart Choice for
Text Retrieval® since 1991

Instantly Search Terabytes of Text

incorporating dtSearch's
own document filters

Executive Summary

The dtSearch product line includes enterprise and developer products for instantly searching terabytes of text with over 25 search options.

dtSearch's own document filters parse a wide variety of data formats, including "Office" documents, PDFs, compression formats, emails and attachments, online data and other databases. After a search, the document filter can display retrieved data with highlighted hits.

Developer SDKs span multiple platforms, with APIs for .NET, Java and C++. See dtSearch.com for "how to" articles on everything from faceted search to Microsoft Azure.

Key Benefits

Terabyte Indexer. Each dtSearch index can hold over a terabyte of data, including any number of online and offline data sources. dtSearch further supports building an unlimited number of terabyte-size indexes, and simultaneously searching all indexes.

Concurrent Searching. dtSearch products support efficient multithreaded searching with no limit on the number of search threads. Indexed search time is typically less than a second.

Federated Searching and the dtSearch Spider. Federated searching can span any number of directories, emails and attachments, databases, and through the dtSearch Spider, local and remote as well as public and secure online content. dtSearch products support integrated relevancy ranking with highlighted hits across all supported data repositories.

25+ Search Options and International Language Support. dtSearch products offer more than 25 different search types, including advanced forensics search options. International language support covers both right-to-left languages, and special Chinese/Japanese/Korean character handling.

Faceted Search and Other Data Classification Options. Developer APIs support faceted search and multiple other full-text and metadata classification options.

Document Filters. The dtSearch document filters can parse, extract, search, and display in a browser with highlighted hits a broad range of online and offline formats (see above). The document filters are embedded across the dtSearch product line, and are also available for separate OEM licensing.

SDKs for Multiple Platforms. dtSearch text retrieval and document filter APIs are available in .NET, C++ and Java. SDKs now cover multiple platforms. See site for topics such as SQL data, faceted search, and Microsoft Azure.

For hundreds of developer case studies and press reviews, and fully-functional evaluations, visit → **dtSearch.com**

Build Integration and Automated Deployment

Visual Studio Online made it easy to set up continuous integration (CI). Whenever code is checked into Team Foundation Server (TFS), it automatically triggers a build and runs the unit tests. We also employed continuous delivery (CD). If the build and automated unit tests are successful, the app is automatically deployed to our Azure test site. You can read about setting up CI/CD to Azure at the documentation page, "Continuous Delivery to Azure Using Visual Studio Online" (bit.ly/1OkMkaW).

Early in the dev cycle, when there were three or more of us actively checking in source code, we used the default build definition trigger to kick off the build/test/deploy cycle at 3 a.m., Monday through Friday. This worked well, giving us all a chance to do a quick test on the Web site when we started working each morning. As the code base stabilized and check-ins were less frequent but perhaps more critical, we set the trigger to CI mode, so each check-in would trigger the process. When we got to the "Code clean up" phase, with frequent low-risk changes, we set the trigger to Rolling builds, where the build/test/deployment cycle is triggered at most every hour. **Figure 8** shows a build summary that includes deployment and test coverage.

Wrapping Up

In this article, we looked at some of the considerations when creating a cloud back end that aggregates and processes data and serves it to mobile clients. No individual piece of our sample app is terribly complicated, but there are a lot of moving parts, which is typical of cloud-based solutions. We also looked at how Visual Studio Online made it possible for a small team to run continuous builds and continuous deployment without a dedicated DevOps manager.

In Part 2, we'll look in detail at the client app and how Xamarin Forms made it easy to target multiple platforms with a minimum of platform-specific code. We'll also delve into the mysteries of OAuth2 for social login. ■

Rick Anderson *works as a senior programming writer for Microsoft, focusing on ASP.NET MVC, Microsoft Azure and Entity Framework. You can follow him on Twitter at twitter.com/RickAndMSFT.*

Kraig Brockschmidt *works as a senior content developer for Microsoft and is focused on cross-platform mobile apps. He's the author of "Programming Windows Store Apps with HTML, CSS and JavaScript" (two editions) from Microsoft Press and blogs on kraigbrockschmidt.com.*

Tom Dykstra *is a senior content developer at Microsoft, focusing on Microsoft Azure and ASP.NET.*

Erik Reitan *is a senior content developer at Microsoft. He focuses on Microsoft Azure and ASP.NET. Follow him on Twitter at twitter.com/ReitanErik.*

Mike Wasson *is a content developer at Microsoft. For many years he documented the Win32 multimedia APIs. He currently writes about Microsoft Azure and ASP.NET.*

Thanks to the following technical experts for reviewing this article:
*Michael Collier, John de Havilland, Brady Gaster, Ryan Jones,
Vijay Ramakrishnan and Pranav Rastogi*

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

vslive.com/redmond

Redmond AUGUST 10 - 14
MICROSOFT HEADQUARTERS, REDMOND, WA

REDMOND
Code Trip
NAVIGATE THE .NET HIGHWAY

CODE HOME

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.

EVENT SPONSOR Microsoft

SUPPORTED BY Visual Studio msdn magazine Visual Studio MAGAZINE

PRODUCED BY 1105 MEDIA INC

SEATTLE
Bay
West
Union
Dep

LAS VEGAS
Code Trip
March 16-20
AUSTIN
Code Trip
June 1-4
SAN FRANCISCO
Code Trip
June 15-18
REDMOND
Code Trip
August 10-14
NEW YORK
Code Trip
Sept. 28- Oct. 1
ORLANDO
Code Trip
Nov. 16-20

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.

Visual Studio MAGAZINE

PRODUCED BY
1105 MEDIA

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 Magenic, 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.

LIVE360EVENTS.COM

Azure Mobile Services: A Great Back End for AngularJS

Jonathan C. Miller

AngularJS is a great framework for building JavaScript applications for both Web and mobile. It's very powerful, but it does come with a bit of a learning curve. To get started, I read blogs and books and watched video courses, which are great for learning client-side features, such as forms, routing and validation. Unfortunately, the client-side topics always seem to overshadow back-end concerns. Most learning resources barely touch on these at all. One course used the Angular $httpBackend service almost exclusively. $httpBackend is a great way to build fakes for testing, but it isn't meant for persisting data for a production application. Another resource used an open source product called deployd (deployd.com), which provides a quick and easy way to get a REST/API back-end server up and running. Deployd is free to download and can run on a development machine or server. It's great for modeling and testing REST APIs. The problem once again is what to do in production. In a production environment, I have to expose a REST/JSON server to the Internet for the AngularJS application to use, but I don't want to be in the business of hosting and managing servers on the Internet. I need to be able to spin up new applications fast and scale up quickly as needed. I need integrated security without a lot of complexity. I need to be able to set up REST/JSON APIs to store the application data. And, I need all of these issues to be easy to learn and integrate with my application. Luckily, in my research I found that Microsoft has already solved these problems. In this article I'll show you how to integrate the Azure Mobile Services back end with an AngularJS front end.

Azure Mobile Services (AMS) is really a back end in a box. It brings together all the back-end parts needed for a production app and has a number of compelling features:

- It provides very fast and redundant cloud storage.
- It makes building tables that can be accessed via REST/JSON very easy.
- It provides built-in security and authentication to popular login providers such as Microsoft, Google, Facebook and Twitter.
- It starts out free and can scale out for high-demand applications.
- It makes server-side validation easy.
- It allows for both JavaScript and .NET back ends. Microsoft has made it very easy to provision an AMS site and integrate it with almost any client platform.

This article discusses:

- Benefits of Azure Mobile Services
- Creating a simple Angular Notes app
- Storing data in Azure Mobile Services
- Authenticating users in Azure Mobile Services

Technologies discussed:

Azure Mobile Services, AngularJS, JavaScript

Code download available at:

msdn.microsoft.com/magazine/msdnmag0815

We know how application development can be.

INTRODUCING

Atalasoft MobileImage®

MOBILE CAPTURE SDK

Try it free for 30 days with full support

ATALASOFT.COM

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

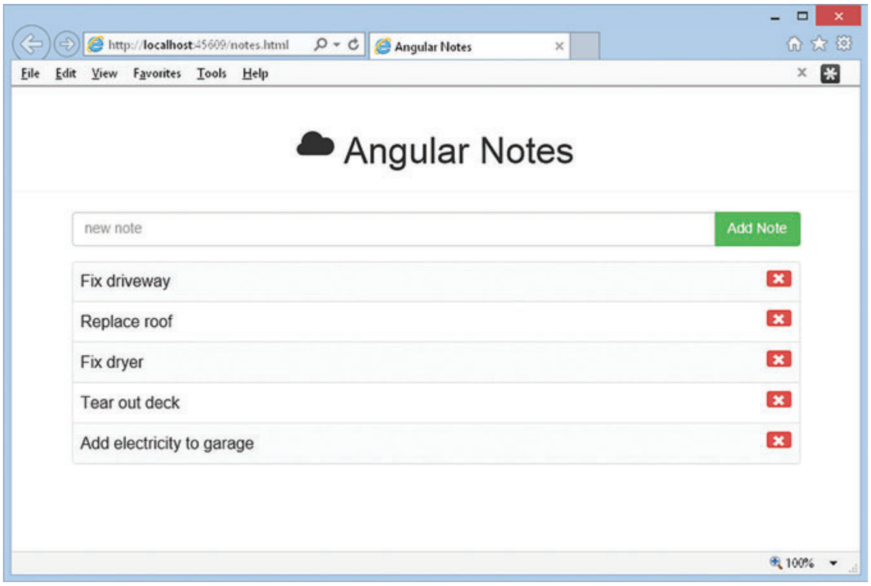
Atalasoft DotImage®

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.

Atalasoft
A Kofax Company

www.atalasoft.com

Image-enabling experts & bacon connoisseurs. Visit us online to see our full line of SDKs for .NET, Java, and Mobile.


http://localhost:45609/notes.html
Angular Notes
File Edit View Favorites Tools Help
Angular Notes
new note
Add Note
100%

Fix driveway
Replace roof
Fix dryer
Tear out deck
Add electricity to garage

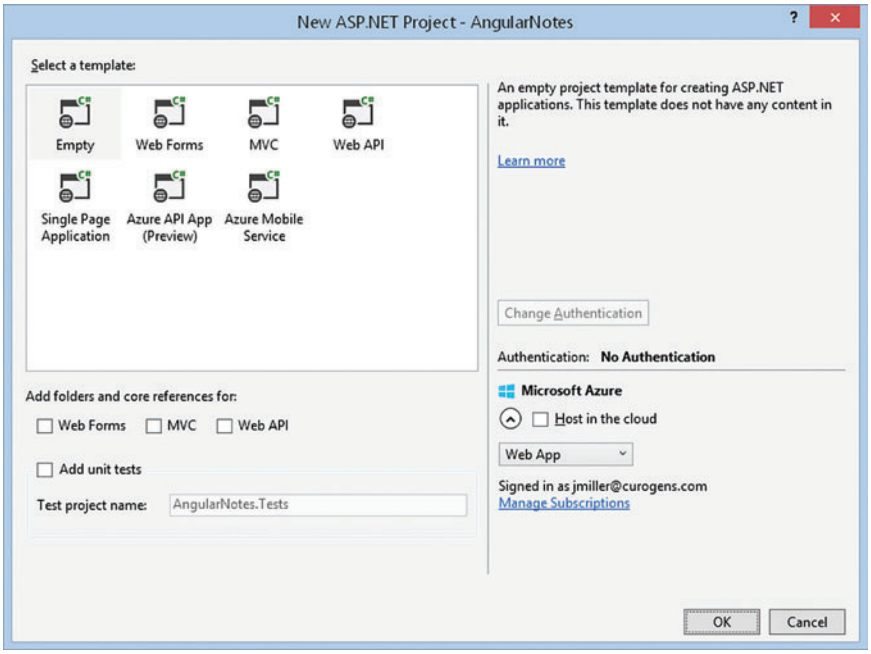
Figure 1 **Angular Notes App**

My Angular Notes App

To demonstrate how to connect AngularJS and AMS, I'm going to put together a very simple Angular Notes application. This app will be a single page and will consist of a list of notes. Each note will have a delete button next to it. There will also be a text box for adding a new note to the list. **Figure 1** shows what my Angular Notes app will look like.

For this app, I'll use Visual Studio 2013 Update 4 and the Azure account I receive with my MSDN benefits. To follow along, first create a new ASP.NET Web Application. Choose the empty template and no options, as shown in **Figure 2**.

Now you should add the AngularJS and Bootstrap libraries, so add the NuGet packages for Angular.Core and Bootstrap.



An empty project template for creating ASP.NET applications. This template does not have any content in it.

Authentication: **No Authentication**

Test project name: AngularNotes.Tests

Signed in as jmiller@curogens.com

Figure 2 **Creating an Empty ASP.NET Project**

To create the initial view, add a new HTML page to the root of the project called notes.html. Modify the HTML so it matches **Figure 3**. Notice that it references Angular and Bootstrap. It also has the ng-app tag, which tells Angular to process this page. Last, it has a body section with an ng-controller tag for the controller I'm going to build later. I've sprinkled in some Bootstrap classes to make it look nice. They aren't required and can be ignored.

To add the list of notes, inside the body section add a new div to the bottom, as shown in **Figure 4**. This *div* will loop through the list of notes and display a table row for each. It also puts a delete button on each row. The key tag that makes this work is the ng-repeat tag, which loops through the notes array on the controller.

Finally, to create a new note box, add one last div to this view so a user can create a new note. Place it above the div for the notes table. In the following code, notice that the input box contents are data-bound to vm.addNoteText and that clicking the button or pressing enter will call the method vm.addNote on the controller:

```
<div class="input-group" style="padding-bottom:15px">
  <input type="text" class="form-control" ng-model="vm.addNoteText"
    placeholder="new note" ng-keypress="($event.which === 13)?vm.addNote():0" />
  <span class="input-group-btn">
    <button ng-click="vm.addNote()" class="btn btn-success">Add Note</button>
  </span>
</div>
```

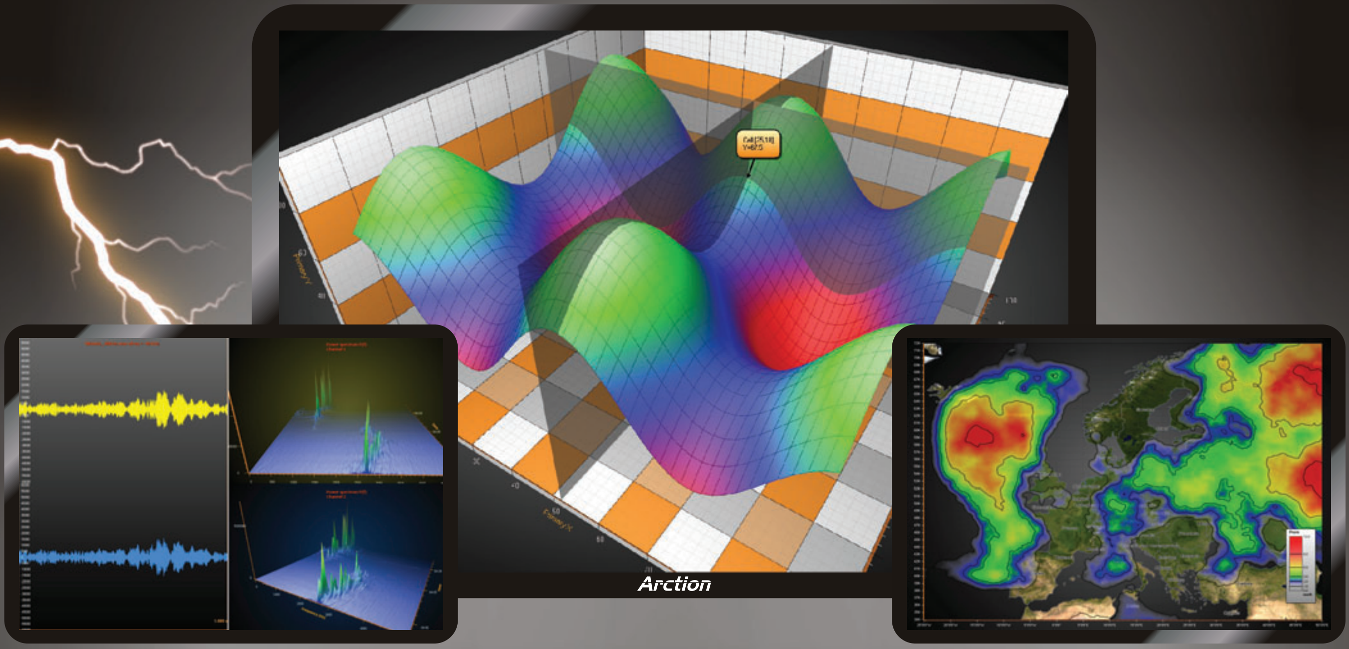
To add the controller, create a new JavaScript file in the root of the project called notesCtrl.js. **Figure 5** shows the code for the entire controller. It consists of the initial notes array to be displayed, the addNote function that adds an item to the array, and the deleteNote function that removes a note from the array. Make sure there's a reference to this script in the notes.html view.

At this point, the core Angular application is done. Running the page displays a list of notes. Clicking the red X next to a note removes it. Typing a new note into the text box and clicking Add Note will add it to the list. So far, however, the list is in memory only. Refreshing the page will bring back the original list, and all the changes will be lost. Let's move the data storage out of memory and into the cloud.

Storing Data in Azure Mobile Services

I'm going to change the storage mechanism for the notes. Instead of using a static, in-memory array, I'm going to load and save my notes from AMS.

In the Azure Portal, create a new Azure Mobile Service. Make sure it uses the JavaScript back end. Next, click on the Data tab inside the mobile service and then create a new data table as shown in **Figure 6**.


Arction

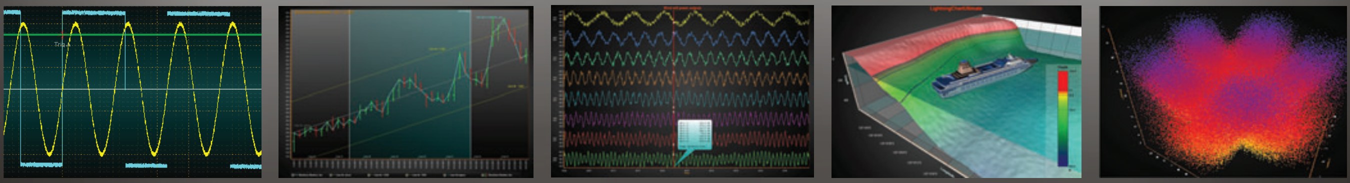


Figure 3 **Initial HTML View**

```
<html ng-app="notesApp">
<head>
  <title>Angular Notes</title>
  <link type="text/css" rel="stylesheet" href="Content/bootstrap.css" />
  <script src="Scripts/angular.js"></script>
  <script src="notesCtrl.js"></script>
</head>
<body ng-controller="notesCtrl as vm">
  <div class="page-header text-center">
    <h1>
      <span class="glyphicon glyphicon-cloud" aria-hidden="true"></span>
      <span style="padding-bottom:10px">Angular Notes</span>
    </h1>
  </div>
</body>
</html>
```

Next, to add the note text column, add a string column named notetext to the notes table.

Now you need to get the application key. On the main Azure Portal page of Azure Mobile Services, you'll see a button labeled Manage Keys. Click this to retrieve the application key and save it somewhere for later. The Angular application will need it to access AMS.

To get AMS working in the Angular app, you need to add two script references to the Notes.html view. The first is for the JavaScript library provided by Microsoft for Azure Mobile Services. The second is an Angular-style service that wraps the Microsoft library. The main benefit of this library is that it extends the Microsoft JavaScript library for AMS and provides Angular-style interfaces and promises. You'll find excellent sample code for this library on GitHub at bit.ly/1po76Vl.

Make sure these new entries appear between the angular.js and the notesCtrl.js references, like so:

```
<script src="Scripts/angular.js"></script>
<script src="http://ajax.aspnetcdn.com/ajax/mobileservices/
MobileServices.Web-1.1.2.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/angular-azure-
mobile-service/1.3.4/angular-azure-mobile-service.min.js"></script>
<script src="notesCtrl.js"></script>
```

To reference Azure Mobile Services in the controller, change the first line of NotesCtrl.js and add a dependency to 'azure-mobile-service.module':

```
angular.module('notesApp', ['azure-mobile-service.module'])
```

Add a constant to the bottom of the notesCtrl.js file with the URL of the Azure mobile site and the application key retrieved earlier. The AMS library will use these to access the AMS site:

```
angular.module('notesApp').constant('AzureMobileServiceClient', {
  API_URL: "https://angularnotes.azure-mobile.net/",
  API_KEY: "gkwGJioLD3jNxrAX6krXh6jVk6SFkeQr",
});
```

Figure 4 **Adding the List of Notes**

```
<div class="container">
  <div class="panel panel-default">
    <table class="table table-striped">
      <tr class="list-group" ng-repeat="note in vm.notes">
        <td>
          {{note.notetext}}
          <button class="btn btn-xs btn-danger pull-right"
            ng-click="vm.deleteNote(note)">
            <span class="glyphicon glyphicon-remove" aria-hidden="true"></span>
          </button>
        </td>
      </tr>
    </table>
  </div>
</div>
```

Now replace the code in the controller that set vm.notes to a static array with data retrieved from AMS. This code retrieves the entire notes table and stuffs the result into the vm.notes array:

```
Azureservice.query('notes', {})
.then(function (items)
{
  vm.notes = items;
});
```

Next, change the vm.addNote function so it saves new notes in the notes data table. Once AMS returns that it's successful, the item is added to the in-memory array. This keeps the application from having to reload the whole list every time something is added to it:

```
vm.addNote = function () {
  if (vm.addNoteText !== '') {
    Azureservice.insert('notes', {
      "notetext" : vm.addNoteText
    }).then(function (newitem) {
      vm.notes.push(newitem);
      vm.addNoteText = '';
    });
  }
}
```

This is one of the main strengths of AMS—it's easy to set up and easy to integrate.

Finally, change the vm.deleteNote function so it deletes the note from the AMS notes table. Once again, the code waits until AMS succeeds and then deletes it from the in-memory array:

```
vm.deleteNote = function (note) {
  Azureservice.del('notes', {
    "id": note.id
  }).then(function () {
    vm.notes.splice(vm.notes.indexOf(note), 1);
  });
}
```

Now all of the notes are retrieved from the notes table in AMS. When a user adds or deletes a note, those operations happen to the data table in AMS. I had to write very little code to achieve

Figure 5 **Adding the Controller**

```
angular.module('notesApp', [])
  .controller('notesCtrl', function () {
    var vm = this;
    vm.addNoteText = '';

    vm.notes = [
      { "notetext": "Fix driveway" },
      { "notetext": "Replace roof" },
      { "notetext": "Fix dryer" },
      { "notetext": "Tear out deck" },
      { "notetext": "Add electricity to garage" }
    ];

    vm.addNote = function () {
      if (vm.addNoteText !== '') {
          vm.notes.push({ "notetext": vm.addNoteText });
          vm.addNoteText = '';
      }
    }

    vm.deleteNote = function (note) {
      vm.notes.splice(vm.notes.indexOf(note), 1);
    }
  });
```

msdn magazine VP

MSDN MAGAZINE VENDOR PROFILE

TEXT CONTROL

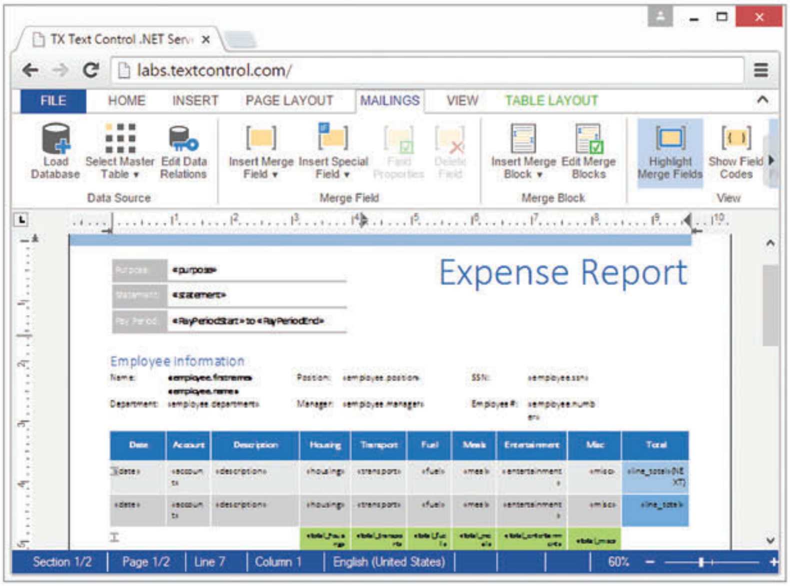
Combine Powerful Reporting with Easy-to-Use Word Processing

A Reporting Q&A with Bjoern Meyer, President Text Control, LLC - USA

Text Control is an award-winning Visual Studio Industry Partner and leading vendor of reporting and word processing components for developers of Windows, web and mobile applications.

Q What is the Text Control Reporting Framework?

A The Text Control Reporting Framework combines powerful reporting features with an easy-to-use, MS Word compatible word processor. Users can create documents and templates using ordinary Microsoft Word skills. It is completely independent from MS Word or any other third-party application and can be completely integrated into business applications. The Text Control Reporting Framework is included in all .NET based TX Text Control products including ASP.NET, Windows Forms and WPF.



Expense Report

Purpose: «purpose»
Statement: «statement»
Pay Period: «PayPeriodStart» to «PayPeriodEnd»

Employee Information

Name: «employee.firstname» «employee.name»
Position: «employee.position»
SSN: «employee.ssn»
Department: «employee.department»
Manager: «employee.manager»
Employee #: «employee.number»

Date	Account	Description	Housing	Transport	Fuel	Meals	Entertainment	Misc	Total
«date»	«account»	«description»	«housing»	«transport»	«fuel»	«meals»	«entertainment»	«misc»	«line_total»(NEXT)
«date»	«account»	«description»	«housing»	«transport»	«fuel»	«meals»	«entertainment»	«misc»	«line_total»
			«total_housing»	«total_transport»	«total_fuel»	«total_meals»	«total_entertainment»	«total_misc»	

Q What sets Text Control Reporting apart from other reporting vendors?

A Text Control Reporting is based on the powerful word processing component TX Text Control. The MS Word compatible template can be merged with a data object (business object) or database content with one line of code. At the same time, Text Control provides a powerful API to customize this merge process completely. The report generation can be fully integrated into .NET applications.

Q Tell us more about the cross-browser, cross-platform WYSIWYG HTML5-based template editor.

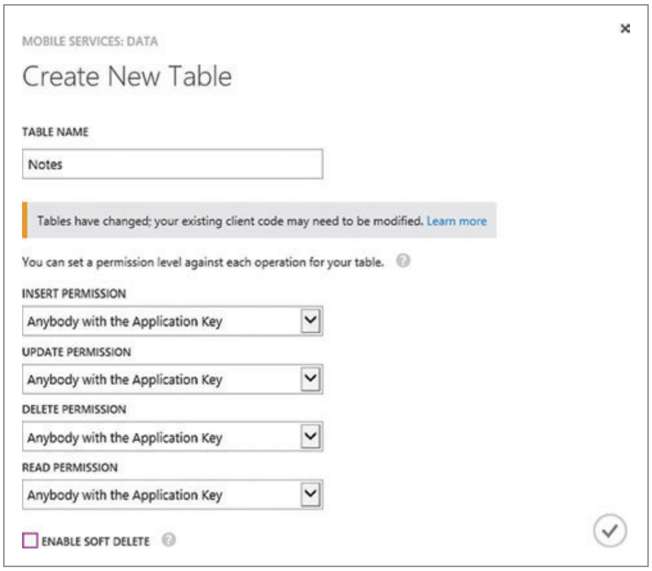
A TX Text Control allows the creation of MS Word compatible documents and templates using any operating system with an HTML5 capable browser including Chrome, Firefox, Safari, Internet Explorer and Edge. Because the product is being built with pure HTML5 and JavaScript, it will have a zero footprint with no client-side browser plugins required. It includes all sophisticated word processing features such as headers and footers, sections, text frames, tab positions, charts, barcodes and spell checking.

Q How do developers typically use Text Control components?

A Our products help thousands of developers add mail merge, reporting and word processing functionality to their Windows Forms, WPF and ASP.NET applications. They create elegant and powerful business reports with an easy-to-use, MS Word compatible template editing interface. Using TX Barcode .NET, developers add fast and accurate 1D and 2D barcodes to.NET based applications. Barcodes are integrated into reports, invoices and mail merge templates. TX Spell .NET enables extremely fast, highly reliable and very accurate spell checking in TX Text Control based applications.

For more detailed information and a 30-day trial version, visit www.textcontrol.com/reporting.

For more information, visit → www.textcontrol.com



MOBILE SERVICES: DATA

Create New Table

TABLE NAME: Notes

Tables have changed; your existing client code may need to be modified. Learn more

You can set a permission level against each operation for your table.

INSERT PERMISSION: Anybody with the Application Key

UPDATE PERMISSION: Anybody with the Application Key

DELETE PERMISSION: Anybody with the Application Key

READ PERMISSION: Anybody with the Application Key

□ ENABLE SOFT DELETE

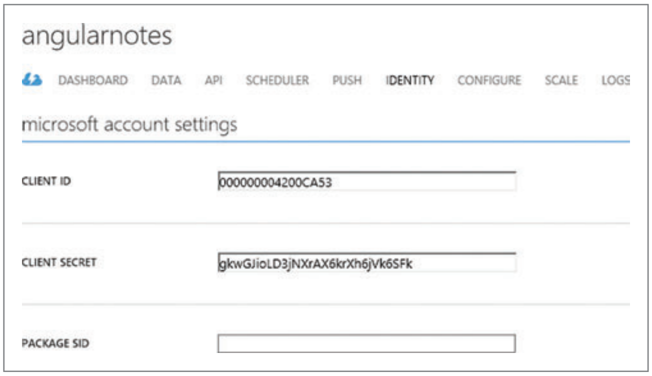
Figure 6 **Creating a New Notes Table**

this. This is one of the main strengths of AMS—it's easy to set up and easy to integrate.

Authenticating Users in Azure Mobile Services

Adding authentication to a Web app can be a hassle. There's always the question of whether to roll your own. These days, unless you really have a compelling case, it almost always makes sense to use one of the major identity providers out there. They can deal with the issues of securely storing passwords and doing resets. AMS makes it easy to connect to the popular identity providers—Microsoft, Facebook, Twitter and Google. AMS seamlessly integrates authentication and authorization with the logon function and the tables created in AMS. For this article, I've chosen to use a Microsoft account for authentication. Once authentication is configured, I'll change the sample so that only authenticated users can view or edit notes and view their own list.

The first step is to set up your app at the Microsoft Live portal (bit.ly/1JS4jq3). This provides the Client ID and Client Secret that AMS needs to get Microsoft Identity working. Now paste them into the Identity tab in AMS as shown in **Figure 7**. There's an excellent article that walks through the process of registering the application at bit.ly/1Ij22hy.



microsoft account settings

CLIENT ID: 000000004200CA53

CLIENT SECRET: gkwGJioLD3jNXrAX6krXh6jVk6SFk

PACKAGE SID:

Figure 7 **Setting up the Microsoft Identity Provider**

Next, add login and logout buttons to the view. Add the following div code above the div that contains the notes table:

```
<div class="text-center">
  <button class="btn btn-primary" ng-if="!vm.isLoggedIn()"
  ng-click="vm.login()">Login</button>
  <button class="btn btn-primary" ng-if="vm.isLoggedIn()"
  ng-click="vm.logout()">Logout</button>
</div>
```

The ng-if code on the login button makes it so the login button is shown only when the user isn't logged in. The ng-if code on the logout button makes it so the logout button is shown only when the user is logged in.

Now, add another ng-if tag to the container div to hide the list and the new note text box when the user isn't logged in. This isn't a security feature. Security will be enforced by AMS. This just makes the page look appropriate:

```
<div class="container" ng-if="vm.isLoggedIn()" style="padding:15px">
```

Next, add the authentication functions to the controller. The isLoggedIn function is used by the view to determine if it should hide or display the login/logout buttons and hide or show the notes list. It just returns the isLoggedIn result from the Azureservice module:

```
vm.isLoggedIn = function ()
{
  return Azureservice.isLoggedIn();
}
```

The app's login function is called when the user clicks the login button. It calls the login function in the Azureservice library. Move the code that queries AMS for the notes list from the top of the controller into this function. Now the list is loaded only after the user authenticates successfully:

```
vm.login = function () {
  Azureservice.login('microsoftaccount')
  .then(function () {
    Azureservice.query('notes', {})
    .then(function (items) {
      vm.notes = items;
    });
  });
}
```

The logout function logs the user out of AMS by calling the logout function on the Azureservice module. It also blanks out the notes array:

```
vm.logout = function () {
  vm.notes = [];
  Azureservice.logout();
}
```

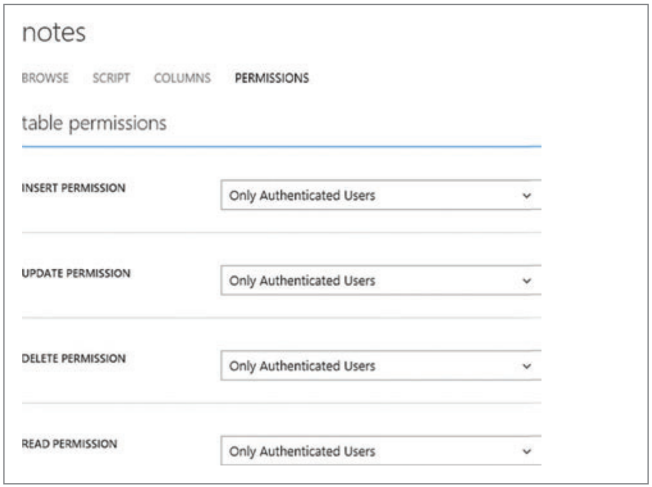


table permissions

INSERT PERMISSION: Only Authenticated Users

UPDATE PERMISSION: Only Authenticated Users

DELETE PERMISSION: Only Authenticated Users

READ PERMISSION: Only Authenticated Users

Figure 8 **Securing the Notes Table**

Right now, the only thing that keeps an unauthenticated user out of the notes list is that the code loads the list only after the user is authenticated. This isn't secure at all. It's far better to have AMS enforce this at the back end. In the Azure Portal, open the AMS service and then open the notes table and click on Permissions. Change all of the permissions to Only Authenticated Users as shown in **Figure 8**. Now any calls to this table will fail if the user isn't authenticated.

Separating User Data

Even though the site authentication is working, all users still share a single list. Let's make it so each user can view and edit only his own notes. This requires just a few changes in the Azure Portal. I won't change the Angular app at all.

First, go into the notes data table. Click on columns and add a new string column called userid. This is how I'll associate a note with a user. Next, go to the script tab on the notes table. Select INSERT from the operation drop-down and add the following to the script:

```
function insert(item, user, request) {
  item.userid = user.userId;
  request.execute();
}
```

This new line sets the new row's userid to the userid from the authentication provider. It's much more secure to do this in the code on the back end. The code runs in Azure; the user (or attacker) doesn't have access to it.

To filter the notes returned by userid, select READ from the operation drop-down and change its script:

```
function read(query, user, request) {
  query.where({ userid: user.userId });
  request.execute();
}
```

The new query.where line filters the rows returned by the userid column; the logged-in user's userid is supplied as the value. Filtering the data on the server before it reaches the client is a much more secure method than trying to do it in the client code.

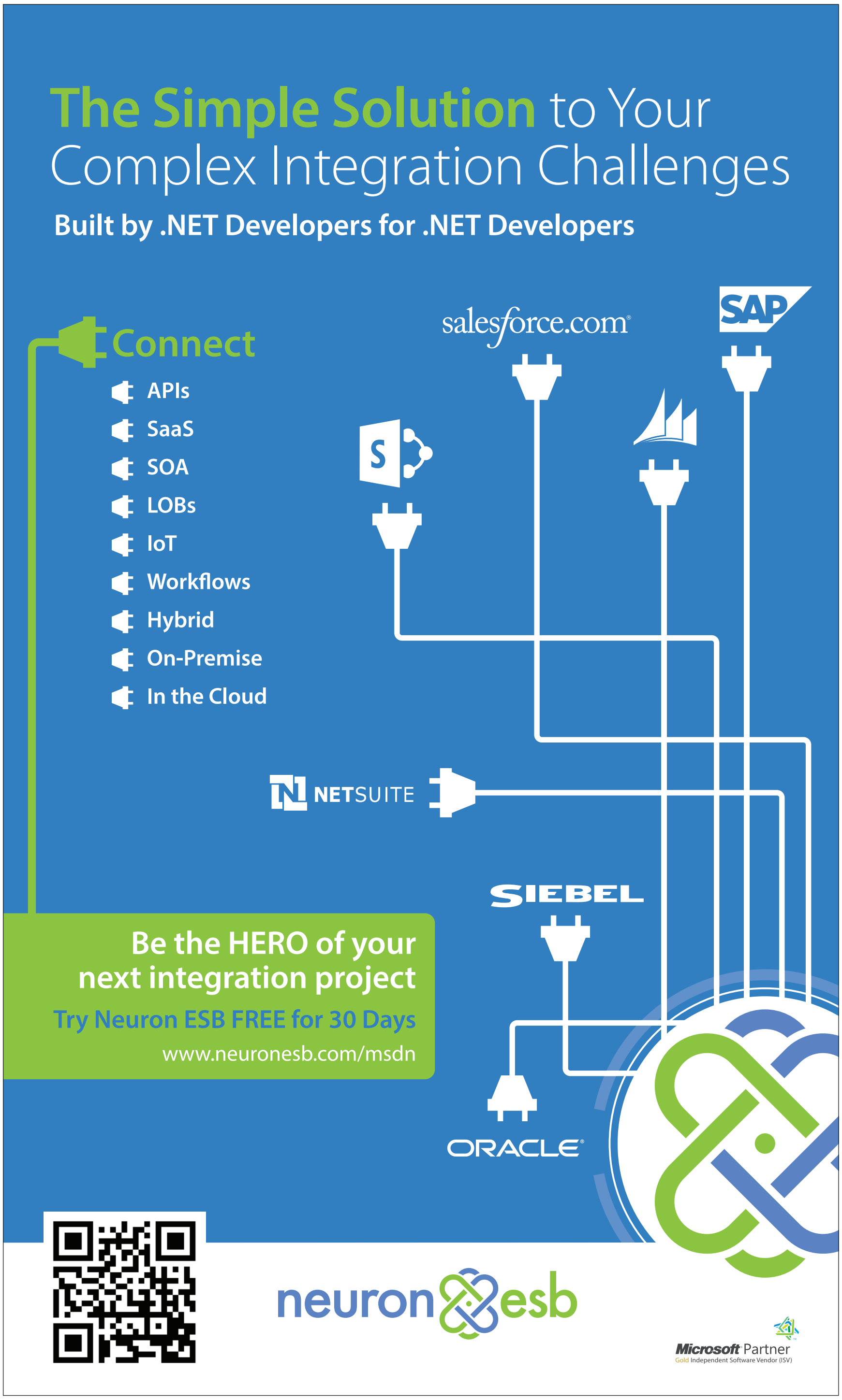
Now the Angular Notes application securely stores notes in Azure Mobile Services. Each user has a separate list of notes that can be accessed only by authenticating first.

Wrapping Up

With just a small amount of code, this demo application is now cloud-powered with first-class storage and authentication. Don't be fooled into believing that if you adopt AngularJS on the front end, you have to abandon the Microsoft stack on the back end. AMS integrates with AngularJS seamlessly. Azure Mobile Services is a great back end for AngularJS applications. ■

JONATHAN MILLER *is a senior architect at CuroGens in Indianapolis. He's been developing products on the Microsoft stack for a decade and programming on .NET since its inception. Miller is a full-stack product developer with expertise in front-end technologies (Windows Forms, Windows Presentation Foundation, Silverlight, ASP.NET, AngularJS/Bootstrap), Middleware (Windows services, Web API), and back ends (SQL server, Microsoft Azure).*

THANKS to the following Microsoft technical experts for reviewing this article: *David Crawford and Simon Gurevich*


The Simple Solution to Your Complex Integration Challenges
Built by .NET Developers for .NET Developers
Connect
APIs
SaaS
SOA
LOBs
IoT
Workflows
Hybrid
On-Premise
In the Cloud
salesforce.com
SAP
NETSUITE
SIEBEL
ORACLE
Be the HERO of your next integration project
Try Neuron ESB FREE for 30 Days
www.neuronesb.com/msdn
neuron esb
Microsoft Partner
Gold Independent Software Vendor (ISV)

Introduction to 3D for Web Games

Michael Oneppo

Adding a third dimension to a game truly brings it to life. You can look around from any viewpoint and see every angle of an object or a scene. But how can you actually pull this off behind the scenes? In this series of articles, I'll walk through the steps for making 3D games, and show you how libraries such as three.js can help you achieve the rich, 3D environment becoming so popular on the Web. In this first installment, I'll keep it simple and focus on building a 3D version of the Ping game first described in "A Web Game in an Hour" (msdn.microsoft.com/magazine/dn913185).

The Illusion of 3D

Any 3D graphics rendering has a surprising trick up its sleeve. Humans can't really see in 3D dimensions—especially on a computer monitor. The entire goal of 3D drawing is to generate, or render, a 3D description of a scene onto a 2D image. When you add a third dimension to get more immersive and realistic scenes, you have to throw away some data to get an image from a specific viewpoint. This concept is called projection. It's an essential element of what makes 3D graphics work, as shown in the basic 3D scene in **Figure 1**.

This article discusses:

- How to render 3D images on a 2D screen
- Adding perspective and distance to 3D objects
- Animating and shading 3D objects

Technologies discussed:

Three.js, HTML, JavaScript

In this scene, the Z axis recedes upward and backward. If I wanted to actually view it on the screen, I could just drop the Z information from every object as a simple and valid way to project the 3D scene, as shown in **Figure 2**.

As you can see, this isn't exactly Halo. For photorealism, a 3D scene requires three things—a proper camera projection, geometry and shading. I'll cover each of these concepts as I rebuild the Ping game as a 3D dueling game.

Getting Started

First, I'll set up the three.js library. This is a fairly quick configuration, as almost everything you do with three.js happens in JavaScript. Here's the HTML code you'll need:

```
<html>
  <head>
    <title>Ping!</title>
    <script src=
      "//cdnjs.cloudflare.com/ajax/libs/three.js/r69/three.min.js"></script>
    <script src="ping3d.js"></script>
  </head>
  <body>
  </body>
</html>
```

In the JavaScript file ping3d.js, I'm going to set up three.js to render a simple scene. First, I need to initialize three.js and add its drawing canvas to the page:

```
var scene = new THREE.Scene();
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
document.body.appendChild( renderer.domElement );
```

The scene is exactly what it sounds like—an object that describes our scene and all the objects within. The renderer is also obviously

msdn magazine VP

MSDN MAGAZINE VENDOR PROFILE

INFRAGISTICS®

INFRAGISTICS ULTIMATE

Build great apps by design when you combine rapid prototyping and the UI controls found in Infragistics Ultimate.

Work Smarter, Not Harder

Don't waste valuable coding time on round after round of revisions: discover the right design, code-free, with our rapid prototyping tool, Indigo Studio.

Top-Notch Modern Web Development Tools

Migration problems are a thing of the past when you use Ignite UI MVC, powered by the rich set of JavaScript-based jQuery UI/HTML5 controls available in Ignite UI.

One Codebase, Three Experiences

Now you can build native apps backed by Infragistics' own native iOS, Android & Windows Phone toolsets, with Infragistics Xamarin.Forms UI controls. Simply use your current C# & XAML skills to create your native apps in the same way as traditional cross-platform applications, by using a single codebase.

Web Controls with Flexibility & Speed

When it comes to your Web apps, performance matters. That's why you can trust Infragistics ASP.NET and Silverlight controls to bring modern, trend-setting applications to market while shortening development time.

Amaze Your Users with Our Native Mobile Tools

Charts with market-leading speed, grids that can easily handle thousands of data points, modern style: Infragistics lightweight mobile iOS and Android controls have it all.

Desktop Apps that Deliver Superior UX

For pixel-perfect, Office-inspired interfaces with a modern touch-enabled experience that your users will love, our Windows Forms and WPF components help you build apps with speed, scalability, and style.

To learn more please visit our website → **www.infragistics.com** named. When given a scene, the renderer will draw it to the screen. This should look similar to some of the 2D drawing systems I described in previous articles, "A Web Game in an Hour," "2D Drawing Techniques and Libraries for Web Games" (msdn.microsoft.com/magazine/dn948109) and "2D Game Engines for the Web" (msdn.microsoft.com/magazine/dn973016). Now I need to add some elements to the screen.

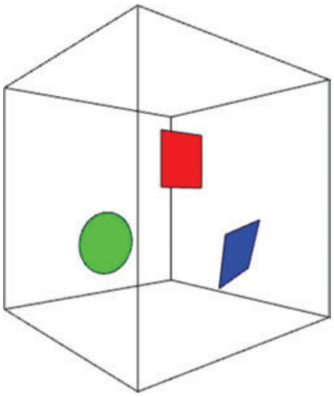


Figure 1 **A Simple 3D Scene**

Geometry

Almost all 3D graphics are built out of polygons. Even curved surfaces like a ball are approximated into triangular facets to approximate its surface. When assembled, these triangles are called a mesh. Here's how I add the ball to the scene:

```
var geometry = new THREE.SphereGeometry(10);
var material = new THREE.BasicMaterial({color: 0xFF0000});
var mesh = new THREE.Mesh(geometry, material);
scene.add(mesh);
```

This code will create a large number of triangles representing a sphere (the "geometry" variable), a simple bright red material (the "material") and a mesh object (the "mesh"). Then it will add the mesh to the scene.

The triangle is the fundamental building block of 3D graphics. Why is this? I'll investigate this further in the next article in this series, but the two main reasons are the straight lines that make up a triangle are easy to work with and you can't break a triangle into a more basic planar surface. The graphics processing unit (GPU) on your computer or phone has dedicated hardware that can quickly convert shapes with straight lines into pixels. This is a good part of what makes high-quality 3D graphics possible.

Modeling

I can pass any geometry into the three.Mesh constructor. This includes generated geometry to make custom shapes or even data from files. For the Ping game, I'd like to have 3D models of each of the players. Therefore, I've taken the liberty of creating geometry in a 3D modeling program for this exercise. It's surprisingly easy to use the model instead of a sphere, as three.js provides a loading tool for this very purpose:

```
var jsonLoader = new THREE.JSONLoader();
jsonLoader.load('tank1.json', function (geometry) {
  var material = new THREE.BasicMaterial({color: 0xFF0000});
  var mesh = new THREE.Mesh(geometry, material);
  scene.add(mesh);
});
```

Camera

The camera represents the viewpoint of the scene. It stores both the position and angle of the viewer within the game. More important, the camera represents how the scene becomes flattened, as described at the beginning of this article.

In my example, the camera was positioned down and to the right. The final image appeared as the view from that direction. However, using this projection method, no matter how far away objects are, they will stay the same size in the final image. This is called an orthographic projection.

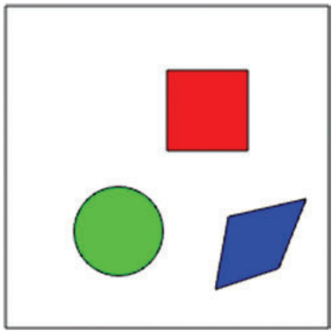


Figure 2 **A Squashed 3D Scene**

This is often useful for games with non-realistic view angles like city simulation games. What I really want to achieve is to make objects appear smaller as they recede into the distance.

Enter perspective projection: Perspective projection imagines the field of view of a camera as a pyramid extending from the lens. When positions are mapped to the screen, they're computed based on their relative distances to the sides of the pyramid. Using this model, as objects recede into the distance, they appear to shrink as in real life.

Thankfully, you don't need to do this mapping yourself because three.js does it for you and provides an object that represents the camera in the scene (and adding another is simple):

```
var camera = new THREE.PerspectiveCamera(
  75, window.innerWidth/window.innerHeight, 0.1, 1000 );
```

Almost all 3D graphics are built out of polygons.

The first argument is the field of view, which indicates how much of an angular distance to take in horizontally. The second argument is the ratio of screen width to height, which you need to ensure things aren't squished because the screen isn't square. The final two parameters define the closest and farthest distances to show. Anything closer or farther than those values isn't drawn. Now I'm at the point where I can actually draw the scene. Let's move the camera back a little to view the whole scene and begin drawing:

```
camera.position.z = 50;
renderer.render(scene, camera);
```

Materials and Lights

Next, I'll place the ball in the arena in which it will bounce:

```
var room = new THREE.BoxGeometry( 50, 30, 100 );
var material = new THREE.MeshPhongMaterial({
    side:  THREE.BackSide,
    map: THREE.ImageUtils.loadTexture('arena.png')
});

var model = new THREE.Mesh(room, material);
model.position.y = 15;
scene.add(model);
```

I'm doing something different than just making box geometry. I'm also making a material. A material is a definition of how something should reflect light in a scene. This generates its overall appearance. In this case, I'm making a Phong material, which is a good default for shiny objects. I'm also adding a texture to the box, which is simple in three.js using the loadTexture function.

One other notable aspect of this code is the line that reads: side: THREE.BackSide. This instructs three.js to draw only the interior sides of the box surfaces, rather than exterior sides. This gives room for the ball to bounce, instead of having a solid box floating in space.

If I were to draw the scene now, the arena wouldn't be visible. It would just draw black. This is because materials define how light reflects off objects, and I do not yet have light in the scene. Three.js makes adding light to a scene simple, as shown here:

```
this.lights = [];

this.lights[0] = new THREE.PointLight( 0x888888, 1, 300 );
this.lights[0].position.set( 0, 10, 40 );
scene.add( this.lights[0] );

this.lights[1] = new THREE.PointLight( 0x888888, 1, 300 );
this.lights[1].position.set( 0, 20, -40 );
scene.add( this.lights[1] );
```

Now if I draw the scene, the arena will render properly. To make a better view, I'll set the camera position to look in the side of the arena before running the code:

```
camera.up.copy(new THREE.Vector3(0,1,0));
camera.position.copy(new THREE.Vector3(0,17, -80));
camera.lookAt(new THREE.Vector3(0,0,40));
```

The first line sets the up variable, which simply tells the camera which way is up. The lookAt function does exactly as it sounds—it points the camera at the specified position.

Making a 3D Game

Now that the game has moved into three dimensions, making the rest should be pretty easy. However, this game is going to end up a bit more verbose than previous implementations because it's composed of 3D instead of 2D objects. So I'll break up the code into separate files to make the additional code easier to handle.

I'll also shift JavaScript styles for object definition to the more traditional constructor model. To demonstrate this, I've wrapped the arena box and lights into an object and placed that in a single file, as shown in **Figure 3**.

If I want to create an Arena, I can create a new object using this constructor function:

```
var arena = new Arena(scene);
```

Next, I'll make a ball object that can bounce around the arena. I know how to make a red ball in three.js, so I'll wrap that code into an object, as well:

```
function Ball(scene) {
  var mesh = new THREE.SphereGeometry(1.5, 10, 10);
  var material = new THREE.MeshPhongMaterial({
    color: 0xff0000,
    specular: 0x333333
  });

  var _model = new THREE.Mesh(mesh, material);
  _model.position.y = 10;
  scene.add(_model);
}
```

Now I'll define the basic physics of bouncing the ball by adding a function to the ball object, as shown in **Figure 4**.

Three.js requires that you render the scene every time using requestAnimationFrame. This should be a familiar pattern:

```
var ball = new Ball(scene);
var Arena = new Arena(scene);
var render = function (time) {
  var t = (time - lastTime) / 1000.0;
  lastTime = time;

  ball.update(t);

  renderer.render(scene, camera);
  requestAnimationFrame( render );
}

requestAnimationFrame(render);
```

Figure 3 **The Arena Object**

```
function Arena(scene) {
  var room = new THREE.BoxGeometry( 50, 30, 100 );
  var material = new THREE.MeshPhongMaterial({
    side:  THREE.BackSide,
    map: THREE.ImageUtils.loadTexture('arena.png')
  });

  var model = new THREE.Mesh(room, material);
  model.position.y = 15;
  scene.add(model);

  this.lights = [];

  this.lights[0]= new THREE.PointLight( 0x888888, 1, 300 );
  this.lights[0].position.set( 0, 10, 40 );
  scene.add( this.lights[0] );

  this.lights[1]= new THREE.PointLight( 0x888888, 1, 300 );
  this.lights[1].position.set( 0, 20, -40 );
  scene.add( this.lights[1] );
}
```

Figure 4 **The Ball Object's Update Function**

```
// Create a private class variable and set it to some initial value.
var _velocity = new THREE.Vector3(40,0,40);

this.update = function(t) {
  // Apply a little gravity to the ball.
  _velocity.y -= 25 * t;

  // Move the ball according to its velocity
  var offset = _velocity.clone()
    .multiplyScalar(t);
  _model.position.add(offset);

  // Now bounce it off the walls and the floor.
  // Ignore the ends of the arena.
  if (_model.position.y - 1.5 <= 0) {
    _model.position.y = 1.5;
    _velocity.y *= -1;
  }

  if (_model.position.x - 1.5 <= -25) {
    _model.position.x = -23.5;
    _velocity.x *= -1;
  }

  if (_model.position.x + 1.5 >= 25) {
    _model.position.x = 23.5;
    _velocity.x *= -1;
  }
}
```

Stay Tuned

Now I have an arena with lights, a well-positioned camera and a ball bouncing around the scene. That's all I'm going to cover in this article. In the next installment, I'll explain how 3D projection works by letting you aim with the mouse. I'll also explain more about textures and make smooth animations using a powerful library called tween.js. In the last of these three articles, I'll look under the hood of three.js and see how it's actually drawing such high-fidelity graphics. ■

Michael Oneppo *is a creative technologist and former program manager at Microsoft on the Direct3D team. His recent endeavors include working as CTO at the technology nonprofit Library for All and exploring a master's degree at the NYU Interactive Telecommunications Program.*

Thanks to the following technical expert for reviewing this article: *Mohamed Ameen Ibrahim*

msdn magazine VP

MSDN MAGAZINE VENDOR PROFILE

LEADTOOLS®
THE WORLD LEADER IN IMAGING SDKs

Convert, View and Annotate 150 File Formats with One LEADTOOLS Control

What do you do when you receive a requirement to view, annotate and save 20, 30, or even 150 different file types including image and document formats using one viewer control? A viewer capable of annotating not only typical document files like PDF and DOC, but also TIFF, JPEG, SVG and more, would be perfect for the job. Meeting this requirement is not an easy task, even for the most experienced programmer, unless you have LEADTOOLS.

The LEADTOOLS Document Converter and Document Viewer can convert, view, and annotate raster and vector based formats in the same control. These unique frameworks satisfy demands and exceed expectations for developers creating ECM, document retrieval and document normalization solutions. With only a few lines of code, programmers can implement document conversion and viewing features that would normally take years to develop.

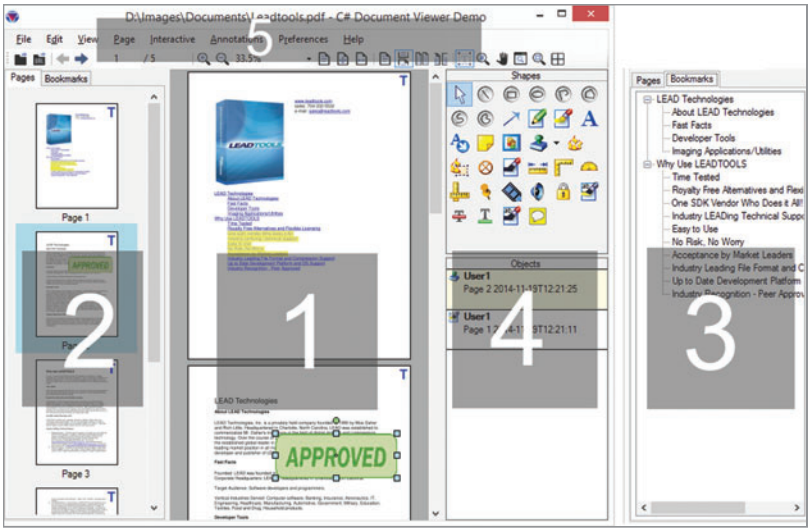
Document Converter

The Document Converter SDK automatically uses a combination of the LEADTOOLS Raster, SVG and OCR engines to convert images and documents using the best possible combination of accuracy and speed. For example, the SVG conversion mode converts between any vector or document format at 100% accuracy without OCR. On the other hand, if the Converter detects a raster input file such as TIFF, it uses OCR to extract the text and convert to a document format.

Document Viewer

The LEADTOOLS Document Viewer is OEM-ready and adds rich document-viewing features including searchable text, annotations, memory-efficient paging, inertial scrolling, and vector display with infinite zoom to applications.

Under the hood, it automatically uses the Document Converter as needed to normalize any document, vector or raster image file into SVG, making it possible to view, search and annotate all file types within a single application.



As seen in the figure above, the Document Viewer is fully customizable and comprised of five components, which you can place anywhere in your layout.

1. Viewer
2. Thumbnail Pane
3. Pages and Bookmarks Pane
4. Annotation Toolbar
5. User Interface Menu and Commands

Simply establish the UI layout with generic containers (e.g. HTML <div>, .NET Panel), then add the Document Viewer components as desired.

Conclusion

Converting and viewing hundreds of document, vector and raster files types with a single viewer control is just one of many real-world solutions you can tackle with LEADTOOLS. Its state-of-the-art Document Viewer and Document Converter frameworks facilitate the creation of dynamic and full-featured document viewing solutions.

→ **Visit www.leadtools.com and download a free evaluation, or give us a call at +1-704-332-5532.**

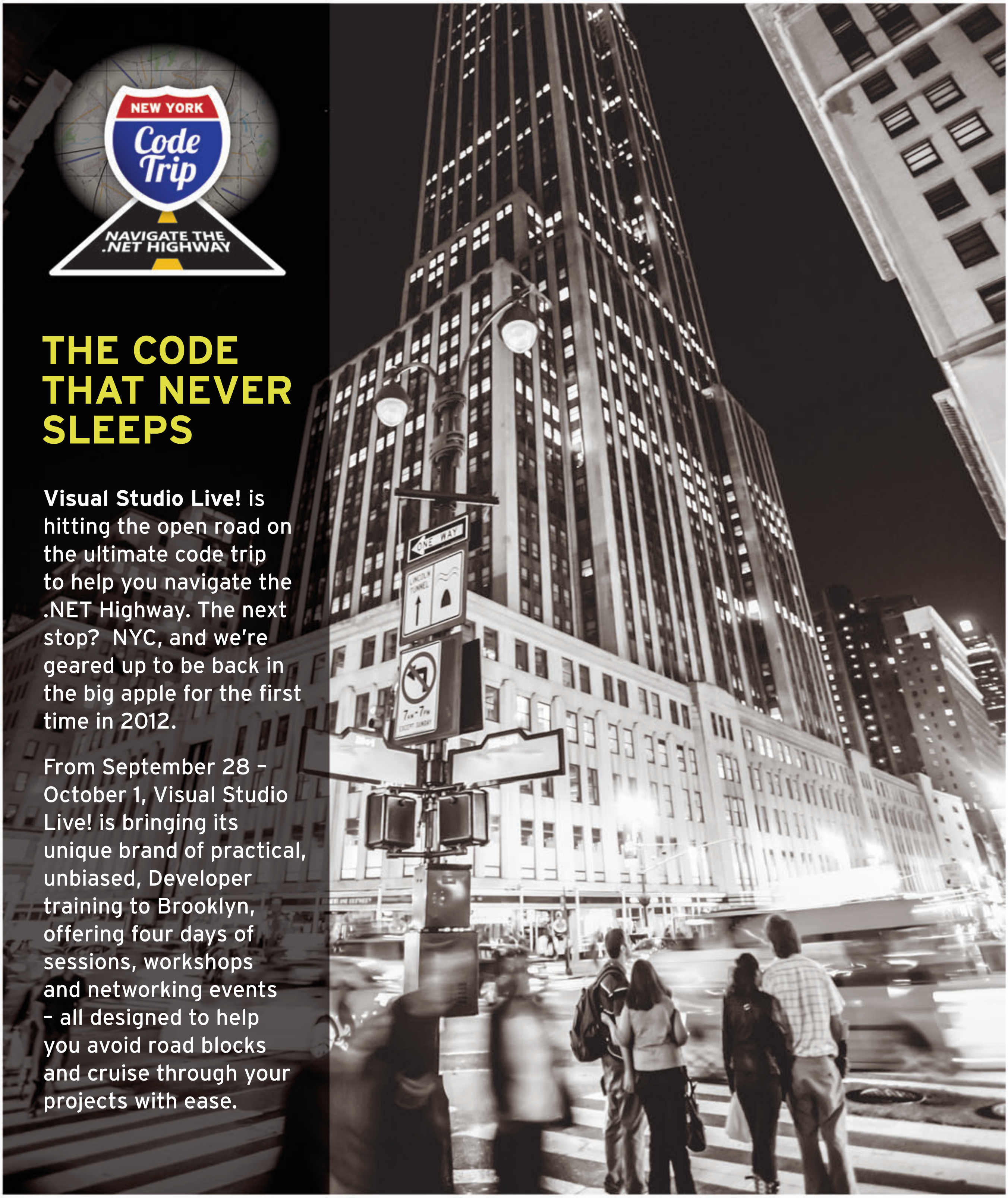
Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

VSLIVE.COM/NEWYORK

New York

SEPTEMBER 28 –
OCTOBER 1

MARRIOTT @ BROOKLYN BRIDGE • NEW YORK, NY



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.

GOLD SPONSOR

GitHub

SUPPORTED BY

Microsoft

Visual Studio

msdn magazine

Visual Studio MAGAZINE

PRODUCED BY

1105 MEDIA INC

NAVIGATE THE .NET HIGHWAY

DEVELOPMENT TOPICS INCLUDE:

- VISUAL STUDIO / .NET
- WEB DEVELOPMENT
- DESIGN
- MOBILE CLIENT
- WINDOWS CLIENT
- DATABASE AND ANALYTICS
- CLOUD COMPUTING

Register by August 5 and Save $300!

USE PROMO CODE NYAUG4

SCAN THE QR CODE TO REGISTER OR FOR MORE EVENT DETAILS.

VSLIVE.COM/NEWYORK

Join us on the Ultimate Code Trip in 2015!

March 16-20 | June 1-4 | June 15-18 | August 10-14 | Sept. 28- Oct. 1 | Nov. 16-20

TURN THE PAGE FOR MORE EVENT DETAILS.

Visual Studio LIVE!

EXPERT SOLUTIONS FOR .NET DEVELOPERS

VSLIVE.COM/NEWYORK

New York

SEPTEMBER 28 – OCTOBER 1

MARRIOTT @ BROOKLYN BRIDGE • NEW YORK, NY

THE HOST HOTEL

The New York Marriott at the Brooklyn Bridge

333 Adams Street
Brooklyn, NY 11201

Special attendee rate: $289*

Book by: September 9, 2015

Located in Renaissance Plaza, the New York Marriott at the Brooklyn Bridge is one block away from nine major subway lines. Take the subway to Manhattan in just 5 minutes, or wander through Brooklyn's best-kept secrets–the Brooklyn Heights Promenade, the sculpture garden at Brooklyn's famous Pratt Institute, and the ultra-hip DUMBO and Williamsburg neighborhoods.

* Rooms are subject to availability and applicable state, local and occupancy taxes.

FEATURED KEYNOTE SPEAKERS

Brian Harry, Corporate Vice President, Microsoft

Mary Jo Foley, Journalist and Author

Register by August 5 and Save $300!

USE PROMO CODE VSLAUG4

SCAN THE QR CODE TO REGISTER OR FOR MORE EVENT DETAILS.

VSLIVE.COM/NEWYORK

FOLLOW US

twitter.com/vslive – @VSLive

facebook.com – Search "VSLive"

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

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

New York AGENDA AT-A-GLANCE

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: Embracing DevOps with Visual Studio** *- 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 - From ASP.NET Site to Mobile App in About an Hour** *- Ryan J. Salva*
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.

vslive.com/newyork

TEST RUN

JAMES MCCAFFREY

K-Means++ Data Clustering

Data clustering is the process of grouping data items so that similar items are placed together. Once grouped, the clusters of data can be examined to see if there are relationships that might be useful. For example, if a huge set of sales data was clustered, information about the data in each cluster might reveal patterns that could be used for targeted marketing.

There are several clustering algorithms. One of the most common is called the k-means algorithm. There are several variations of this algorithm. This article explains a relatively recent variation called the k-means++ algorithm.

Take a look at the demo program in **Figure 1**. The program starts with 20 data items, each consisting of a person's height in inches and weight in pounds. Next, the number of clusters is set to 3. In most data clustering scenarios, the number of clusters must be specified by the user.

The demo program then clusters the data using the k-means++ algorithm. Each of the 20 data items is assigned to one cluster with an ID of 0, 1 or 2. The cluster assignments are stored in an array, where the array index corresponds to a data index and the array value is the associated cluster ID. For example, the final clustering of the demo data is:

```
0 2 1 1 2 . . . 1
```

This indicates data item 0 (height = 65.0, weight = 220.0) is assigned to cluster 0, data item 1 is assigned to cluster 2, data item 2 is assigned to cluster 1, and so on. The demo concludes by displaying the data, grouped by cluster. Here a very clear pattern is revealed. There are eight people characterized by medium height and heavy weight, seven people with short height and light weight, and five people with tall height and medium weight.

This article assumes you have at least intermediate programming skills but doesn't assume you know anything about the k-means++ algorithm. The demo program is coded using C# but you shouldn't have much difficulty refactoring the code to another language, such as Python or JavaScript. The demo code is too long to present in its entirety, but the complete source is available in the code download that accompanies this article.

Understanding the K-Means++ Algorithm

The k-means++ algorithm is a variation of the standard k-means algorithm, so in order to understand k-means++ you must first understand the regular k-means. The k-means algorithm has an interesting history, and is sometimes called Lloyd's algorithm. The "k" in k-means refers to the number of clusters. In very high-level pseudo-code, the most common form of standard k-means is deceptively simple:

```
pick k initial means
loop until no change
  assign each data item to closest mean
  compute new means based on new clusters
end loop
```

In spite of its simple appearance, the standard k-means algorithm is, in fact, very subtle, and implementation is surprisingly tricky. Suppose the data to be clustered consists of the 20 data items shown in **Figure 1**, with k set to 3. The first step is to select three of the data items to act as initial means. A common approach is to select data items at random. Suppose the three randomly selected data items are item 2 (59.0, 110.0) as the mean of cluster 0, item 4 (75.0, 150.0) as the mean of cluster 1, and item 6 (68.0, 230.0) as the mean of cluster 2.

Inside the main processing loop, each data item is examined and assigned to the cluster with the closest mean. So, data item 0 (65.0, 220.0) would be assigned to cluster 2 because item 0 is closer to (68.0, 230.0) than to the other two means. Each of the remaining 19 data items would be assigned to a cluster. Note that the data items that were initially selected as means would be assigned to the correct cluster because the distance would be 0.

After each data item is assigned to one of the clusters, a new mean for each cluster is calculated. Suppose that cluster 0 currently contains just three data items: (59.0, 110.0), (70.0, 210.0), (61.0, 130.0). The new mean for this cluster would be:

```
( (59 + 70 + 61)/3, (110 + 210 + 130)/3 ) =
(190/3, 450/3) =
(63.3, 150.0)
```

New means for clusters 1 and 2 would be calculated similarly. Notice the new means are not necessarily one of the actual data items anymore. Technically, each of the new means is a "centroid," but the term "mean" is commonly used.

After computing new means, each data item is examined again and assigned to the cluster with the closest new mean. The iterative update-clusters, update-means process continues until there's no change in cluster assignments.

This all sounds relatively simple, but a lot can go wrong with a naive implementation of the standard k-means algorithm. In particular, a bad selection for the initial means can lead to a very poor clustering of data, or to a very long runtime to stabilization, or both. As it turns out, good initial means are ones that aren't close to each other. The k-means++ algorithm selects initial means that aren't close to each other, then uses the standard k-means algorithm for clustering.

Code download available at msdn.microsoft.com/magazine/msdnmag0815.

msdn magazine VP

MSDN MAGAZINE VENDOR PROFILE

Innovasys

Documentation and Help Authoring Solutions from Innovasys

Q&A with Richard Sloggett, Chief Technical Officer at Innovasys

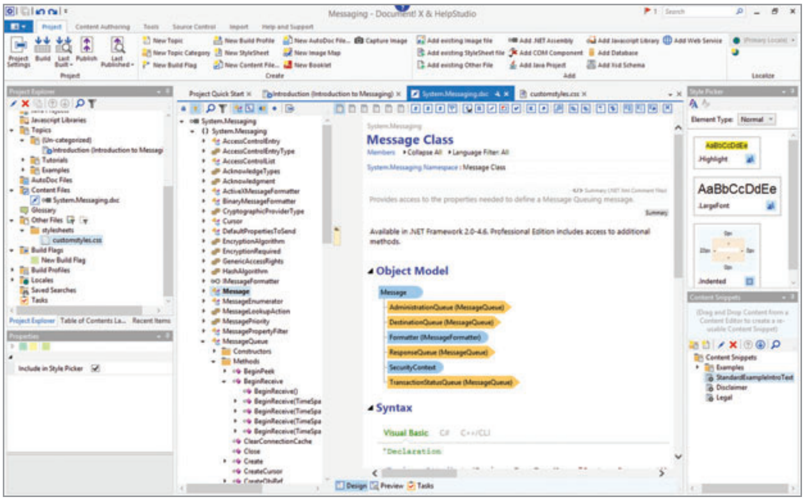
Innovasys has been a leading provider of Documentation and Help Authoring tools since 1998 and is focused on producing tools that enable developers and technical writers worldwide to produce professional quality documentation, help systems and procedures with minimum friction.

Q What is Document! X?
A Document! X is a documentation and help authoring solution that combines the ability to automatically generate documentation with a full featured and mature content authoring environment. This unique combination delivers all the time and cost benefits of automating as much of the documentation workflow as possible whilst retaining the ability to fully customize and supplement the automatically generated output.

Q What kinds of documentation can Document! X help produce?
A Document! X can document .NET assemblies, Web Services (REST or SOAP), Databases (Access, SQL Server, Oracle, OLE DB), XSD Schemas, JavaScript, COM Components, Type Libraries and Java. In addition to supplementing the automatically generated content, you can also author any number of free format conceptual Topics to round out your documentation.

The same rich and mature authoring environment and fully customizable templates are available for all of the supported documentation types.

Q How does Document! X make documentation less painful?
A With Document! X, documentation can be automatically produced throughout design, development and beyond without requiring investment of developer resources, providing development teams with an accurate and up to date reference and allowing new developers to jump the learning curve of new components, schemas or web services. Document! X makes producing documentation a natural and productive activity for developers and technical writers alike.



Q Can Document! X edit Xml comments in Visual Studio?
A The Document! X Visual Comment editor integrated with Visual Studio provides a WYSIWYG environment in which you can create and edit your Xml source code comments. Content can be authored both in source code comments and in Document! X Content Files outside of the source. This approach makes it easy to make post-development edits and combine the efforts of developers and technical writers.

Q Why choose Document! X?
A Document! X has been trusted by developers and technical writers worldwide since 1998 to produce accurate and comprehensive professional quality documentation. We believe that Document! X is the most comprehensive and mature documentation tool available today, backed with first class support, and we encourage you to try our free trial version to document your own components and see if you agree.

For more information and a free trial version, visit → www.innovasys.com

The K-Means++ Initialization Mechanism

In high-level pseudo-code the k-means++ initialization mechanism to select means is:

```
select one data item at random as first mean
loop k-1 times
  compute distance from each item to closest mean
  select an item that has large distance-squared
    as next initial mean
end loop
```

Again, the pseudo-code is deceptively simple. The k-means++ initialization mechanism is illustrated in **Figure 2**. There are nine data points, each of which has two components. The number of clusters is set to 3, so 3 data items must be selected as initial means.

The diagram in **Figure 2** shows the k-means++ initialization process after two of the three initial means have been selected. The first initial mean at (3, 6) was randomly selected. Then the distance-squared from each of the other 8 data items to the first mean was computed, and using that information, the second initial mean at (4, 3) was selected (in a way I'll explain shortly).

To select a data item as the third initial mean, the squared distance from each data point to its closest mean is computed. The distances are shown as dashed lines. Using these squared distance values, the third mean will be selected so that data items with small squared distance values have a low probability of being selected, and data items with large squared distance values have a high probability of being selected. This technique is sometimes called proportional fitness selection.

Proportional fitness selection is the heart of the k-means++ initialization mechanism. There are several ways to implement proportional fitness selection. The demo program uses a technique called roulette wheel selection. In high-level pseudo-code, one form of roulette wheel selection is:

```
p = random value between 0.0 and 1.0
create an array of cumulative probabilities
loop each cell of cum prob array
  if cum[i] >= p
    return i
  end if
end loop
```

A concrete example will help clarify roulette wheel selection. Suppose there are four candidate items (0, 1, 2, 3) with associated values (20.0, 10.0, 40.0, 30.0). The sum of the values is 20.0 + 40.0 + 10.0 + 30.0 = 100.0. Proportional fitness selection will pick item 0 with probability 20.0/100.0 = 0.20; pick item 1 with probability 10.0/100.0 = 0.10; pick item 2 with probability 40.0/100.0 = 0.40; and pick item 3 with probability 30.0/100.0 = 0.30.

If the probabilities of selection are stored in an array as (0.20, 0.10, 0.40, 0.30), the cumulative probabilities can be stored in an array with values (0.20, 0.30, 0.70, 1.00).

Now, suppose a random p is generated with value 0.83. If i is an array index into the cumulative probabilities array, when i = 0, cum[i] = 0.40, which isn't greater than p = 0.83, so i increments to 1. Now cum[i] = 0.60, which is still not greater than p, so i increments to 2. Now cum[i] = 0.70, which is greater than p, so i = 2 is returned as the selected item.

Notice that the distances between the cumulative probabilities differ, with larger differences corresponding to those items with higher probabilities of selection.

To summarize, the k-means++ algorithm selects initial means so the means are dissimilar, then uses the standard k-means algorithm to cluster data. The initialization process uses proportional fitness selection, which can be implemented in several ways. The demo program uses roulette wheel selection.

Overall Program Structure

The overall structure of the demo program, with a few minor edits to save space, is presented in **Figure 3**. To create the demo program, I launched Visual Studio and created a new C# console application project named KMeansPlus. The demo program has no significant Microsoft .NET Framework dependencies so any relatively recent version of Visual Studio will work.

After the template code loaded into the editor, in the Solution Explorer window I right-clicked on file Program.cs, renamed it to the more descriptive KMeansPlusProgram.cs and allowed Visual Studio to automatically rename class Program. In the editor window, at the top of the template-generated code, I deleted all references to namespaces except the ones to the top-level System namespace and the Collections.Generic namespace.

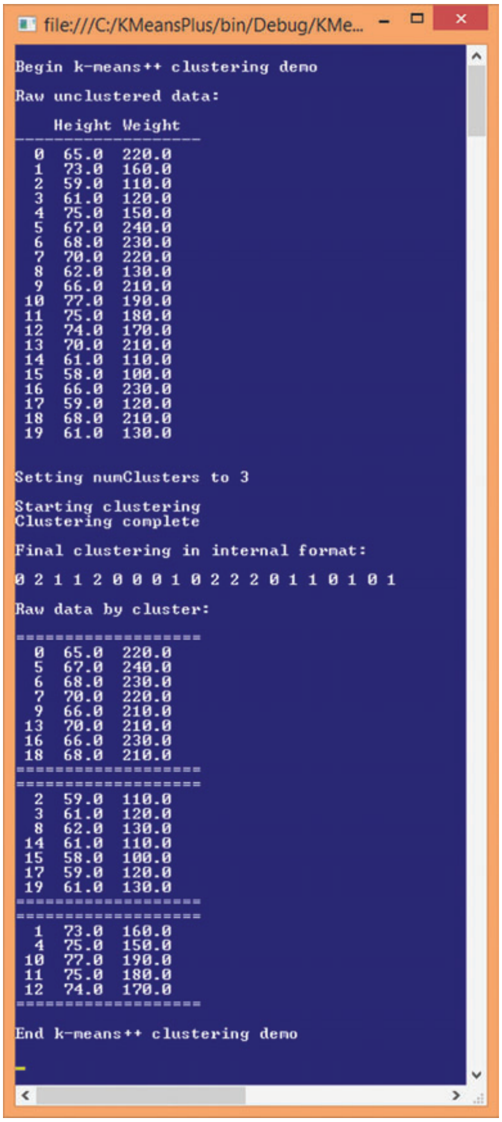
The Main method begins by setting up 20 raw data items:

```
double[][] rawData = new double[20][];
rawData[0] = new double[] { 65.0, 220.0 };
...
rawData[19] = new double[] { 61.0, 130.0 };
```

In a non-demo scenario you'd probably read data from a text file or SQL database. After displaying the raw data using a program-defined helper method named ShowData, the data is clustered:

```
int numClusters = 3;
int[] clustering = Cluster(rawData,
  numClusters, 0);
```

Although there are some techniques you can use to guess the best number of clusters, in general you must use trial and error. The Cluster method accepts numeric raw data to cluster in an array-of-array style matrix; the number of clusters to use (I could have used "k" but "numClusters" is more readable); and a seed value to use for randomization.



```
Begin k-means++ clustering demo

Raw unclustered data:

    Height Weight
-------------------
  0  65.0  220.0
  1  73.0  160.0
  2  59.0  110.0
  3  61.0  120.0
  4  75.0  150.0
  5  67.0  240.0
  6  68.0  230.0
  7  70.0  220.0
  8  62.0  130.0
  9  66.0  210.0
 10  77.0  190.0
 11  75.0  180.0
 12  74.0  170.0
 13  70.0  210.0
 14  61.0  110.0
 15  58.0  100.0
 16  66.0  230.0
 17  59.0  120.0
 18  68.0  210.0
 19  61.0  130.0


Setting numClusters to 3

Starting clustering
Clustering complete

Final clustering in internal format:

0 2 1 1 2 0 0 0 1 0 2 2 2 0 1 1 0 1 0 1

Raw data by cluster:

===================
  0  65.0  220.0
  5  67.0  240.0
  6  68.0  230.0
  7  70.0  220.0
  9  66.0  210.0
 13  70.0  210.0
 16  66.0  230.0
 18  68.0  210.0
===================
===================
  2  59.0  110.0
  3  61.0  120.0
  8  62.0  130.0
 14  61.0  110.0
 15  58.0  100.0
 17  59.0  120.0
 19  61.0  130.0
===================
===================
  1  73.0  160.0
  4  75.0  150.0
 10  77.0  190.0
 11  75.0  180.0
 12  74.0  170.0
===================

End k-means++ clustering demo

_
```

Figure 1 **K-Means++ Clustering in Action**

msdn magazine VP

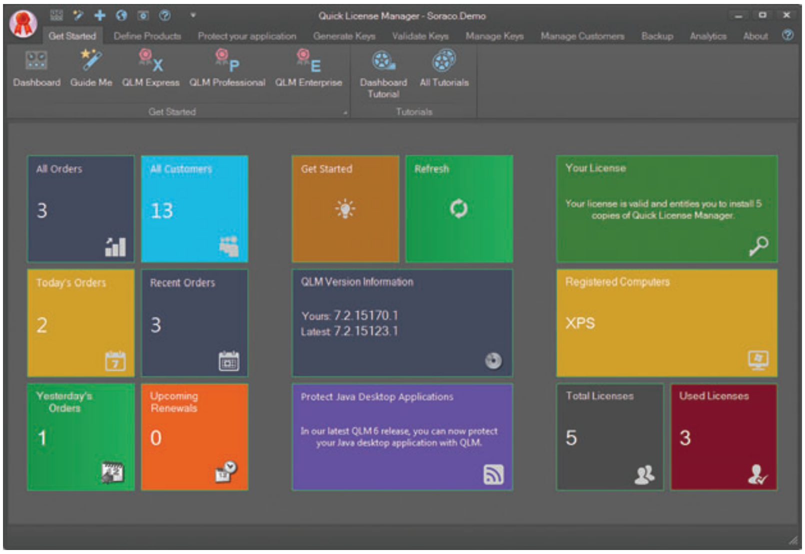
MSDN MAGAZINE VENDOR PROFILE

SORACO

Quick License Manager Protects Your Application with Secure License Keys

Most software applications are now protected with license keys. The key to success is full automation of the license key distribution and management process. You want to focus on your business. Licensing should be seamless and just work. Quick License Manager (QLM) provides the framework and the tools to fully automate the license key lifecycle.

eCommerce Provider Integration: We start off by integrating with leading ecommerce providers to issue a license key seamlessly when a user purchases your product. Through our ecommerce provider integration, the QLM license server is automatically contacted when an order is placed and is provided with detailed order information. QLM generates the proper license key and sends it back to the eCommerce provider. The license key is then embedded in the order confirmation email that is sent to the customer.



All Orders 3

All Customers 13

Your License

Your license is valid and entitles you to install 5 copies of Quick License Manager.

Today's Orders 2

Recent Orders 3

QLM Version Information

Yours: 7.2.15170.1
Latest: 7.2.15123.1

Registered Computers XPS

Yesterday's Orders 1

Upcoming Renewals 0

Protect Java Desktop Applications

In our latest QLM 6 release, you can now protect your Java desktop application with QLM.

Total Licenses 5

Used Licenses 3

Online Activation: Once a user receives his license key, he needs to activate the license. The activation process ensures that a given key will only work on a single system by binding the license to a specific computer identifier. QLM provides a ready to use wizard for online activation. Of course, offline activation is also a must for users who do not have internet access.

Subscription Model: Subscription based models have now become prevalent. Through our ecommerce integration, when a subscription is renewed, the end user's license can be automatically extended without any user interaction or the need to issue new license keys. It all just happens behind the scene without any user input.

Resellers: The QLM Portal is a web interface that empowers your resellers to manage their customers while keeping you, the vendor, in total control. You define what operations a reseller can perform, limit the number of keys they can create and which products they have access to. At any time, you can generate reports for each reseller to view their activity.

Platforms: QLM can protect your Windows applications (desktop or web) as well as your iOS, Android or OS-X applications.

License Key Management: The QLM Management Console is a feature rich and complete license management interface that allows you to manage your customers and license keys. In addition, QLM can mass email your customers notifications about specific events. Our fully integrated backup tool can backup your QLM database to your local system without the need for third-party products.

For more information, visit → soraco.co

The Main method concludes by displaying the clustering array and showing the raw data grouped by cluster:

```
ShowVector(clustering, true);
ShowClustered(rawData, clustering, numClusters, 1);
```

I used a static method approach rather than an OOP approach. Method Cluster calls four helper methods. Helper method Normalized accepts a matrix of raw data and returns a matrix where the data has been normalized so that all values are roughly the same magnitude (typically between -6.0 and +6.0). Method InitMeans implements the k-means++ initialization mechanism. Methods UpdateClustering and UpdateMeans implement the core parts of the standard k-means algorithm.

Methods InitMeans and UpdateClustering both call helper method Distance, which returns the Euclidean distance between two data items. For example, if one data tuple is (3.0, 9.0, 5.0) and a second tuple is (2.0, 6.0, 1.0), the Euclidean distance between the two items is:

```
Sqrt( (3-2)^2 + (9-6)^2 + (5-1)^2) ) =
Sqrt( 1 + 9 + 16 ) =
Sqrt(26) = 5.1
```

Other distance definitions can be used. In general, k-means and k-means++ are used to cluster strictly numeric data rather than categorical data.

Implementing K-Means++

The code for method Cluster is presented in **Figure 4**. Method Cluster begins by normalizing the raw data so that large components in data items (such as weight values in the demo) don't dominate smaller components (height values). The demo uses Gaussian normalization. Two common alternatives are min-max normalization, and order of magnitude normalization. A design alternative is to normalize your raw data in a preprocessing step, then pass the normalized data directly to method Cluster.

Method InitMeans implements the k-means++ initialization mechanism and returns a set of means that are far apart from each other in terms of Euclidean distance. Inside the main clustering loop, method UpdateClustering iterates through each data item and assigns each item to the cluster associated with the closest current means/centroids. The method returns false if there's no change to cluster assignments (indicating that clustering is complete) or if the new clustering would result in a cluster that has no data items (indicating something is wrong). An alternative is to throw an exception on a zero-count cluster situation.

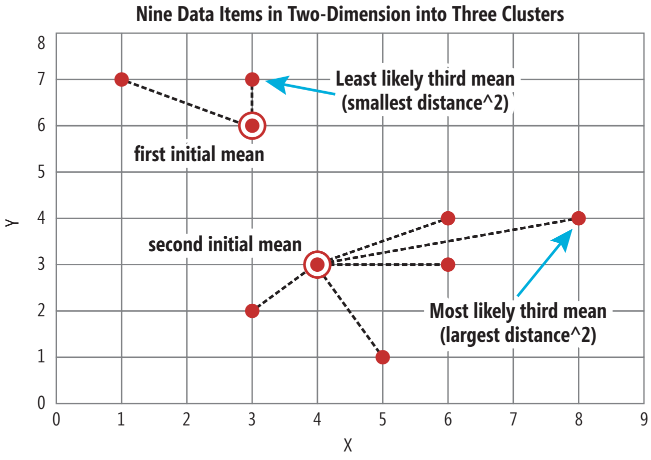


Figure 2 **K-Means++ Initialization Mechanism**

Method UpdateMeans iterates through the data assigned to each cluster and computes a new mean/centroid for each cluster. The method returns false if one or more means can't be calculated because a cluster has no data items.

The main clustering loop uses a sanity count check to prevent an infinite loop. The k-means algorithm typically stabilizes very quickly, but there's no guarantee the algorithm will stabilize at all. The value of maxCount is set to 10 times the number of data items, which is arbitrary but has worked well for me in practice.

The definition of method InitMeans begins with:

```
public static double[][] InitMeans(int numClusters,
  double[][] data, int seed)
{
  double[][] means = MakeMatrix(numClusters, data[0].Length);
  List<int> used = new List<int>();
...
```

The local array-of-arrays style matrix named means holds the method return, where the row index is a cluster ID and each row is an array that holds the components of the associated mean. The List<int> named used holds indices of data items that have been assigned as initial means, so duplicate initial means can be prevented. This approach assumes there are no data items with identical values. When clustering, how

Figure 3 **Overall Program Structure**

```
using System;
using System.Collections.Generic;
namespace KMeansPlus
{
  class KMeansPlusProgram
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Begin k-means++ demo");

      // All program code here

      Console.WriteLine("End k-means++ demo");
      Console.ReadLine();
    }

    public static int[] Cluster(double[][] rawData,
      int numClusters, int seed) { . . }

    public static double[][] InitMeans(int numClusters,
      double[][] data, int seed) { . . }

    private static double[][] Normalized(double[][] rawData) { . . }

    private static double[][] MakeMatrix(int rows, int cols) { . . }

    private static bool UpdateMeans(double[][] data,
      int[] clustering, double[][] means) { . . }

    private static bool UpdateClustering(double[][] data,
      int[] clustering, double[][] means) { . . }

    private static double Distance(double[] tuple,
      double[] mean) { . . }

    private static int MinIndex(double[] distances) { . . }

    static void ShowData(double[][] data, int decimals,
      bool indices, bool newLine) { . . }

    static void ShowVector(int[] vector, bool newLine) { . . }

    static void ShowClustered(double[][] data, int[] clustering,
      int numClusters, int decimals)
  }
} // ns
```

SQL Server LIVE!
TRAINING FOR DBAs AND IT PROS

ORLANDO
ROYAL PACIFIC RESORT AT
UNIVERSAL ORLANDO

NOV
16-20

LIVE! 360
TECH EVENTS WITH PERSPECTIVE

5 Great Conferences
1 Great Price

SQL Server LIVE!
SQL SERVER FOR MODERN DEVELOPERS

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

SharePoint LIVE!
TRAINING FOR COLLABORATION

Modern Apps LIVE!
MOBILE, CROSS-DEVICE & CLOUD DEVELOPMENT

TECHMENTOR
IN-DEPTH TRAINING FOR IT PROS

Driving Data Forward

After 5 days of workshops, deep dives and breakout sessions, SQL Server Live! will leave you with the skills needed to Drive Data Forward.

With timely, relevant content, SQL Server Live! helps administrators, DBAs, and developers do more with their SQL Server investment. Sessions will cover performance tuning, security, reporting, data integration, adopting new techniques, improving old approaches, and modernizing the SQL Server infrastructure.

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

Use promo code SQLAUG1

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

SUPPORTED BY

Visual Studio

msdn magazine

Visual Studio MAGAZINE

PRODUCED BY

1105 MEDIA INC

SQLLIVE360.COM

you deal with duplicate data items depends on your particular problem scenario. One alternative to removing duplicate items from the source data is to weight duplicate items by their frequency.

Next, the first initial mean is selected and stored:

```
Random rnd = new Random(seed);
int idx = rnd.Next(0, data.Length);
Array.Copy(data[idx], means[0], data[idx].Length);
used.Add(idx);
```

The first initial mean is selected at random from all data items. The initial means are existing data items and they are sometimes called medoids.

Next, a for loop is constructed to select the remaining k-1 means:

```
for (int k = 1; k < numClusters; ++k)
{
  double[] dSquared = new double[data.Length];
  int newMean = -1;
...
```

Array dSquared holds the squared distances between each data item and the closest existing initial mean. Variable newMean holds the index of a data item that will be the next initial mean. Next, each (normalized) data item is examined and its dSquared value is computed and stored:

```
for (int i = 0; i < data.Length; ++i)
{
  if (used.Contains(i) == true) continue;
  double[] distances = new double[k];
  for (int j = 0; j < k; ++j)
    distances[j] = Distance(data[i], means[k]);
  int m = MinIndex(distances);
  dSquared[i] = distances[m] * distances[m];
}
```

The check to determine if a data item has already been used as an initial mean isn't really necessary because if the item has been used, the distance squared to the closet mean will be the distance to itself, which is 0. The array named distances holds the Euclidean distances from the current data item to each of the existing k-initial means that have been selected so far.

Recall that Euclidean distance in the Distance method takes the square root of the sum of the squared differences between data item components. Because k-means++ uses squared distances, the squaring operation in InitMeans undoes the square root operation in Distance. Therefore, you could simplify the code by defining a method that returns squared distance directly.

Next, the loop to scan through cumulative probabilities for roulette wheel selection is prepared:

```
double p = rnd.NextDouble();
double sum = 0.0;
for (int i = 0; i < dSquared.Length; ++i)
  sum += dSquared[i];
double cumulative = 0.0;
int ii = 0;
int sanity = 0;
```

Figure 4 **Method Cluster**

```
public static int[] Cluster(double[][] rawData,
  int numClusters, int seed)
{
  double[][] data = Normalized(rawData);
  bool changed = true;
  bool success = true;
  double[][] means = InitMeans(numClusters, data, seed);
  int[] clustering = new int[data.Length];
  int maxCount = data.Length * 10;
  int ct = 0;
  while (changed == true && success == true &&
    ct < maxCount)
  {
    changed = UpdateClustering(data, clustering, means);
    success = UpdateMeans(data, clustering, means);
    ++ct;
  }
  return clustering;
}
```

A random value between 0.0 and 1.0 is generated and the sum of the squared distances is calculated as explained in the section describing proportional fitness selection. Instead of explicitly creating an array of cumulative probabilities, it's more efficient to generate the current cumulative probability on the fly.

Each cumulative probability is computed and examined in a while loop that implements roulette wheel selection:

```
while (sanity < data.Length * 2)
{
  cumulative += dSquared[ii] / sum;
  if (cumulative >= p && used.Contains(ii) == false)
  {
    newMean = ii; // the chosen index
    used.Add(newMean); // don't pick again
    break;
  }
  ++ii; // next candidate
  if (ii >= dSquared.Length) ii = 0; // past the end
  ++sanity;
}
```

The while loop advances until the cumulative probability value is greater than or equal to the random p value. However, duplicate initial means can't be allowed so if the selected mean is in the "used" List<int>, the next available data item is selected. If the ii index runs past the end of the data, it's reset to 0. Note that if a data item has already been selected as an initial mean, the next available data item will probably not be the next most likely item.

Method InitMeans concludes by saving the selected initial mean, and returning the array of selected means:

```
...
    Array.Copy(data[newMean], means[k], data[newMean].Length);
  } // k, each remaining mean
  return means;
} // InitMean
```

The purpose of the InitMeans method is to find k dissimilar data items to be used as initial means. Roulette wheel selection doesn't guarantee that the selected means are maximally different from each other, and there's a very small chance the selected means could be quite close to each other. Therefore, you may want to refactor method InitMeans so that roulette wheel selection is used several times to generate candidate sets of means, and then return the set containing means that are most different from each other.

Wrapping Up

This article is based on the 2007 research paper, "K-Means++: The Advantages of Careful Seeding," by D. Arthur and S. Vassilvitskii. As is usually the case with research papers, virtually no implementation details are given. However, the paper carefully explains how k-means++ initialization works and establishes theoretical boundaries for its behavior. ■

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 Research technical expert for reviewing this article: *Kirk Olynyk*

SEE THE WORLD AS A DATABASE
ADO.NET ▪ JDBC ▪ ODBC ▪ SQL SSIS ▪ ODATA
MYSQL ▪ EXCEL ▪ POWERSHELL
Visual Studio
Java
ODBC
SQL Server
Excel
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

How To Be MEAN: Getting Started

Parallel universes are all the rage, especially if you follow the stories in comic books. Clearly, I've stepped into one myself. When I first started reading this magazine, it was called *Microsoft Systems Journal* (*MSJ* for short). The languages of choice were C++ and Visual Basic. The underlying technology platform was a managed platform (COM) written with both unmanaged languages like C++ and managed languages like Visual Basic. The OS of choice was clearly, unfailingly, unquestionably and unhesitatingly Windows. Perhaps it was Windows 3.0, 3.1, Chicago or NT—but always Windows.

> The MEAN stack is a "full stack" quartet, meaning it covers front end, back end and storage.

Look where we are today. It's clearly not a "Microsoft" world anymore, even within the Microsoft world. Forget, for a moment, the "competitors" against which Microsoft wrestles, such as Java or Ruby. The company has partnerships with those languages, and even supports them natively on the next-generation Microsoft Azure platform. Forget the languages that Microsoft "owns," like C# or Visual Basic or F#. They've all been made open source. As has its Web framework, and its data-access framework. And the new version of Visual Studio will ship with an Android emulator.

And just when we thought it couldn't get more different, Microsoft again went and did something entirely different. As of this writing, Microsoft just announced a partnership with Cyanogen, an Android distributor. Can somebody please tell me which portal I stepped through?

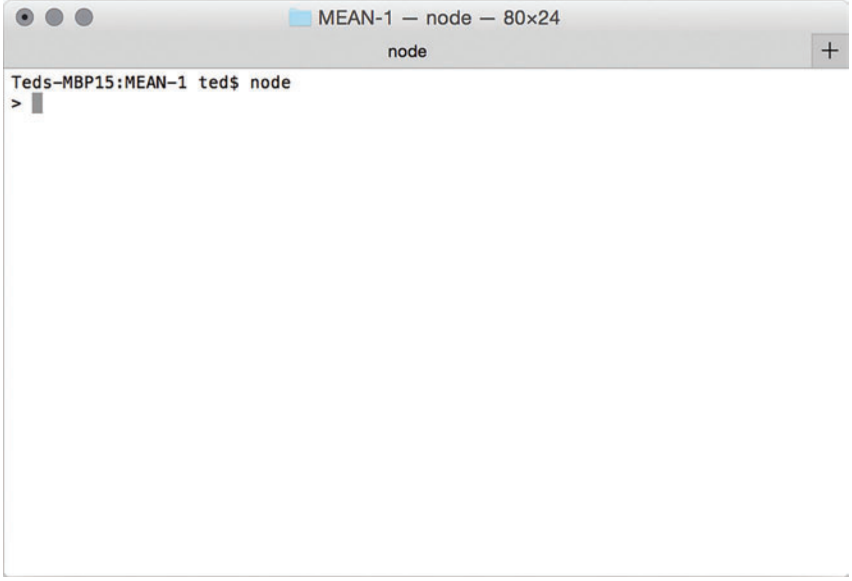
In the comics, whenever a hero steps into a new universe, there's a moment when the hero has to embrace the new world or risk being identified as the outsider—and possibly locked away in a sanitarium. Unless you fancy the sanitarium, you probably should embrace this new world.

Get MEAN

Let's talk about Node.js. Or, more precisely, let's talk about one of the favored software stacks for the Node.js platform: MEAN (MongoDB, Express, AngularJS, Node.js). It's quickly becoming one of the key players in the new technology world. Microsoft's support for Node.js and MongoDB on Azure (not to mention the fact you can easily run these on a standard Windows machine on-premises) means—if you'll pardon the pun—it's something every Microsoft developer should know.

You can do MEAN a couple of ways, including using Visual Studio. Start off by "embracing the Node Way." Use just the preferred set of Node-ish tools: a text editor, the command line and (if you have one lying around) a Mac. That's what I'll be using to start, although most of the command-line commands will be pretty adjustable to Windows without difficulty, it will be different.

Before I get too far into this parallel universe, take a look at the major players. The MEAN stack is a "full stack" quartet, meaning it covers front end, back end and storage. Starting from the front end, AngularJS provides a complete Single-Page Application Web client framework, complete with Model-View-Controller abstractions and two-way binding for the UI. AngularJS resides entirely on the front end, though, and requires a back end with which to communicate. This typically uses Web API calls. These are also



```
Teds-MBP15:MEAN-1 ted$ node
> █
```

Figure 1 **Node.js Running; No, Seriously, It's Running**

AWSInsider.net

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

AWS Insider.net

THE INDEPENDENT RESOURCE FOR
AMAZON WEB SERVICES

Home
Newsletters
Whitepapers
Webcasts
Advertise
Contact Us

NEWS

Pivotal Adds One-Click AWS Integration

Pivotal Software added one-click integration with the Amazon Web Services cloud to its Cloud Foundry Platform-as-a-Service offering.

NEWS

Amazon Adds Unlimited Cloud Drive Plans

Amazon.com Inc. added two unlimited cloud storage plans to its Cloud Drive offering, upping the storage ante with competitors such as Microsoft and Google.

Most Popular Articles

- AWS Is Leader in Enterprise Cloud Migrations
- Google Challenges AWS with Nearline Storage
- Informal Survey Shows Azure Use Almost Equals AWS
- Oracle Linux Now Available on AWS Marketplace

AWS Insider.net

THE INDEPENDENT RESOURCE FOR
AMAZON WEB SERVICES

known in some circles as RESTful endpoints, although that can lead to zealous debates about what REST is, so let's leave it as Web APIs for now.

Those Web API endpoints are built with the Express framework sitting on top of the Node.js platform. This is much the same way ASP.NET Web API sits on top of the ASP.NET pipeline and the Microsoft .NET Framework. These back-end Web APIs will clearly need a database for storage, which is where MongoDB comes into play. MongoDB is a schema-less, document-oriented data store (in contrast to SQL Server, which is a schemaed, relational-oriented data store) with some built-in sharding and map-reduce capabilities.

It may have already occurred to you these three parts—front end, back end and storage—are actually pretty interchangeable. For example, it wouldn't be hard to imagine using AngularJS to talk to ASP.NET Web APIs that in turn talk to MongoDB. Or to use a Windows Forms application as the front end, making HttpClient calls to Node.js that in turn talks to MongoDB. Or, just to round out the hat trick, AngularJS to Node.js to SQL Server.

Any of these three components is easily "swappable," so long as the front end uses HTTP (or something similarly platform-neutral) to talk to the back end. The back end also needs a driver to communicate to your choice of data store.

Install Node.js

Node.js is really a trivial beast to install, when you get right down to it. Developers who have the Azure SDK already installed have Node.js, and it's probably already on the PATH, to boot. Just type node in a command prompt to verify it's installed. Ctrl+C takes you out of the interactive shell that fires up if Node.js is installed, by the way.

You'll have to install Node.js on a fresh-from-the-factory Windows box, or a similarly new Mac OS box for that matter. On Windows, the best way to do this is either get the Azure SDK, or go to the Node.js Web site for an MSI installer for Node.js, which puts it on the PATH by default. There's also an installer for the Mac OS, but the better approach on the Mac is to install another package manager called Homebrew. It's available at brew.sh. Once installed, this becomes your "go-to tool" for installing anything on the Mac, including Node.js.

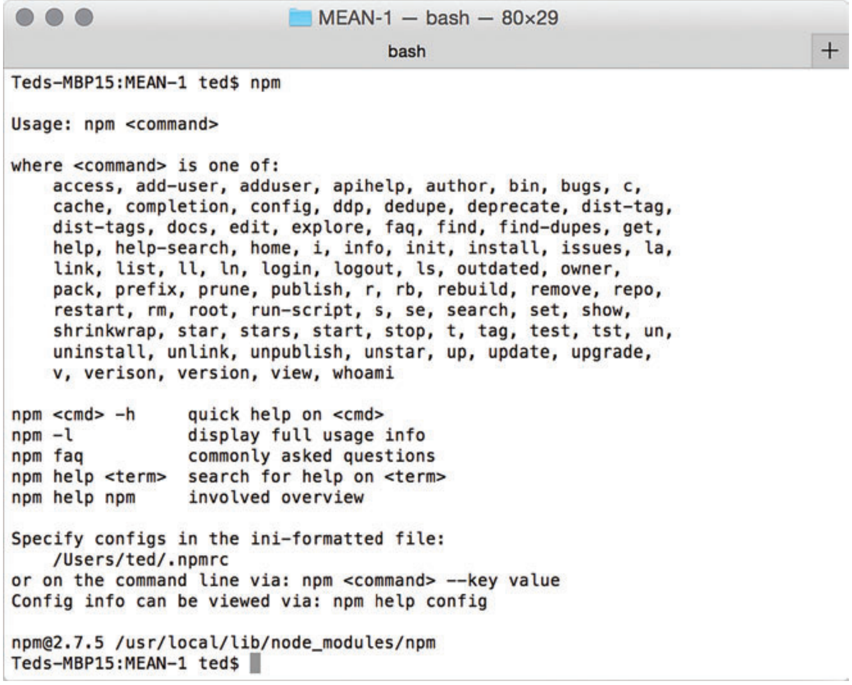
Homebrew has a simple "brew install node" that will pull down all the Node.js bits, install them to the right places (without requiring root access to do it), and put it implicitly on the PATH. Again, "node" at the command line will verify that the installation worked. When launched, it provides the lowest-noise response of any utility you'll ever install (as shown in **Figure 1**).

Hit Ctrl+C (twice) to get Node.js to quit. You're started, but you're not done.

Install the NPM

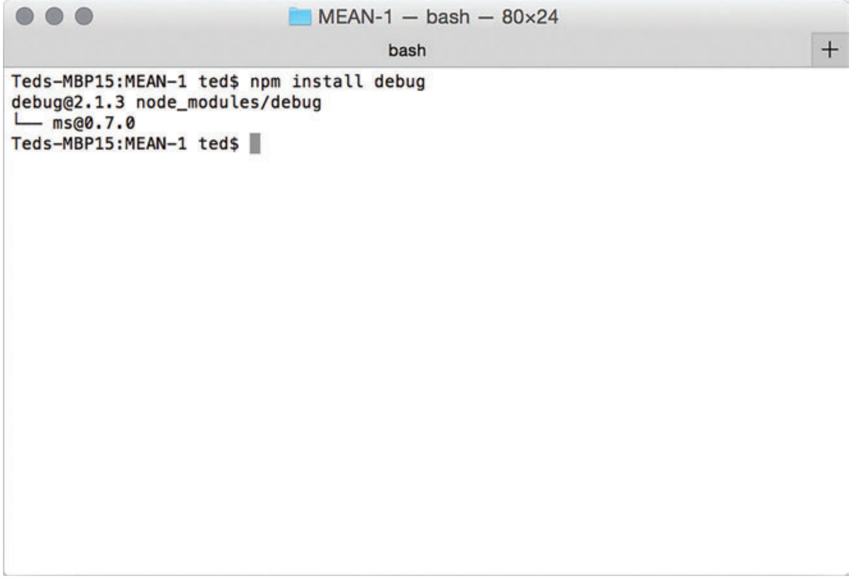
The .NET universe has NuGet. The Ruby universe has gems. The Node.js universe has the Node Package Manager. It's called the npm and it installs as part of Node.js itself (npm is actually a small Node.js app that runs from the command line). Without having to do anything else once Node.js is installed, you should be able to fire up npm using just "npm" at the command line, as shown in **Figure 2**.

The two commands you'll care most about are npm install and npm update. Each can take one or more npm package names as a parameter.



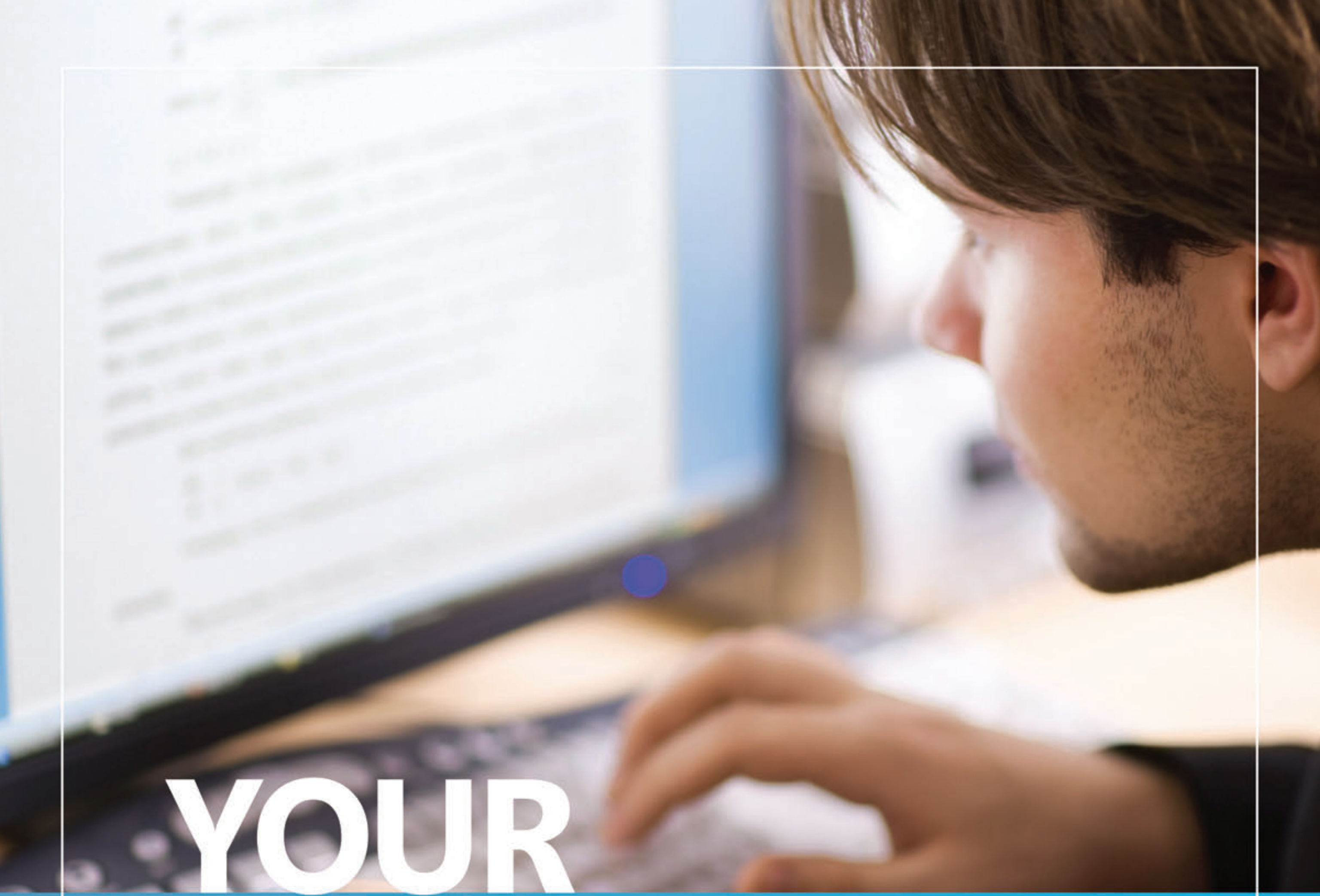
```
Teds-MBP15:MEAN-1 ted$ npm

Usage: npm <command>

where <command> is one of:
    access, add-user, adduser, apihelp, author, bin, bugs, c,
    cache, completion, config, ddp, dedupe, deprecate, dist-tag,
    dist-tags, docs, edit, explore, faq, find, find-dupes, get,
    help, help-search, home, i, info, init, install, issues, la,
    link, list, ll, ln, login, logout, ls, outdated, owner,
    pack, prefix, prune, publish, r, rb, rebuild, remove, repo,
    restart, rm, root, run-script, s, se, search, set, show,
    shrinkwrap, star, stars, start, stop, t, tag, test, tst, un,
    uninstall, unlink, unpublish, unstar, up, update, upgrade,
    v, verison, version, view, whoami

npm <cmd> -h     quick help on <cmd>
npm -l           display full usage info
npm faq          commonly asked questions
npm help <term>  search for help on <term>
npm help npm     involved overview

Specify configs in the ini-formatted file:
    /Users/ted/.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@2.7.5 /usr/local/lib/node_modules/npm
Teds-MBP15:MEAN-1 ted$ █
```

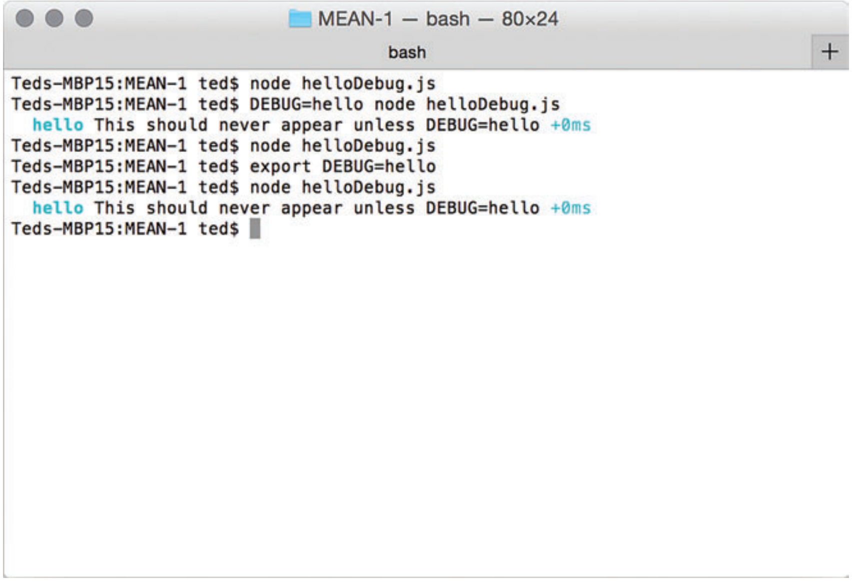
Figure 2 **The Node Package Manager**



```
Teds-MBP15:MEAN-1 ted$ npm install debug
debug@2.1.3 node_modules/debug
└── ms@0.7.0
Teds-MBP15:MEAN-1 ted$ █
```

Figure 3 **Install an npm Package**





```
Teds-MBP15:MEAN-1 ted$ node helloDebug.js
Teds-MBP15:MEAN-1 ted$ DEBUG=hello node helloDebug.js
  hello This should never appear unless DEBUG=hello +0ms
Teds-MBP15:MEAN-1 ted$ node helloDebug.js
Teds-MBP15:MEAN-1 ted$ export DEBUG=hello
Teds-MBP15:MEAN-1 ted$ node helloDebug.js
  hello This should never appear unless DEBUG=hello +0ms
Teds-MBP15:MEAN-1 ted$ 
```

Figure 4 **Hello, Debug World**

When you install a package, npm will download the package from the npm Web site, just as NuGet does. It then installs locally on the hard drive under the current directory. So, for example, in an empty directory, tell npm to install the debug package (as shown in **Figure 3**).

The npm tool responds by telling you it has downloaded version 2.1.3 of the debug package, which in turn depends on version 0.7.0 of the ms package. More important, both have been installed into a local directory called node_modules. This is the "local repository" of all Node.js packages you'll use for this application. If for some reason you want an npm package installed globally (to a node_modules directory somewhere in a shared location), you'll need to use npm install –g debug.

Once those packages are installed in the current directory, reference them using the "require" call in Node.js. It takes a string describing the package name, which Node expects to find inside the node_modules directory directly under the current directory. So the following code, in a file called helloDebug.js, loads the debug package, assigns it to a variable of the same name (the Node.js convention), and uses it to obtain a debug stream for emitting debug information (similar to System.Diagnostics.Trace):

```
var debug = require('debug')('hello');

debug("This should never appear unless DEBUG=hello");
```

Now when you run that code (node helloDebug.js), if there's an environment variable named DEBUG set to "hello," debug calls will print to the console. If not, nothing will appear. On a Mac or Unix system, you can temporarily set an environment variable for one run of Node.js by prefixing the assignment right in front of the node command (as shown in **Figure 4**).

It's not a lot, but it starts to give you a feel for how Node.js development works. Most important, you should realize a require call is looking to load a package out of the local node_modules directory. So if a require fails, it means the package was either corrupted locally or was never installed. In the next column, I'll talk about how to keep track of which npm packages are installed so you don't have to remember.

Did I forget to mention all of the Node.js code is JavaScript? If you're uncomfortable with JavaScript, now's a good time to brush up. Douglas Crockford's "JavaScript: The Good Parts" (O'Reilly Media, 2008) is a great place to start.

Install MongoDB

Getting MongoDB onto a local development system is indeed trivial. Download the .zip file appropriate for your system from the MongoDB Web site (mongodb.org), unzip it and put the binaries on your PATH. MongoDB is also available in several "as-a-Service" flavors, such as MongoLab (mongolab.com), which offers a forever-free tier for data loads less than half a gig (which is plenty for most introductory purposes). Either install MongoDB locally or create a MongoLab account.

The MongoDB download also has the "mongo" command-line client (similar in style and scope to the SQL Server command-line client). This is useful for accessing a MongoDB database from shell scripts and the like. If you're more GUI-centric, there are a few free MongoDB GUI tools in the world. My favorite for the Mac is RoboMongo, and for MongoVue for Windows.

By default, assume Mongo is running locally (meaning the server is "localhost" and the default port is 27017). If you're unfamiliar with Mongo, you can either check out my earlier MongoDB column at msdn.microsoft.com/magazine/ee310029, or spend a few minutes brushing up on any of the tens of thousands of articles on MongoDB online. Bing is your friend here.

Running MongoDB on a local machine is also trivial. Assume the MongoDB bin directory is on the PATH and just fire up "mongod." It will assume it can write to the "/var" directory to store data. That's usually not what you want, so pass a "--dbpath" argument (note the double hypen) to specify the directory of choice in which to store data.

The MongoDB Web site also has instructions on how to run MongoDB as a Windows service or as a daemon on *nix platforms. However, launching it "by hand" in the foreground has the added advantage of being able to see the MongoDB logs being written to screen as they happen.

Wrapping Up

That's all the space for this month, but you have the core bits you need to get started. Next time, I'll start spinning up some HTTP endpoints on the server, use Express, talk a little bit more about how a Node.js application is structured, and how to build those endpoints to run in Azure. I'll also start looking at some of the tools Node.js applications use as part of their development. But for now …

Happy coding! ■

Ted Neward *is the CTO at iTrellis, a consulting services company. He has written more than 100 articles and authored and co-authored a dozen books, including "Professional F# 2.0" (Wrox, 2010). He's an F# MVP and speaks at conferences around the world. He consults and mentors regularly—reach him at ted@tedneward.com or ted@itrellis.com if you're interested.*

MSDN Magazine Online

MSDN Magazine

Search MSDN Magazine with Bing

United States - English · Sign in

Home · Topics · Issues and Downloads · Subscribe · Submit an Article · RSS

THE MICROSOFT JOURNAL FOR DEVELOPERS · SEPTEMBER 2014 VOL 29 NO 9

msdn magazine

Soup to Nuts: From Raw Hardware to Cloud-Enabled Device

There is a new gold rush happening and it's not about precious metals. It's about building innovative, consumer-based devices and connecting them to the cloud. In this month's Azure Insider feature, the authors roll up their sleeves and explore what it takes to embrace this new computing paradigm.

Bruno Terkaly
Steven Edouard

Modern Apps Build Universal Apps for the Windows Platform

Windows platform developers can now write apps that target both Windows 8.1 and Windows Phone 8.1 with a single, shared codebase by creating a Universal App. Rachel Appel shows how you can get started.

Rachel Appel

Features

Microsoft Azure: Introduction to Machine Learning Studio

Machine learning systems use data to make predictions. The Azure Machine Learning Studio gives you a much easier way to create a machine learning model than writing code from scratch.

James McCaffrey

Columns

Cutting Edge

A Look at ClearScript

Add ClearScript into any .NET application to add scripting capabilities, publish objects and enable users to create and run their own scripts within the application.

Dino Esposito

MSDN Magazine Blog

MSDN Magazine July Issue Preview: All About Azure Web Sites

The July issue of MSDN Magazine is themed around the updated features and functionality of Microsoft Azure Web Sites, and reflects the

It's like ***MSDN Magazine***—only better. In addition to all the great articles from the print edition, you get:

- Code Downloads
- The *MSDN Magazine* Blog
- Digital Magazine Downloads
- Searchable Content

All of this and more at **msdn.microsoft.com/magazine**

msdn magazine

Visual Studio LIVE!

EXPERT SOLUTIONS FOR .NET DEVELOPERS

ORLANDO

ROYAL PACIFIC RESORT AT
UNIVERSAL ORLANDO

NOV 16-20

Code In The Sun

Fill-up on real-world, practical information and training on the Microsoft Platform as Visual Studio Live! returns to warm, sunny Orlando for the conference more developers rely on to code with industry experts and successfully navigate the .NET highway.

Connect with Visual Studio Live!

twitter.com/vslive
@VSLive

facebook.com
Search "VSLive"

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

EVENT PARTNERS: Microsoft, Magenic

PLATINUM SPONSORS: GRIDSTORE, pluralsight

GOLD SPONSOR: GitHub

SUPPORTED BY: Visual Studio, msdn magazine

5 Great Conferences
1 Great Price

Whether you are an

- Engineer
- Developer
- Programmer
- Software Architect
- Software Designer

You will walk away from this event having expanded your .NET skills and the ability to build better applications.

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

Use promo code ORLAUG4 by August 12

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

Visual Studio MAGAZINE

PRODUCED BY
1105 MEDIA

Take the Tour

LIVE! 360
TECH EVENTS WITH PERSPECTIVE

Visual Studio Live! Orlando is part of Live! 360, the Ultimate Education Destination. This means you'll have access to four (4) other co-located events at no additional cost:

SharePoint LIVE!
TRAINING FOR COLLABORATION

SQL Server LIVE!
SQL SERVER FOR MODERN DEVELOPERS

Modern Apps LIVE!
MOBILE, CROSS-DEVICE & CLOUD DEVELOPMENT

TECHMENTOR
IN-DEPTH TRAINING FOR IT PROS

Five (5) events and 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!

TURN THE PAGE FOR MORE EVENT DETAILS.

VSLIVE.COM/ORLANDO

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

ORLANDO
ROYAL PACIFIC RESORT AT
UNIVERSAL ORLANDO

NOV
16-20

Check Out the Additional Sessions for Devs, IT Pros, & DBAs at Live! 360

LIVE! 360

SharePoint LIVE!
TRAINING FOR COLLABORATION

SharePoint Live! features 12+ developer sessions, including:

- Workshop: Developer OnRamp to the Office 365 & Azure AD Highway! - Andrew Connell
- An Introduction to SharePoint 2013 Add-ins for Developers - Rob Windsor
- SharePoint Developers, Come to the Client Side (We Have Cookies) - Bill Ayers
- SharePoint Development Lifecycle for Solutions and Add-ins - Robert Bogue
- Building Office Add-Ins with Visual Studio - Bill Ayers
- Utilizing jQuery in SharePoint - Get More Done Faster - Mark Rackley
- Using Azure to Replace Server-Side Code in Office 365 - Paul Schaeflein

SQL Server LIVE!
SQL SERVER FOR MODERN DEVELOPERS

SQL Server Live! features 15+ developer sessions, including:

- Encrypting Data in SQL Server - David Dye
- Kick Start! SQL Server 2014 Performance Tips and Tricks - Pinal Dave
- Database Development with SQL Server Data Tools - Leonard Lobel
- Exploring T-SQL Enhancements: Windowing and More - Leonard Lobel
- Python and R for SQL and Business Intelligence Professionals - Jen Stirrup

TECHMENTOR
IN-DEPTH TRAINING FOR IT PROS

TechMentor features IT Pro and DBA sessions, including:

- Workshop: Windows PowerShell Scripting and Toolmaking (BYOL-RA) - Jeffery Hicks
- Your Approach to BYOD Sucks! - Simon May
- Windows Server vNext: Here's What We Know - Don Jones
- Ethical Hacking: Methodologies and Fundamentals - Mike Danseglio & Avril Salter
- DISCUSSION; Career Strategies for the IT Pro - Greg Shields
- PowerShell Unplugged: Stuff I Stole from Snover - Don Jones
- Workshop: SQL Server 2014 for Developers - Leonard Lobel

SEE THE FULL AGENDA AT LIVE360EVENTS.COM

Featured Live! 360 Speakers

ANDREW BRUST

MIGUEL CASTRO

ANDREW CONNELL

DON JONES

DEBORAH KURATA

ROCKFORD LHTOKA

MATTHEW MCDERMOTT

TED NEWARD

JOHN PAPA

BRIAN RANDELL

RACHEL REESE

GREG SHIELDS

AGENDAS AT-A-GLANCE VISUAL STUDIO LIVE! & MODERN APPS LIVE!

Cloud Computing	Database and Analytics	Lessons Learned and Advanced Practices	Mobile Client	Visual Studio / .NET Framework	Web Development	Windows Client	Modern Apps Live! Sponsored by: Magenic

START TIME	END TIME	Visual Studio Live! & Modern Apps Live! Pre-Conference: Sunday, November 15, 2015
6:00 PM	9:00 PM	Dine-A-Round Dinner @ Universal CityWalk - 6:00pm - Meet at Conference Registration Desk to walk over with the group

START TIME	END TIME	Visual Studio Live! & Modern Apps Live! Pre-Conference Workshops: Monday, November 16, 2015			
8:00 AM	5:00 PM	VSM01 Workshop: Service Oriented Technologies: Designing, Developing, & Implementing WCF and the Web API - *Miguel Castro*	VSM02 Workshop: Triple D: Design, Development, and DevOps - *Billy Hollis and Brian Randell*	VSM03 Workshop: Busy Developer's Guide to MEANJS - *Ted Neward*	MAM01 Workshop: Modern App Technology Overview - Android, iOS, Cloud, and Mobile Web - *Nick Landry, Kevin Ford, & Steve Hughes*
5:00 PM	6:00 PM	EXPO Preview			
6:00 PM	7:00 PM	**Live! 360 Keynote:** Microsoft 3.0: New Strategy, New Relevance - *Pacifica 6* *Mary Jo Foley, Journalist and Author; with Andrew Brust, Senior Director, Datameer*			

START TIME	END TIME	Visual Studio Live! & Modern Apps Live! Day 1: Tuesday, November 17, 2015				
8:00 AM	9:00 AM	**Visual Studio Live! Keynote:** To Be Announced - *Jay Schmelzer, Director of Program Management, Visual Studio Team, Microsoft*				
9:00 AM	9:30 AM	Networking Break • Visit the EXPO - *Pacifica 7*				
9:30 AM	10:45 AM	VST01 AngularJS 101 - *Deborah Kurata*	VST02 A Tour of Azure for Developers - *Adam Tuliper*	VST03 Busy Developer's Guide to NoSQL - *Ted Neward*	VST04 Visual Studio, TFS, and VSO in 2015 - What's New? -*Brian Randell*	MAT01 Defining Modern App Development - *Jason Bock*
11:00 AM	12:15 PM	VST05 From ASP.NET Site to Mobile App in About an Hour - *Ryan J. Salva*	VST06 Introduction to Next Generation of Azure PaaS – Service Fabric and Containers - *Vishwas Lele*	VST07 Real World SQL Server Data Tools (SSDT) - *Benjamin Day*	VST08 Automate Your Builds with Visual Studio Online or Team Foundation Server - *Tiago Pascoal*	MAT02 Modern App Architecture - *Brent Edwards*
12:15 PM	2:00 PM	Lunch • Visit the EXPO - *Oceana Ballroom / Pacifica 7*				
2:00 PM	3:15 PM	VST09 I Just Met You, and "This" is Crazy, But Here's My NaN, So Call(Me), Maybe? - *Rachel Appel*	VST10 Cloud or Not, 10 Reasons Why You Must Know "Websites" - *Vishwas Lele*	VST11 Windows 10 for Developers: What's New in Universal Apps - *Nick Landry*	VST12 Defensive Coding Techniques in C# - *Deborah Kurata*	MAT03 ALM with Visual Studio Online (TFS) and Git - *Brian Randell*
3:15 PM	4:15 PM	Networking Break • Visit the EXPO - *Pacifica 7*				
4:15 PM	5:30 PM	VST13 Better Unit Tests through Design Patterns for ASP MVC, WebAPI, and AngularJS - *Benjamin Day*	VST14 Running ASP.NET Cross Platform with Docker - *Adam Tuliper*	VST15 Build Your First Mobile App in 1 Hour with Microsoft App Studio - *Nick Landry*	VST16 Putting CodedUI Tests on Steroids - *Donovan Brown*	MAT04 Reusing Logic Across Platforms - *Kevin Ford*
5:30 PM	7:30 PM	Exhibitor Reception				

START TIME	END TIME	Visual Studio Live! & Modern Apps Live! Day 2: Wednesday, November 18, 2015				
8:00 AM	9:00 AM	**Live! 360 Keynote:** To Be Announced - *Pacifica 6 - Sponsored By:* pluralsight				
9:15 AM	10:30 AM	VSW01 Go Mobile with C#, Visual Studio, and Xamarin - *James Montemagno*	VSW02 Notify Your Millions of Users with Notification Hubs - *Matt Milner*	VSW03 Let's Write a Windows 10 App: A Basic Introduction to Universal Apps - *Billy Hollis*	VSW04 To Git or Not to Git for Enterprise Development - *Benjamin Day*	MAW01 Coding for Quality and Maintainability - *Jason Bock*
10:30 AM	11:00 AM	Networking Break • Visit the EXPO - Pacifica 7				
11:00 AM	12:15 PM	VSW05 Automated UI Testing for Android and iOS Mobile Apps - *James Montemagno*	VSW06 Busy Developer's Guide to the Clouds - *Ted Neward*	VSW07 Designing and Building UX for Finding and Visualizing Data in XAML Applications - *Billy Hollis*	VSW08 Anything C# Can Do, F# Can Do Better - *Rachel Appel & Rachel Reese*	MAW02 Start Thinking Like a Designer - *Anthony Handley*
12:15 PM	1:45 PM	Birds-of-a-Feather Lunch • Visit the EXPO - *Oceana Ballroom / Pacifica 7*				
1:45 PM	3:00 PM	VSW09 Stop Creating Forms In Triplicate - Use Xamarin Forms - *Matt Milner*	VSW10 To Be Announced	VSW11 Developing Awesome 3D Games with Unity and C# - *Adam Tuliper*	VSW12 Unit Testing Makes Me Faster: Convincing Your Boss, Your Co-Workers, and Yourself - *Jeremy Clark*	MAW03 Applied UX: iOS, Android, Windows - *Anthony Handley*
3:00 PM	4:00 PM	Networking Break • Visit the EXPO • Expo Raffle @ 3:30 p.m. - *Pacifica 7*				
4:00 PM	5:15 PM	VSW13 Mobile App Development with Xamarin and F# - *Rachel Reese*	VSW14 To Be Announced	VSW15 Recruiters: The Good, The Bad, & The Ugly - *Miguel Castro*	VSW16 DI Why? Getting a Grip on Dependency Injection - *Jeremy Clark*	MAW04 Leveraging Azure Services - *Kevin Ford*
8:00 PM	10:00 PM	Live! 360 Dessert Luau - *Wantilan Pavilion*				

START TIME	END TIME	Visual Studio Live! & Modern Apps Live! Day 3: Thursday, November 19, 2015				
8:00 AM	9:15 AM	VSH01 Getting Started with ASP.NET 5 - *Scott Allen*	VSH02 Lessons Learned: Being Agile in a Waterfall World - *Philip Japikse*	VSH03 Windows, NUI and You - *Brian Randell*	VSH04 Improving Performance in .NET Applications - *Jason Bock*	MAH01 Building for the Modern Web with JavaScript Applications - *Allen Conway*
9:30 AM	10:45 AM	VSH05 Build Data Driven Web Applications with ASP.NET MVC - *Rachel Appel*	VSH06 User Story Mapping - *Philip Japikse*	VSH07 Building Adaptive Uis for All Types of Windows - *Ben Dewey*	VSH08 Asynchronous Tips and Tricks - *Jason Bock*	MAH02 Building a Modern App with Xamarin - *Nick Landry*
11:00 AM	12:15 PM	VSH09 Automated Cross Browser Testing of Your Web Applications with Visual Studio CodedUI - *Marcel de Vries*	VSH10 Performance and Debugging with the Diagnostic Hub in Visual Studio - *Sasha Goldshtein*	VSH11 XAML Antipatterns - *Ben Dewey*	VSH12 Roslyn and .NET Code Gems - *Scott Allen*	MAH03 Building a Modern Cross-Platform App - *Brent Edwards*
12:15 PM	1:30 PM	Lunch on the Lanai - *Lanai / Pacifica 7*				
1:30 PM	2:45 PM	VSH13 Hate JavaScript? Try TypeScript. - *Ben Hoelting*	VSH14 Advanced Modern App Architecture Concepts - *Marcel de Vries*	VSH15 WPF MVVM In Depth - *Brian Noyes*	VSH16 Getting More Out of Visual Studio Online: Integration and Extensibility - *Tiago Pascoal*	MAH04 DevOps And Modern Applications - *Dan Nordquist*
3:00 PM	4:15 PM	VSH17 Grunt, Gulp, Yeoman and Other Tools for Modern Web Development - *Ben Hoelting*	VSH18 The Vector in Your CPU: Exploiting SIMD for Superscalar Performance - *Sasha Goldshtein*	VSH19 Building Maintainable and Extensible MVVM WPF Apps with Prism - *Brian Noyes*	VSH20 Readable Code - *John Papa*	MAH05 Analyzing Results with Power BI - *Steve Hughes*
4:30 PM	5:45 PM	Live! 360 Conference Wrap-Up - *Andrew Brust (Moderator), Andrew Connell, Don Jones, Rockford Lhotka, Matthew McDermott, Brian Randell, & Greg Shields*				

START TIME	END TIME	Visual Studio Live! & Modern Apps Live! Post-Conference Workshops: Friday, November 20, 2015		
8:00 AM	5:00 PM	VSF01 Workshop: Angular in 0 to 60 - *John Papa*	VSF02 Workshop: Native Mobile App Development for iOS, Android and Windows Using C# - *Marcel de Vries & Roy Cornelissen*	MAF01 Workshop: Modern App Development In-Depth - iOS, Android, Windows, and Web - *Brent Edwards, Anthony Handley, & Allen Conway*

Sessions and speakers subject to change.

Don't Get Me Started

DAVID S. PLATT

Why Can't We Be Friends?

I just returned from the Microsoft Ignite conference, held in Chicago the first week of May. Ignite is aimed at IT people more than developers. I don't see that side of the industry very often, which is why I wanted to explore it.

Ignite had some good ideas flying around. I especially admired the login by automatic facial recognition that's coming in Windows 10—finally, a security improvement that requires less user effort, not more. I also liked the continuing message, "It's the user that's mobile, not the app," as well as Harry Shum's adaptation of Arthur C. Clarke's statement when he said, "Any sufficiently advanced technology is invisible." I'll probably write a future column on each of these.

My biggest takeaway from Ignite was the burgeoning use of the term "DevOps," a clipped compound of "development" and "operations" that illustrates the blurring line between these disciplines. I'm seeing a lot of that at my clients today.

Ignite had some very good sessions on DevOps. The best I saw was the Channel 9 interview (bit.ly/1NzI5s1) with Brian Harry and Donovan Brown, in which they describe Microsoft's own experiences in bringing DevOps to Visual Studio Online.

> Developers need to start getting along with IT operations people, and vice versa.

The fusion of development and operations results from the pressure that drives our industry: the constantly rising standard of care, as software permeates more and more of human activity. Development cycles get shorter, as clients demand that important functionality reach them sooner. Customer demand for higher quality requires more and better testing, which means testing has to be automated. Deployment also has to be automated as it gets more frequent. Customers are demanding an integrated whole, instead of the piecemeal we've historically served them.

As a result, developers need to start getting along with IT operations people, and vice versa. Historically, this has been problematic. The devs cut corners to meet unrealistic schedules (what other kind is there?), push out buggy software, and leave the IT people to field calls from angry users who lose their work to crashes. The devs never have time to build good IT administration tools. (Two words: "MMC Snap-In.")

On the other hand, devs feel that IT people obstruct them at every turn by treating them like ordinary users. The devs have to work around Internet filters and password changes, automatic inactivity logouts and boring crap like software licenses, with which the IT people delight in complicating their lives.

This opposition stems from the two teams' historically different view of their own goals. IT people prize stability over all else, and the No. 1 cause of instability is change. Devs view their role as extending and enhancing the product, for which some instability is a small price to pay. Both camps say, very loudly, "It's hard enough to do our important work without fighting our supposed teammates."

It's all reminiscent of the feuding between the farmers and cowmen in "Oklahoma!" the musical. In the classic song, "The Farmer and the Cowman Should Be Friends," Aunt Eller has to pull a gun to get the two groups to stop fighting at a square dance (bit.ly/1H3C2sK).

Aunt Eller has a point. We're going to have to make it work, devs and IT together. It's historically inevitable. The ops part of the team has to accept continual change as part of their lives, because they don't have a product without it. The devs have to accept stability as one of their prime requirements, because customers value it highly.

With all that in mind, I've ~~shamelessly plagiarized~~ brilliantly re-used the code of that song:

Oh the devs and the IT guys should be friends
Oh the devs and the IT guys should be friends
One just bangs out code all day
The other has to make it play
But that's no reason why they can't be friends

Software folks should stick together,
Software folks should all be free
IT guys remove obstructions
Devs improve on MMC

As the original song ushered in the Golden Age of musical theater, I expect my version will usher in the Platinum Age of software. And you heard it here first, dear reader. Customers don't care who's in charge of what; they only care that they are, or aren't, getting what they need. DevOps. Make it go. ■

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.*

Get the Toolkit. Get the Team.

NEW RELEASE

Whether you're developing an external application for a client or maintaining your company's internal legacy software, ComponentOne Studio's controls will speed up your development time and allow you to get to coding the fun stuff.

Solve business problems with controls that meet your users' every need.

Grids & Data Management

Get every function you need from a spreadsheet—without any heavy lifting—using our datagrid controls.

Data Visualization

Present large data sets out-of-the-box with our beautiful, flexible charts and gauges.

Reporting & Documentation

Migrate reports from another platform, or generate your own with a full collection of controls.

Scheduling

25

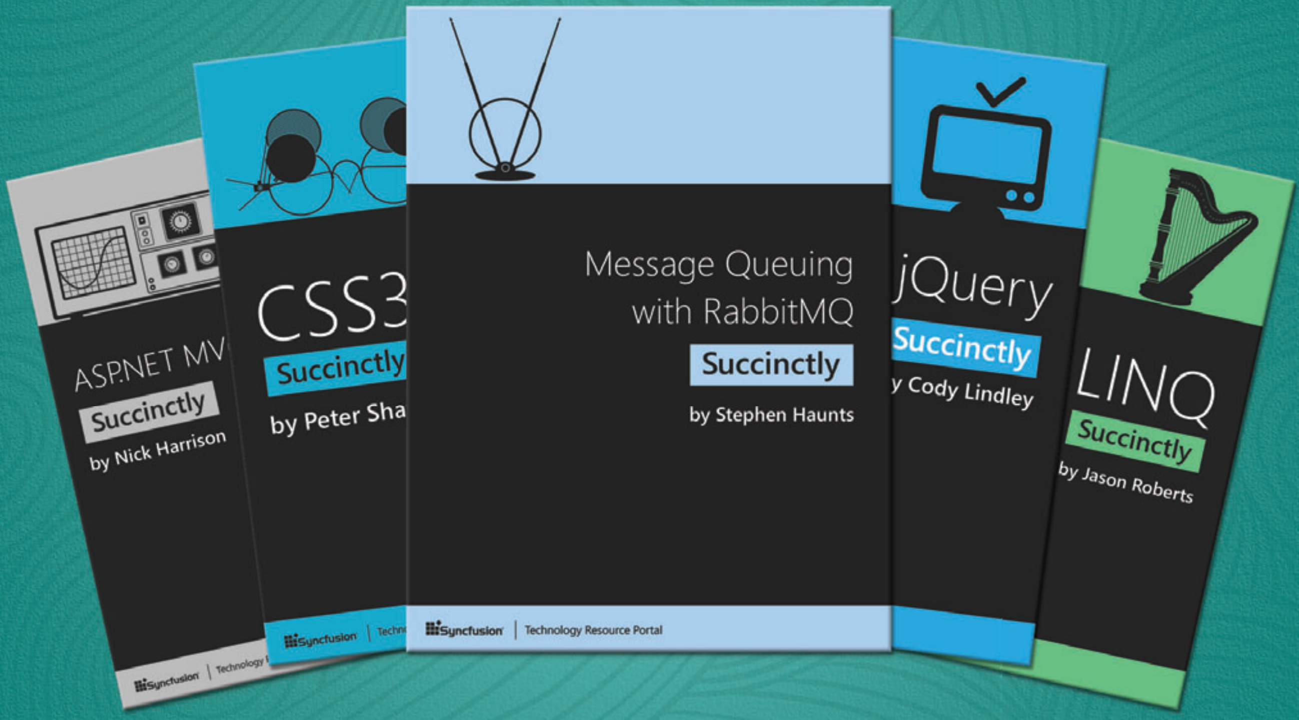
Offer instant Outlook-like functionality in any application.

WinForms | WPF | ASP.NET Web Forms | ASP.NET MVC | WinRT | LightSwitch | Silverlight | ActiveX

Download your free trial at
www.ComponentOne.com

Visual Studio Microsoft Partner

ComponentOne Studio


ASP.NET MV
Succinctly
by Nick Harrison
CSS3
Succinctly
by Peter Sha
Message Queuing
with RabbitMQ
Succinctly
by Stephen Haunts
Syncfusion | Technology Resource Portal
jQuery
Succinctly
y Cody Lindley
LINQ
Succinctly
by Jason Roberts

DOC
PPT
XLS
PDF
MSG

Aspose.Cells

Work with spreadsheets and data without depending on Microsoft Excel

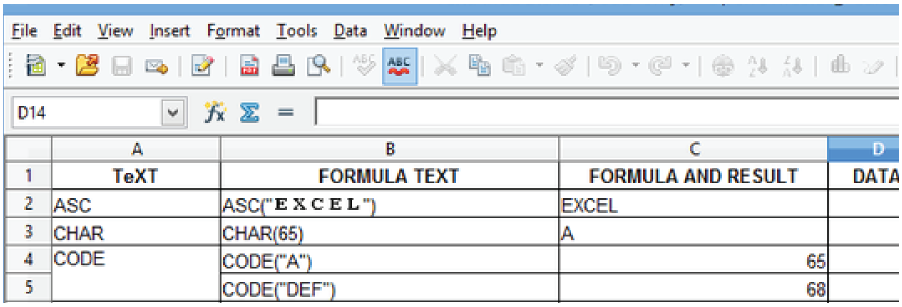
- Solution for spreadsheet creation, manipulation and conversion.
- Import and export data.

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.

A flexible API for simple and complex spreadsheet programming.

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



	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.

Aspose.Words

Program with word processing documents independently of Microsoft Word

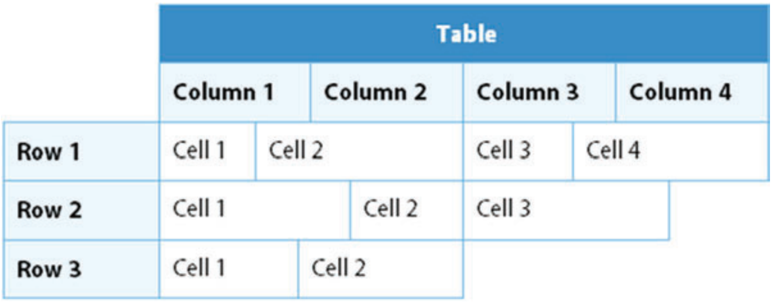
- Solution for document creation, manipulation and conversion.
- Advanced mail merge functionality.

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.

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

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



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.

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 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 member 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 exatly the same as the DOCX file.

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 are 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.

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 image, from background and bar color, through image quality, rotation angle, X-dimension, captions, and resolution.

Robust and reliable barcode generation and recognition.

Aspose.BarCode can read and recognize most common 1D and 2D barcodes from any image and at any angle. Filters help developers clean up difficult to read images to improve recognition.

Aspose.BarCode offers a large number of symbologies and formatting options.

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

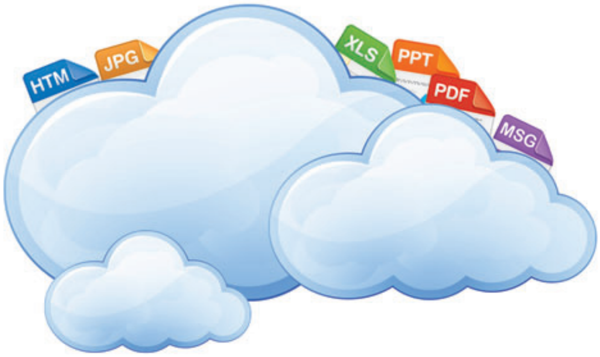
Linear: EAN13, EAN8, UPCA, UPCE, Interleaved2of5, Standard2of5, MSI, Code11, Codabar, EAN14(SCC14), SSCC18, ITF14, Matrix 2 of 5, PZN, Code128, Code39 Extended, Code39 Standard, OPC, Code93 Extended, Code93 Standard, IATA 2 of 5, GS1Code128, ISBN, ISMN, ISSN, ITF6, Pharmacode, DatabarOmniDirectional, 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.


HTM
JPG
XLS
PPT
PDF
MSG

Aspose.Email

Work with emails and calendars without Microsoft Outlook

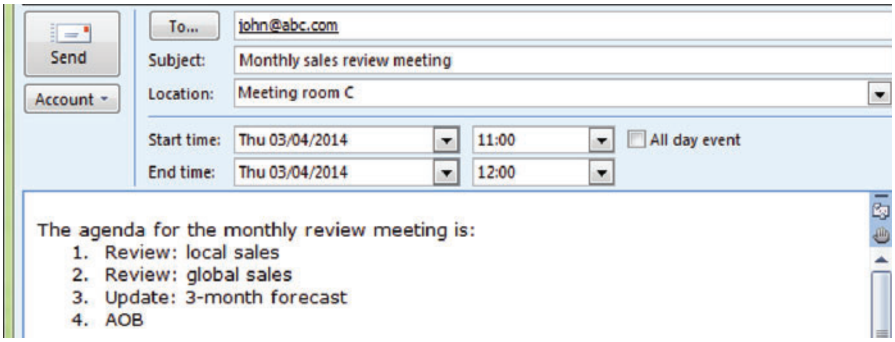
- Complete email processing solution.
- Message file format support.

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 works with HTML and plain text emails, attachments and embedded OLE objects.

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.



john@abc.com

Subject: Monthly sales review meeting

Location: Meeting room C

Start time: Thu 03/04/2014 11:00 □ All day event

End time: Thu 03/04/2014 12:00

The agenda for the monthly review meeting is:
1. Review: local sales
2. Review: global sales
3. Update: 3-month forecast
4. AOB

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.

Aspose.Pdf

Create PDF documents without using Adobe Acrobat

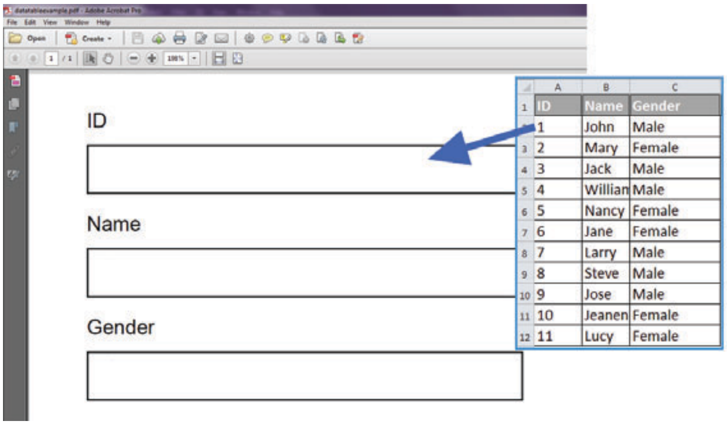
- A complete solution for programming with PDF files.
- Work with PDF forms and form fields.

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, extract or inset pages, and print PDF documents.



ID

Name

Gender

	A	B	C
1	ID	Name	Gender
2	1	John	Male
3	2	Mary	Female
4	3	Jack	Male
5	4	Willian	Male
6	5	Nancy	Female
7	6	Jane	Female
8	7	Larry	Male
9	8	Steve	Male
10	9	Jose	Male
11	10	Jeanen	Female
12	11	Lucy	Female

Aspose.Pdf can be used to automatically complete PDF forms with external data.

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.

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	
Microsoft OneNote 2010, 2010 SP1, 2013	Load, Save
PDF	Save
Images (BMP, GIF, JPG, PNG)	Save

Rendering and Printing
Save as Image (BMP, GIF, JPG, PNG)
Save as PDF

Document Management

- Extract text
- Get the number of pages in a document.
- Get page information.
- Extract images.
- Get image information from a document.
- Replace text in document.

Aspose.Imaging

Create Images from scratch.

- Load existing images for editing purposes.
- Render to multiple file formats.

Aspose.Imaging allows creation and manipulation of images.

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 draw on Image surface either by manipulating the bitmap information or by using the advanced functionality like Graphics and Paths.

Create images from scratch. or load existing ones...

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.

Aspose.Slides

Work with presentations without using Microsoft PowerPoint

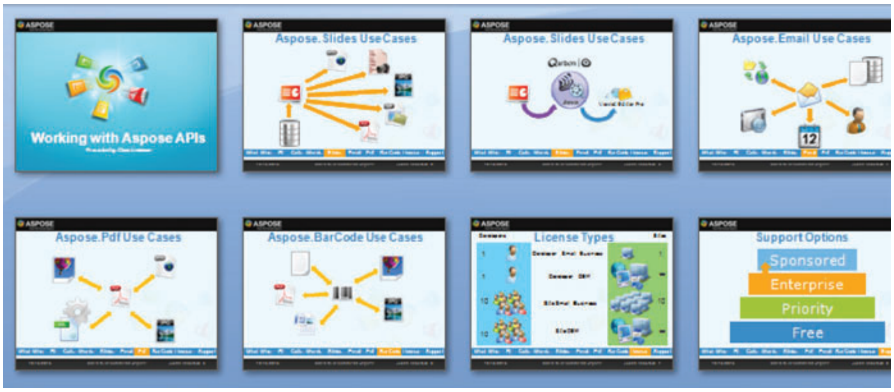
- Complete solution for working with presentation files.
- Export presentations and slides to portable or image formats.

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 you the tools you need to work with presentation files.

Aspose.Slides gives developers the tools they need to work with presentation files. It integrates quickly and saves time and money.



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.

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.

Aspose's file format experts are here to help you with a project or your support questions

When you need help to get a project off the ground, Aspose's developers can help.

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.

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

ASPOSE
Your File Format APIs