

msdn magazine



14 Top Features of Visual Basic 14.....16

When Only the Best Will Do

Our high-performance and feature-complete UI Controls and Libraries will help you build your best, without limits or compromise



Free 30-day Trial
devexpress.com/try

Unleash the **UI Superhero** in You

With DevExpress tools, you'll deliver amazing user-experiences for Windows®, the Web and Your Mobile World



Experience the DevExpress difference today.

Download your free 30-day trial.

devexpress.com/try

msdn magazine



**14 Top Features
of Visual Basic 14.....16**

Protect Your Online Services with Big Data
and Machine Learning
Alisson Sol, Don Ankney and Eugene Bobukh..... 34

Occasionally Connected Data
in Cross-Platform Mobile Apps
Kevin Ashley 42

BYOD Challenge: Connect Mobile Devices
to the Enterprise
Greg Oliver and Bruno Terkaly..... 48

Understanding TypeScript
Peter Vogel 52

Adding Application Data to the
Windows Phone 8.1 Map Control
Keith Pijanowski..... 58

Cross-Platform Cloud Automation
with JavaScript
Steven Edouard 68

VISUAL STUDIO 2015 & MICROSOFT AZURE

VISUAL BASIC 14
Lucian Wischik, page 16

**WEB-BASED TEST CASE
MANAGEMENT**
Manoj Bableswhar, page 22

HDINSIGHT AND HADOOP
Omid Afnan, page 28

COLUMNS

CUTTING EDGE
Mobilize an Existing Web Site
Dino Esposito, page 6

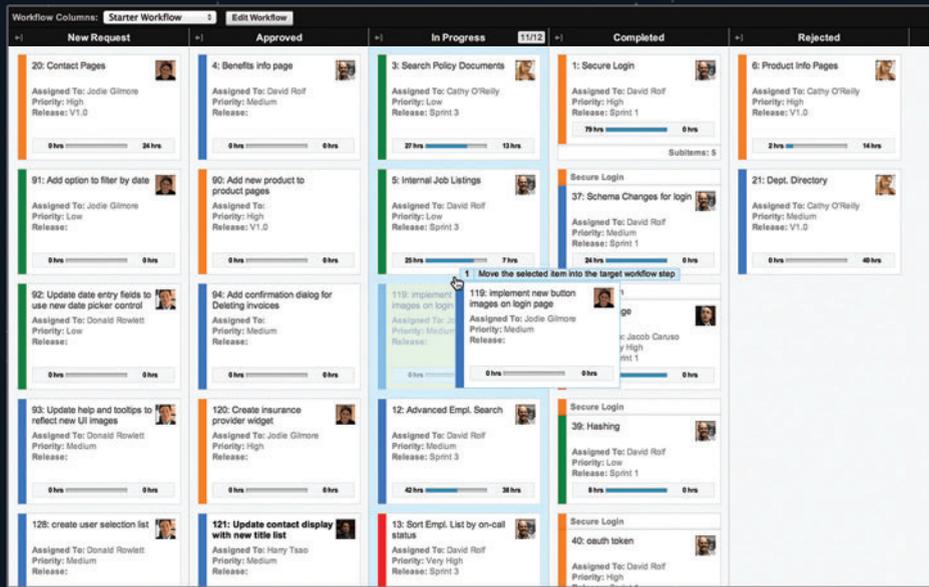
DATA POINTS
Looking Ahead to Entity
Framework 7
Julie Lerman, page 10

TEST RUN
Logistic Regression Classification
with Multi-Swarm Optimization
James McCaffrey, page 74

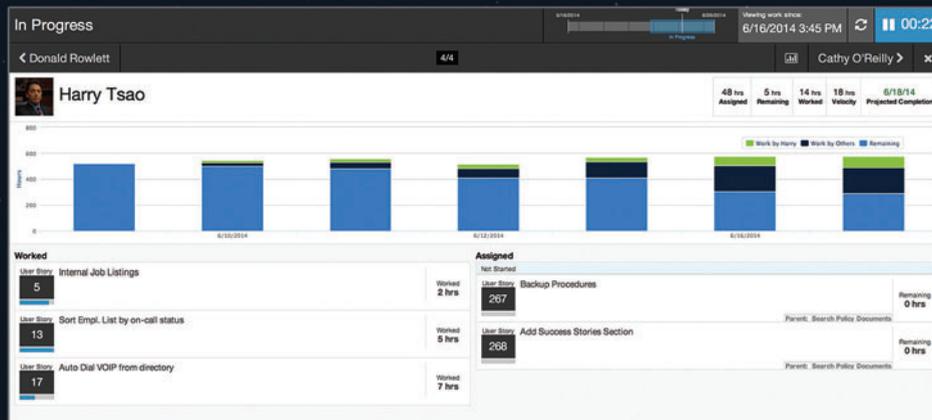
DON'T GET ME STARTED
An Upstart Again
David Platt, page 80

Don't be left in the dark when it comes to your team's development progress.

Axosoft empowers you and your team with instant visibility.



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



View team members' work capacity and assignments in our Daily Scrum mode

- Track user stories & defects
- Create custom process workflows
- Log work spent on each item
- Plan out releases and sprints
- Manage users, teams, & project roles
- Flexible notification system
- Help desk and self-service portal
- Powerful reporting tools

Integrates with popular tools:



slack



Visual Studio



zendesk

and more



Automated burndown charts, velocity, projected ship dates and more

Let us illuminate how we'll help you ship on time and on budget!

Try us free today

at axosoft.com/MSDN



The #1
Scrum Software

800.653.0024



dtSearch®

Instantly Search Terabytes of Text

25+ fielded and full-text search types

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

Supports databases as well as static and dynamic websites

Highlights hits in all of the above

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

"lightning fast" Redmond Magazine

"covers all data sources" eWeek

"results in less than a second" InfoWorld

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

dtSearch products: Web with Spider
 Desktop with Spider Engine for Win & .NET-SDK
 Network with Spider Engine for Linux-SDK
 Publish (portable media) Engine for Android-SDK **beta**
 Document Filters – included with all products, and also available for separate licensing

Ask about fully-functional evaluations

The Smart Choice for Text Retrieval® since 1991

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

msdn magazine

JANUARY 2015 VOLUME 30 NUMBER 1

KEITH BOYD Director

MOHAMMAD AL-SABT Editorial Director/mmeditor@microsoft.com

KENT SHARKEY Site Manager

MICHAEL DESMOND Editor in Chief/mmeditor@microsoft.com

LAFE LOW Features Editor

SHARON TERDEMAN Features Editor

DAVID RAMEL Technical Editor

WENDY HERNANDEZ Group Managing Editor

SCOTT SHULTZ Vice President, Art and Brand Design

JOSHUA GOULD Art Director

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

RedmondMediaGroup

Henry Allain President, Redmond Media Group

Becky Nagel Vice President, Digital Strategy

Michele Imgrund Vice President, Lead Services Division

Tracy Cook Director, Client Services & Webinar Production

Irene Fincher Director, Audience Development & Lead Generation Marketing

ADVERTISING SALES: 818-674-3416/dlbianca@1105media.com

Dan LaBianca Chief Revenue Officer

Chris Kourtoglou Regional Sales Manager

Danna Vedder Regional Sales Manager/Microsoft Account Manager

David Seymour Director, Print & Online Production

Anna Lyn Bayaua Production Coordinator/msdnadproduction@1105media.com

1105 MEDIA

Rajeev Kapur Chief Executive Officer

Henry Allain Chief Operating Officer

Richard Vitale Senior Vice President & Chief Financial Officer

Michael J. Valenti Executive Vice President

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

Jeffrey S. Klein Chairman of the Board

MSDN Magazine (ISSN 1528-4859) is published 13 times a year, monthly with a special issue in December by 1105 Media, Inc., 9201 Oakdale Avenue, Ste. 101, Chatsworth, CA 91311. Periodicals postage paid at Chatsworth, CA 91311-9998, and at additional mailing offices. Annual subscription rates payable in US funds are: U.S. \$35.00, International \$60.00. Annual digital subscription rates payable in U.S. funds are: U.S. \$25.00, International \$25.00. Single copies/back issues: U.S. \$10, all others \$12. Send orders with payment to: *MSDN Magazine*, P.O. Box 3167, Carol Stream, IL 60132, email MSDNmag@1105service.com or call (847) 763-9560. **POSTMASTER:** Send address changes to *MSDN Magazine*, P.O. Box 2166, Skokie, IL 60076. Canada Publications Mail Agreement No. 40612608. Return Undeliverable Canadian Addresses to Circulation Dept. or XPO Returns: P.O. Box 201, Richmond Hill, ON L4B 4R5, Canada.

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

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

Corporate Address: 1105Media, Inc., 9201 Oakdale Ave., Ste. 101, Chatsworth, CA 91311, www.1105media.com

Media Kits: Direct your Media Kit requests to Matt Morollo, VP Publishing, 508-532-1418 (phone), 508-875-6622 (fax), mmorollo@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

All customer service inquiries should be sent to MSDNmag@1105service.com or call 847-763-9560.



Printed in the USA



Version 19 has been crafted with developers in mind. It has been optimized to give you an all-in-one development solution for all your document, medical and multimedia needs.



Document
Imaging SDKs

Document Viewer ^{New}

Document Converter ^{New}

OCR, MICR, OMR & ICR

Barcode & Forms Recognition

Create, Load, Save, Edit & View PDF

Annotations, TWAIN & SVG ^{New}



Medical
Imaging SDKs

DICOM, CCOW & HL7 ^{New}

PACS Client & Server Framework

DICOM Storage Server Framework

Web & Desktop Viewers

Image Processing & Annotations

Medical 3D (MPR, MIP, VRT)



Multimedia
Imaging SDKs

Playback, Capture & Convert

Streaming - MPEG2-TS & RTSP

MPEG-4, H.264 & H.265 ^{New}

DVD & DVR

Media Foundation & DirectShow

Distributed Transcoding

.NET Windows API WinRT Linux iOS OS X Android HTML5





Back to Basic

When Microsoft convened its Connect() event in New York City in November, the company released so many vital development technologies—including open sourcing the .NET Core Runtime and libraries—that it prompted us to publish an extra issue of *MSDN Magazine* covering some of the new updates. Included in that coverage were looks at updated cross-platform tooling in Visual Studio, a dive into ASP.NET 5 and the Azure SDK 2.5, and, of course, a run through the newest versions of C# and C++.

What we weren't able to include in that special issue, but do feature in our pages this month, is a trifecta of articles focused on developments at Connect(). These include Lucian Wischik's "14 Top Improvements to Visual Basic 14," Omid Afnan's "Hadoop Made Easier for Microsoft Developers," and Manoj Bableshtar's "Web-Based Test Case Management with TFS."

Ultimately, the most important improvement in this latest version of Visual Basic may be the ironclad commitment Microsoft has shown for it.

I was particularly interested in Wischik's article, given that Microsoft essentially rebuilt Visual Basic from the ground up for the version 14 release. Wischik, a member of the Visual Basic/C# Language Design Team at Microsoft, told me that his team spent more than four years rewriting the programming language in Visual Basic (it was written in C++ before). He described the effort as both an opportunity and a challenge. Powerful new functionality, such as native refactoring and support for analyzers, are

enabled by "Roslyn" compiler technology that would be impossible to access without the rewrite. And Wischik says the wholesale update offered an opportunity to rearchitect the language from scratch based on what the group had learned over the years.

But the risks involved were not insignificant, he says. "You're investing a lot of developer years just to regain the functionality you already had, even before you start to add any new end-user value. There's also the risk that you might fail to reproduce all of the quirks or bugs of the old code base that people might inadvertently depend upon."

Complicating matters is the surprising complexity of Visual Basic—a function of the extensive array of features that the language, with its cultural roots in Classic Visual Basic, had accrued over the years. As Wischik observes, the team faced the challenging task of both replicating and modernizing the developer experience in Visual Basic.

Looking forward, Wischik expects more IDE improvements to arrive after Visual Studio 2015 ships. "We've been working on a Read Eval Print Loop (REPL), a sort of souped-up immediate window for quick experimenting and prototyping. We also want to write more analyzers and other plug-ins," he says.

Wischik urges Visual Basic developers to migrate to Visual Studio 2015, because it provides a host of benefits—including faster compile times and support for analyzers—while allowing round tripping of code back to Visual Studio 2013. From there, developers can make the decision to move to Visual Basic 14, which boasts new features like string interpolation, nameof and the ?. operator.

Ultimately, the most important improvement in this latest version of Visual Basic may be the ironclad commitment Microsoft has shown for it.

"It's the strongest vote of confidence a business can make," Wischik says. "We believe our product has such a strong future in the decades to come, that we can invest all our developer years now to prepare for that future."

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

© 2015 Microsoft Corporation. All rights reserved.

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

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

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



Some things are just better together.

Make your Windows Azure environment
more appetizing with New Relic.



Get more visibility into your entire Windows Azure environment.
Monitor all your Virtual Machines, Mobile Services, and Web Sites
- all from one powerful dashboard.

newrelic.com/azure



Mobilize an Existing Web Site

It's a common scenario. You arrive at a hotel, airport or some other public space. You connect to the local wireless network and you're redirected to a login page to enter your credentials. The login page on your mobile device is tiny. All you can do is pinch-and-zoom the screen, take the input form to a reasonable size and then type in your credentials.

This is an annoying experience, yet it happens more often than not. And it's more than simply an annoyance. The future of Web sites that don't render well on all devices is bleak. Not offering mobile users a good experience can cut into business profits by driving users away from the site and toward a competitor's site.

The simplest approach that doesn't require a full-site rewrite and won't break any existing features is to create a separate site specific for your mobile audience.

The magic wand too many developers wave to cast Web spells is responsive Web design (RWD). RWD is definitely an option, but, like anything else, it has its pros and cons. The problem with RWD is—for the most part—it requires a new project.

The basic idea of RWD is everything is fluid. Any elements will never be wider than the screen. So by definition, RWD adapts to an infinite number of screen widths. The point of RWD is to create a new Web site inspired by a new idea of content and targeting a fixed and smaller number of break points. Pushing RWD or a library like Twitter Bootstrap onto an existing Web site designed for the desktop is hardly a viable option.

Furthermore, selling the RWD approach is easy. There's one site, one brand and one code base serving multiple UIs. Implementing RWD is less easy, as it changes the Web development experience. You need a clear and strong vision of how the site will present information. RWD may be easy to sell, but it doesn't support improvisation. Content adaptation isn't magic. It comes from attentive and diligent planning.

RWD also changes the consolidated rules of project management and commonly ends up costing more than developing a plain non-responsive site. Gantt charts tracking progress make little sense as any new step requires team members to measure effects across the full spectrum of supported views. So in the end, when having one responsive Web site isn't an option, what can you do to meet your customers' expectations and make their life easier?

Add a Connected Mobile Site

When your Web site helps your company stay in business and plays a central role in attracting, maintaining and serving customers, then providing a mobile-specific experience is vital to keep those customers coming back. The simplest approach that doesn't require a full-site rewrite and won't break any existing features is to create a separate site specific for your mobile audience. This approach is referred to as building an m-site.

An m-site, much like a native application, is most likely to be consumed by users on the move—people who are standing up, in a hurry, busy or waiting in line. Users willing to connect to a site under these conditions and using a tiny device truly expect some benefit from the site. Therefore, the site must be direct, concise, accurate, and fast to load and respond.

Taking the m-site approach can be inexpensive for two reasons. It's a brand-new project, but it's expected to inherit core services from the existing site. More often than not, it inherits large shares of the business logic, which doesn't mean it duplicates code base. Also, an m-site usually implements the 80/20 rule. It typically supports no more than 20 percent of the features available in the full site. The selection of the 20 percent features, though, isn't a walk in the park.

When planning an m-site, you should reason with the mindset of a site user on the go. Which pieces of information does a user on the go really expect to consume on your site? Which functions

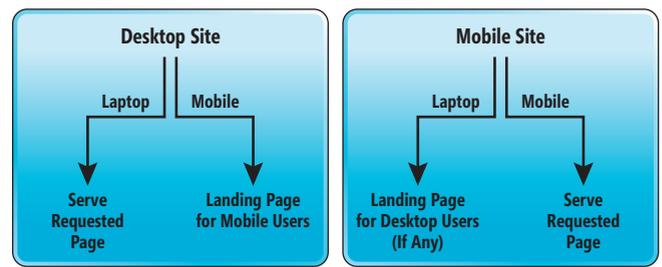
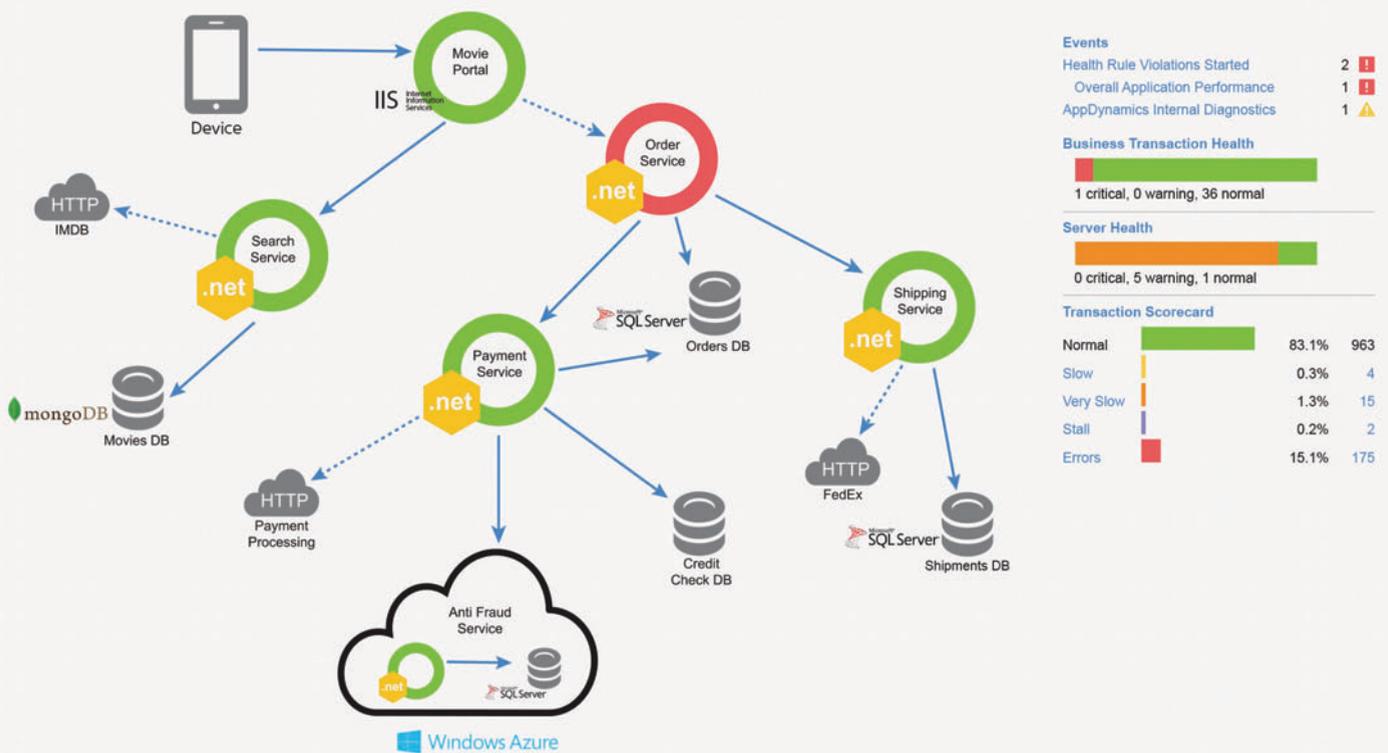


Figure 1 The Desktop/Mobile View Switcher Algorithm

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

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



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

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

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

does the user expect to perform? What can you do to smooth and speed up interaction as much as possible? This latter point involves using HTML5 features to the fullest.

On a site you know will be used on mobile devices, it's arduous to not set the type attribute of an input element to a number or date when numbers or dates are expected. This would greatly simplify typing. By the same token, there may be new use cases introduced on top of similar business logic to make life easier and more enjoyable to mobile users.

Accessing Your M-Site

Because an m-site is a discrete Web site, which URL should visitors use? Is a subdomain like `m.yourserver.com` the best option? Would a `.mobi` top-level domain name or `/mobile` subdirectory under the desktop work better? An m. subdomain is easier to manage than

a `.mobi` domain simply because it's a subdomain and not a new domain name. A directory is probably more error-prone as it means the mobile site will be hosted under the same application as the desktop. This can make the whole solution more brittle.

The simplest approach is to use an m. subdomain with some redirection logic that automatically sends mobile users to the `m.yourserver.com` site. This solution is good for users because they can use the direct m. subdomain and the canonical URL. For developers, though, it has the drawback of requiring device detection logic on both the desktop and the mobile site.

Implement Site Redirection

When two sites are in place and work under the same URL, you should process the host name and decide what to do for every request. If the host name belongs to the desktop site and the requesting browser is detected to be a desktop browser, then everything works as expected.

Otherwise, the user should see a landing page, and be informed they're trying to access a desktop site with a mobile device. Then they'll have a chance to save their preference for future similar situations. That preference is stored to a cookie and checked next. If the request refers to a URL in the mobile site and the user seems to have a desktop browser, consider showing a different landing page or just let the request go as usual. Finally, if a request is placed from a mobile device to the mobile site, it will be served as expected. You can have your site look into the device capabilities and determine the most appropriate view. **Figure 1** presents a diagram of the algorithm.

In ASP.NET, you can implement the routing algorithm as an HTTP module and register it on both the desktop and the mobile site. The module captures the `BeginRequest` event and uses plain redirection or—if possible—rewriting the URL to change the target page as appropriate, as shown in **Figure 2**.

Once installed in the desktop site, this HTTP module captures every request and checks the requesting browser. If the browser is running within a mobile device, the module redirects to the specified landing page. The landing page will be a mobile-optimized page offering a couple of links to the desktop site and the mobile site.

Where do you place the landing page to let mobile users decide what to do? Generally, it doesn't matter. However, if you put it on the mobile site, you can deploy your mobile site with all the required routing logic without touching the desktop site code base. Changes to the desktop site are limited to configuring the HTTP module in the `web.config` file.

What Does Mobile Really Mean?

The world of the mobile Web isn't simply split into two camps of desktop computers and all the rest. Mobile encompasses smartphones and tablets at the very least. An m-site that targets smartphones may not work as effectively for tablets and vice versa. Serving smartphone-optimized pages to tablets goes against the common user's expectation of getting a desktop-like experience.

Should you have an additional t-site for tablets and a site for legacy devices and perhaps one for wearable devices? All in all, smartphones deserve an ad hoc experience and a dedicated site. Tablets may blissfully render the same content as desktop computers with minor adjustments, such as a different set of CSS files with more padding and larger fonts and some media query support

Figure 2 Implementation of a Mobile Router HTTP Module

```
public class MobileRouter : IHttpModule
{
    private const String FullSiteModeCookie = "FullSiteMode";
    public void Dispose()
    {
    }
    public void Init(HttpApplication context)
    {
        context.BeginRequest += OnBeginRequest;
    }

    private static void OnBeginRequest(Object sender, EventArgs e)
    {
        var app = sender as HttpApplication;
        if (app == null)
            throw new ArgumentNullException("sender");

        var isMobileDevice = app.Context.Request.IsMobileDevice();

        // Mobile on desktop site, but FULL-SITE flag on the query string
        if (isMobileDevice && HasFullSiteFlag(app))
        {
            app.Response.AppendCookie(new HttpCookie(FullSiteModeCookie));
            return;
        }

        // Mobile on desktop site, but FULL-SITE cookie
        if (isMobileDevice && HasFullSiteCookie(app))
            return;

        // Mobile on desktop site => landing page
        if (isMobileDevice)
            ToMobileLandingPage(app);
    }

    private static Boolean HasFullSiteFlag(HttpApplication app)
    {
        var fullSiteFlag = app.Context.Request.QueryString["m"];
        if (!String.IsNullOrEmpty(fullSiteFlag))
            return String.Equals(fullSiteFlag, "f");
        return false;
    }

    private static Boolean HasFullSiteCookie(HttpApplication app)
    {
        var cookie = app.Context.Request.Cookies[FullSiteModeCookie];
        return cookie != null;
    }

    private static void ToMobileLandingPage(HttpApplication app)
    {
        var landingPage =
            ConfigurationManager.AppSettings["MobileLandingPage"];
        if (!String.IsNullOrEmpty(landingPage))
            app.Context.Response.Redirect(landingPage);
    }
}
```

for when the orientation switches to portrait or landscape. These adjustments require changes to the existing desktop site code base. However, it's not too much of an intrusive change.

The problem remains of how to detect smartphones and what's a definition of a smartphone that works for your business scenario.

Device Detection Tools

There's no common definition for a smartphone. A smartphone isn't a physical attribute like an OS version number. When browsers request a page, they send the user agent string, meaning some server-side code must process the user agent string and determine whether the device is a tablet, a smartphone or something else.

Device detection requires ad hoc tools, which have varying levels of reliability and accuracy. For example, you can use some Modernizr plug-ins or common routines to check user agents for known substrings. This may work reliably enough if all you want to do is separate desktops from everything else.

In this context, detecting anything like an Android OS tells the browser it's not a desktop. However, detecting Android doesn't help determine if it's a smartphone or tablet. The WURFL framework (wurfl.sourceforge.net) is a popular choice with tools that work on-premises, in the cloud and even the Web server level (no code, just configuration) on IIS and other environments.

If you opt for building an m-site to serve ad hoc content to mobile users, you need to be clear which devices fall under the jurisdiction of the m-site (choosing among smartphones, tablets and more) and how you detect devices so users have a single URL to reach the site regardless of the device.

Wrapping Up

Users expect to seamlessly use Web pages on their smartphones and tablets just as they do on their desktops. Processing power or cache size is limited on mobile devices, but that can't be an excuse to serve pages that require zooming and take seconds to load. Separating desktop from anything else isn't an insightful choice, either. Some level of device detection is required to provide a good service. When it comes to device detection and form factors, complexity and development costs will grow.

Many companies think RWD is the classic Columbus Egg—an obviously simple and brilliant solution to a thorny problem. Responsive Web sites are more complex and expensive to create than commonly reckoned. More important, an RWD approach works for new or significantly reworked sites. You can't just easily add RWD to mobilize an existing site that was devised for the desktop. Nevertheless, addressing the needs of your mobile audience is a business necessity. And the sooner you address it, the better. ■

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, Esposito frequently speaks at industry events worldwide and 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

msdnmagazine.com

MSDN Magazine Online



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



Looking Ahead to Entity Framework 7

Development of the next version of Entity Framework is well underway. I got my first glimpse of what the EF team was working on at TechEd North America 2014, when Program Manager Rowan Miller talked about the goals for Entity Framework 7 (EF7) and demonstrated some very early bits.

That was five months ago as I'm writing this column, and although EF7 is still an early alpha, it has already come a long way. In this column I want to make you aware of what EF7 will bring to developers, the motivations behind decisions being made about EF7, and what this version means to existing apps that use EF6 or earlier. I'll also give you a peek at what some of the code will look like.

Open Source, but Now on GitHub

The first thing to know about EF7 is that, like EF6, it's open source. But rather than being developed on CodePlex, EF7 is on GitHub, along with the rest of the upcoming version of ASP.NET. The URL for EF7 development is github.com/aspnet/EntityFramework. As with EF6, you'll be able to see the details of EF7 as it evolves. You can explore the source, as well as its progress through branches and commits, follow the discussions, raise issues, fork the source, and submit pull requests for the team to examine and potentially commit to the code base.

EF6 Is Not Going Away Any Time Soon

Don't worry—you won't be forced to move to EF7. Think back to ADO.NET DataSets and DataReaders. Much like ASP.NET Web Forms, which are still supported and even benefiting from occasional tweaks, ADO.NET is still part of the Microsoft .NET Framework, even though EF has been the primary data access technology for .NET for many years. Not much has happened to enhance those technologies, but they're still there, and still supporting a lot of legacy code (mine included). One of the big advantages EF6 has over those other technologies is that it's open source, so even though the team at Microsoft won't be making big investments in EF6, the community will still be able to. And the EF team is committed to EF6. They will continue to make tweaks, closely inspect pull requests and update EF6. Even though they've been going full-bore on EF7 for a good part of 2014, EF6 was updated. Version 6.1.0 was released in February 2014; 6.1.1 in June 2014; and as I'm writing this article, 6.1.2

is in beta, to be released soon. I was originally concerned, but am no longer worried, about being able to keep older apps going. The only ones I worry about are the earliest apps that used EF with the .NET Framework 3.5,ObjectContext and more. But if you haven't updated those apps to leverage all of the great improvements to EF over the years, you may not be worrying too much about EF7, anyway. You can also find all of the earlier packages for EF on NuGet, all the way back to EF 4.1.10311.

EF7: The Short List

Here's the high-level view of what's exciting in EF7:

- Support for non-relational data stores and even in-memory data for testing.
- Support for machines and devices that don't use the full .NET Framework. This means you can use EF7 in Windows Phone and Windows Store apps, as well as on Linux and Macintosh machines that are running Mono.
- Support for many features that developers have requested but couldn't be achieved with the existing code base.
- Continued support for applications that use the full .NET Framework such as Windows Presentation Foundation and other client applications.
- EF7 will be distributed in the same manner as ASP.NET 5 and can be used with ASP.NET 5 apps.

Familiar Coding Surface, New Code Base

Each version of EF has evolved the framework, adding new capabilities and fine-tuning performance and APIs. As I've written about before in this column, as well as in an overview article in the December 2013 issue, "Entity Framework 6: The Ninja Edition" (bit.ly/1cwjyCD), the latest version brought EF to a new level, giving the framework many of the features users have been asking for along the way, such as asynchronous database execution, tapping into the query pipeline, customizing Code First conventions and so much more. I drill much further into these features in my Pluralsight course, "Entity Framework 6, Ninja Edition: What's New in EF6" (bit.ly/PS-EF6).

There were even more features developers sought for EF that Microsoft was eager to implement, but the 10-plus-year-old code base EF is built on—with a continued dependency on ObjectContext and less flexible coding patterns—prevented the team from getting to this next level of capabilities. A difficult decision—which surely many of you have faced with your own legacy software—was made to rebuild Entity Framework from scratch.

This is based on an alpha version of Entity Framework 7. All information is subject to change.



Your Document Collaboration APIs

Professional APIs that allow developers to empower their apps with document collaboration capabilities.



GroupDocs.Viewer

Native-text, high-fidelity HTML5 document viewer with support for over 50 file formats.



GroupDocs.Signature

Electronic signature API that allows users to place signatures on documents within your apps.



GroupDocs.Annotation

A powerful API that lets developers annotate Microsoft Office, PDF and other documents within their apps.



GroupDocs.Comparison

A diff view API that allows end users to quickly find differences between two revisions of a document.



GroupDocs.Conversion

Universal document converter for fast conversion between more than 50 file formats.



GroupDocs.Assembly

Merges data entered by users through online forms into both Microsoft Office and PDF documents.

100% Standalone - No Office Automation



.NET Libraries



Java Libraries



Cloud APIs



Cloud Apps

SALES INQUIRIES

+1 214 329 9760

✉ sales@groupdocs.com

🏠 groupdocs.com

EF7 isn't creating a new framework for data access. Instead, it's building a new, more sustainable base on which to support not only the features and workflow you've depended on for years with EF, but also one that will enable so much more. The team struggled with whether this should be the next EF or a new data access technology. At one point, I even questioned whether it would be "EF Light." But the core functionality of EF is still there and after a lot of consideration, I agree it makes sense to think of this as the next version of Entity Framework. You can read more about this in the team's blog post, "EF7 – v1 or v7?" (bit.ly/1EFedRH).

Shedding Some Legacy Weight, Keeping the Good Parts

Yet there's also news about EF7 that's worrisome for some developers. While the most common EF classes, patterns and workflows will remain intact, some of the lesser-used members will be left behind. But please don't panic; I'll talk more about this in a bit.

Enabling developers to continue using familiar patterns and even be able to port good amounts of existing code to EF7 was a critical goal. You'll still use DbContext, DbSet, LINQ queries, SaveChanges and many of the means of interaction that have been part of EF for a long time.

Here's a DbContext class I defined in EF7:

```
public class BreweryContext : DbContext {  
  
    public DbSet<Brewery> Breweries { get; set; }  
    public DbSet<Beer> Beers { get; set; }  
}
```

And here's a simple update in EF7 that's the same as in EF6. I'm using a synchronous save, but all of the async methods are there, as well:

```
public void StoreBeers(List<Beer> beers) {  
    using (var context = new BreweryContext()) {  
        context.Beers.AddRange(beers);  
        context.SaveChanges();  
    }  
}
```

And a simple query:

```
using (var context = new BreweryContext()) {  
    return context.Breweries.Where(b=>b.Location.Contains("Vermont"));  
}
```

I'm using the version of EF7 that's in the packages with version beta2-11616. EF7 isn't really beta at this time, but the "beta2" is related to a NuGet package-naming decision. By the time this article is published, EF7 will have evolved further, so please consider this a look, not a promise.

I still have a DbContext and define DbSets, just as I've always done. OnModelCreating is still there, although I'm not using it here.

EF4.1 introduced the DbContext API, which was much more focused on typical EF usage. Underneath, it still relied on the originalObjectContext that provides database interaction, manages transactions and tracks the state of objects. Since then, DbContext has become the default class to use and you'd dip down to the lower-level APIs if you wanted to do a rare interaction with theObjectContext. EF7 will shed the obeseObjectContext; only the DbContext will remain. But some of those tasks you've relied on theObjectContext for will still be accessible.

Some of the very complicated mappings that are difficult to support and not commonly used will go away with EF7. The aforementioned

blog post says, "For example, you could have an inheritance hierarchy that combined TPH, TPT and TPC mappings, as well as Entity Splitting, all in the same hierarchy." If you've ever attempted to work directly with the MetadataWorkspace API and run away screaming, you know it's an intricate and complex beast, useful for being able to support this kind of flexibility. But that complexity has prevented the team from being able to support other scenarios for which users have asked. By simplifying the mapping possibilities, the MetadataWorkspace API also became simpler and much more flexible. You can easily get to metadata about your model schema from the DbContext API in EF7, which gives you a low-level capability to perform advanced techniques without having to have the low-levelObjectContext at your disposal.

Dropping EDMX, but Database First Will Continue

Entity Framework currently has two ways to describe a model. One uses an EDMX in the designer; the other involves the classes, a DbContext and mappings that are used by the Code First APIs. If you're using the EDMX and designer, at run time EF creates an in-memory model from the XML behind the EDMX. If you choose the Code First path, EF creates the same in-memory model by reading the classes, DbContext and mappings you provided. From that point on, EF works the same, regardless of how you describe your model. Note that with the EDMX/Designer workflow, you also get POCO classes and a DbContext to work with in your code. But because the EDMX is there, they aren't used to create that in-memory model. This is important to understand as you read the next sentences: EF7 will not support the designer-based EDMX model. It will not have the ability to read the EDMX XML at run time to create the in-memory model. It will use only the Code First workflow.

I still have a DbContext
and define DbSets, just as I've
always done.

When the team blogged about this, it caused panic among developers. Partly this was due to the fact that many still don't realize you can reverse-engineer a database to POCO classes, DbContext and mappings. In other words, you can start with a database to get a Code First model. This has been possible since the EF Power Tools Beta was first released in early 2011. It's supported by the EF6.1 designer and it will definitely be supported for EF7. I've said many times that the "Code First" moniker is a little confusing and misleading. It was originally called "Code Only," but the name was changed to "Code First" to make a nice match with "Database First" and "Model First."

So you don't need the designer or an EDMX to start with an existing database.

But what if you have existing EDMX models and don't want to lose the ability to use a designer? There are third-party designers that support Entity Framework, such as the LLBLGen Pro Designer,

which already supports EF Code First (bit.ly/110LIN2), and the Devart Entity Developer (bit.ly/1yHWbB2). Look for those tools and possibly others to potentially provide designer support for EF7.

There is yet another path to keep in mind: sticking with EF6!

Smaller Footprint, More Devices and OSes

Additionally, Microsoft strove to streamline the distribution of the EF APIs. The NuGet package folder for EF6.1.1 is about 22MB. This includes a 5.5MB assembly for the .NET Framework 4.5 and another in case you're using .NET Framework 4. With EF7, there are a number of smaller DLLs. You'll combine only the DLLs necessary to support your workflow. For example, if you're targeting SQL Server, you'd use a core EntityFramework.dll, a DLL for SQL Server and another with APIs common to relational data stores. If you want to use migrations, that's a separate assembly you can skip. Otherwise, you may want to create and execute migrations from the Package Manager Console. There's an API for commands. Using the NuGet package manager, the proper packages will be identified and downloaded via their dependencies, so you won't have to worry too much about the details.

When Entity Framework was first introduced, Microsoft had a vision of it being used for a variety of data stores, though the first pass focused on relational databases.

What this does is minimize the EF7 footprint on the end user's computer or device, which is especially important on devices. ASP.NET is going this route, as well. Both of these technologies are dropping their reliance on the full .NET Framework. Instead, they'll distribute only the DLLs necessary for accomplishing the tasks of a given application. This means the already-streamlined version of .NET used by Windows Phone and Windows Store apps will be able to use EF7.

It also means that OSes like OS X and Linux that use Mono rather than the full .NET Framework will also be able to support client-side Entity Framework.

Beyond Relational

When Entity Framework was first introduced, Microsoft had a vision of it being used for a variety of data stores, though the first pass focused on relational databases. Non-relational databases existed at that time, but were not widely used, unlike the NoSQL databases—especially document databases—that are so popular today.

While EF is an Object Relational Mapper (ORM), developers who use it want to be able to use the same constructs to interact with non-relational databases. EF7 will provide a high level of support

for this, but keep in mind what high level really means. There are vast differences between relational databases and non-relational databases and EF will not make any attempt to mask those differences. But for basic querying and updates, you'll be able to use the patterns with which you're already familiar.

Figure 1 shows code from a sample app that targets Microsoft Azure Table Storage, which is a non-relational document database. The sample comes from EF Program Manager Rowan Miller at github.com/rowanmiller/Demo-EF7. Note that the sample runs against the 11514 version of the EF7 alpha nightly builds.

The OnConfiguring method is new. It's a way to affect how EF configures the DbContext at run time, somewhat like you can do today with the DbConfiguration class. Notice the builder.UseAzureTableStorage extension method, which exists because I've also installed the EntityFramework.AzureTableStorage package into my project.

EF7 uses this pattern for its various providers. Here's an OnConfiguring method in a DbContext class within a project that targets SQLite:

```
protected override void OnConfiguring(DbContextOptions builder) {
    string dir = ApplicationData.Current.LocalFolder.Path;
    string connection = "Filename=" + Path.Combine(dir, "VermontBrewery.db");
    builder.UseSQLite(connection);
}
```

This project has the EntityFramework.SQLite package installed, so now I have the UseSQLite extension method instead.

Back in the WarrantyContext class in **Figure 1**, you can see the familiar OnModelCreating override for DbContext and in there I'm doing some special mapping. Again, I have methods provided by the EntityFramework.AzureTableStorage NuGet package. I get to pick and choose the packages I want based on the features I need. Azure Table Storage relies on a key-value pair for unique identity and to support table partitioning. In order to retrieve or store data, it's critical to know what values are to be used for the PartitionKey and the RowKey, so the API provides a method—PartitionAndRowKey—that allows you to map the properties to the appropriate keys. The concept is no different from how you've been able to use the fluent API or Data Annotations to specify the property that maps to a relational database's primary key.

Thanks to this mapping, I can write a familiar LINQ query to retrieve some data:

```
var warranty = _context.Warranties
    .Where(w =>
        w.BikeModelNo == modelNo
        && w.BikeSerialNo == serialNo)
    .SingleOrDefault();
```

So you're seeing a typical LINQ query, but it's being executed against the Azure Table Storage data store, just as you can do with a relational database today.

This same demo also updates warranty objects; creates and inserts new ones using DbSet.Add; and uses DbContext.SaveChanges to persist everything back to the data store, just as it's done today with EF6—and has been done throughout the history of EF.

Also interesting to consider is how Entity Framework has always supported a set of canonical features for mapping to relational databases, but left it up to the database providers to specify how those would translate to the database they target. EF7 will have a high-level set of canonical features that can be understood by relational and

non-relational data stores. There's also a lower-level set of features that focus on relational databases, and they're encapsulated in the Entity-Framework.Relational assembly. All of the relational database providers will depend on those and, just like today, their specific handling of database interaction will be housed in their own provider APIs, like the EntityFramework.SQLite I used earlier. You'll find extension methods in the providers that spin off of an AsRelational method, which is in the Relational API. It's an extension method of DbContext.

There's even an in-memory data store provider, which is for unit testing when you want to avoid a database interaction that might be involved in the logic you're testing. Typically in these scenarios you use fakes or mocking frameworks to impersonate the database interaction.

If you were setting up a test to perform a query or update the database, you'd have some code to instantiate the database, such as:

```
using (var context = new BreweryContext()) {  
    // Perform some action against the context  
}
```

You can easily switch to an in-memory store by first installing the entityframework.InMemory package to your test project, defining a DbContextOption for InMemoryStore and then specifying that the context should use that option. Again, this is possible thanks to extension methods provided by this API:

```
var options = new DbContextOptions().UseInMemoryStore();  
using (var context = new BreweryContext(options)){  
    // Perform some action against the context  
}
```

More Features, More Capabilities, Much More Flexible

You can already see the benefits of the new code base in the flexibility the extension methods provide, and in the ability to affect the Entity Framework pipeline with the OnConfiguring overload. There are extensibility points throughout the new code base, not only for changing EF7, but also for making it simpler for you to plug in your own logic to EF7.

The new core code base gives the EF team a chance to solve some age-old problems. For example, the version I'm using already has support for batch updating, which is the default for relational databases. I've played with code that allows me to use my own methods inline in LINQ queries without receiving the dreaded "Entity Framework cannot translate this method into SQL." Instead,

Figure 1 A DbContext Defined for Working with Azure Table Storage

```
public class WarrantyContext : DbContext  
{  
    public DbSet<WarrantyInfo> Warranties { get; set; }  
  
    protected override void OnConfiguring(DbContextOptions options) {  
        var connection =  
            ConfigurationManager.ConnectionStrings["WarrantyConnection"]  
                .ConnectionString;  
        options.UseAzureTableStorage(connection);  
    }  
  
    protected override void OnModelCreating(ModelBuilder builder) {  
        builder.Entity<WarrantyInfo>()  
            .ForAzureTableStorage()  
            .PartitionAndRowKey(w => w.BikeModelNo, w => w.BikeSerialNo);  
    }  
}
```

EF and the providers are able to parse out which part of the query will become SQL and which will get run locally on the client. I'm sure there will be protection from and guidance for avoiding some potential performance issues for that particular feature.

The team was able to add the long-requested Unique Foreign Keys capability for models. They're also looking closely at providing support for table-valued functions and cleaner ways to handle disconnected data, which is something I've focused on for many years with Entity Framework. It's a common problem with disconnected applications—not just when Entity Framework is involved—and it's not easy to create algorithms that will work consistently in every scenario. So a new approach is needed, for sure.

There's a lot more to get excited about with EF7. I highly recommend a close look at the posts on the ADO.NET Team Blog at blogs.msdn.com/adonet. In addition to the post I linked to earlier, Rowan Miller wrote in-depth about the decision to drop designer support in EF7; see "EF7 - What Does 'Code First Only' Really Mean" at bit.ly/1sLM3Ur. Keep an eye on that blog, as well as the GitHub project. The wiki on GitHub (bit.ly/1viwqXu) has links to how to access the nightly builds; how to download, compile and debug the source code; some walk-throughs and the design meeting notes. The team is eager for your feedback and is excited to receive pull requests.

A Decision Not Taken Lightly

It's important to me to write about EF7 to help allay some fears about such a big change and that some of the existing EF features that might be integral to your applications will not make it into EF7. These fears are not unfounded and the team is not taking them lightly, nor am I. But understanding that EF6 will not go away and will continue to evolve with contributions from the community is critical. If you want to take advantage of the forward movement, you'll have some tough choices to make. Upgrading big applications will not be easy and you should weigh the options carefully. Perhaps you can break up your application, rewriting only some portions to benefit from EF7.

Again, as I'm writing this column, EF7 is still in its early stages, and I'm not sure how far along it will be by the time you read this. But the current available source and NuGet packages are there to explore, experiment with and provide feedback. Bear in mind that the team may not always keep all of the provider APIs (such as Redis, SQLite and others) up-to-date as they evolve the core API. According to the post at bit.ly/1ykagf0, "EF7 - Priorities, Focus and Initial Release," the first release of EF7 will focus on compatibility with ASP.NET 5. Subsequent releases will add more features. Still, even though EF7 is not yet stable enough to begin building applications with, there's definitely enough there to let you start planning ahead. ■

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

THANKS to the following Microsoft technical expert for reviewing this article:
Rowan Miller

Working with Files?

30 DAYS
FREE TRIAL



Try **Aspose File APIs**

Save Time and Effort

Creating, Converting, Combining
Editing & Printing Files



Aspose.Words

DOC, DOCX, RTF, HTML, PDF...



Aspose.Email

MSG, EML, PST, EMLX...



Aspose.Pdf

PDF, XML, XLS-FO, HTML, BMP...



Aspose.BarCode

JPG, PNG, BMP, GIF, TIFF, WMF...



Aspose.Cells

XLS, XLSX, XLSM, XLTX, CSV...



Aspose.Imaging

PDF, BMP, JPG, GIF, TIFF, PNG..



Aspose.Slides

PPT, PPTX, POT, POTX, XPS...



Aspose.Tasks

XML, MPP, SVG, PDF, TIFF, PNG...

... and many more

100% Standalone - No Office Automation



Scan for 20% Savings!



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

EU: +44 141 416 1112
sales.europe@aspose.com

AU: +61 2 8003 5926
sales.asiapacific@aspose.com

14 Top Improvements in Visual Basic 14

Lucian Wischik

Visual Basic 14 is the newest version of Visual Basic, and it will ship as part of Visual Studio 2015. This version has been rewritten from scratch in about 1.3 million lines of VB code—earlier versions were actually written in C++. The team took advantage of the rewrite to rethink every part of VB from the ground up. I asked the team to pick out their top 14 improvements. They chose their favorites from across the board—in the coding experience, in the project system fundamentals and in the language itself.

Better Coding Experience

1. Refactorings “Just the fact that we finally get refactoring built directly into the product is huge.” —*.NET MVP Jim Wooley*

It used to be you had to buy add-on products just to get essential refactorings for Visual Basic, like extracting a method or inlining a temporary variable. There were a few refactorings for C#, and Microsoft had partnered with Developer Express to make its Refactor! add-in available to Visual Basic users since Visual Studio 2005. Now refactorings are built into Visual Studio 2015! To use them, click on an identifier, or highlight a sub-expression. Then hit Ctrl+Dot, or right-click and choose Quick Actions. This brings up a light bulb context menu of relevant actions, as shown in **Figure 1**.

Note that the refactorings are context-aware. For instance, if you extract the right-hand side of “Dim circleArea = Math.PI * radius * radius” out into a method, Visual Basic suggests the name “GetCircleArea” for that method. And it puts you into inline-rename mode if you want to change the name further. What’s smart about this inline-rename is that it can detect and warn you about name

clashes if you pick a name that’s already in use, it can avoid clashes where possible, and it works across your entire solution, even changing names in C# projects, as well.

2. Analyzers “The feature is spectacular. I have so many ideas for how to use this feature ... all the little things I see in people’s code.” —*Overheard from a Windows PowerShell MVP*

Analyzers are a way to put those light bulbs, code actions and error squiggles into your own hands. You can use them to enforce coding guidelines throughout your team. If you’re flown in to debug a problem, you can plug in an analyzer to quickly find common code defects in the entire solution. And many of the libraries you use can become “code-aware” with their own built-in analyzers. For instance, suppose you haven’t yet used the Microsoft Azure Storage library or haven’t read articles on best practices. Because the library now comes with an analyzer that detects common pitfalls in use of its APIs, you can be immediately confident that you’re using it properly. It’s like having an expert code reviewer stand over your shoulder as you type.

You can add analyzers to your project under the new References | Analyzers node, (or via NuGet). Once there they become part of your project’s compilation—they run live as you type to show live error squiggles. They run when you build your project in Visual Studio or on the command line, and will even run on build servers. Analyzers get a chance to “crack open” the internals of the compiler, to look at the syntax trees of the project’s source code and its types and members. Developers are pleasantly surprised to discover how easy it is to code their expert domain knowledge into analyzers, thanks to these syntax trees and types+members. My favorite analyzer is one that detects where my team has been using Async Sub methods that should have been Async Function ... As Task and issues a warning. This is a rough corner of asynchronous programming that not enough people are aware of, and it leads to difficult-to-catch concurrency bugs, so it’s great that my team can now catch the error at compile time. To get started writing your own analyzers, go to roslyn.codeplex.com.

3. No Need to Cursor off the Line “We do the right thing now.” —*Dustin Campbell, Visual Basic Team Member*

As a VB user, you’ve long been used to typing some code, then doing a quick “down-then-up cursor” to see if any error squiggles

This article refers to prerelease versions of Visual Basic 14 and Visual Studio 2015. All information is subject to change.

This article discusses:

- A better coding experience
- Enhanced project system fundamentals
- Language improvements

Technologies discussed:

Visual Basic 14, Visual Studio 2015

appear. Or you'd write code to fix an error squiggle, but then have to do the "down-then-up" for the squiggle to disappear.

Now you don't have to go through all of that. Just leave the cursor where it is, and error squiggles will appear or disappear themselves.

4. References in XML Doc Comments "For people passionate about docs, this is an enormous step in the right direction." —*.NET MVP Sam Harwell*

Are you passionate about XML doc comments? Here's a small example:

```
''' <summary>
''' Similar to <see cref="List(Of Integer).Count"/>
''' </summary>
''' <param name="e">Thing to count</param>
''' <remarks></remarks>
Sub Count(e As IEnumerable)
End Sub
```

In previous versions of VB, when you typed out cref and param-name in your comments, you got completion-list help, but beyond that you were on your own. The compiler did some minimal validation to check that the names existed, but they were typeset in grey and not easy to find, look up or refactor.

Now in Visual Basic 14 the cref and param-name arguments are colorized properly. You can hover over them to see tooltips. When you do a rename-symbol refactoring (Ctrl+R, Ctrl+R), Visual Basic renames all references to a symbol, including those in cref and param-name. You can right-click on one of them and Go to Definition, or Find All References. If you want to refer to a method that has several overloads, you can now unambiguously refer to the single overload you want. These changes all make it easier to type references in your XML doc-comments—and get them right.

Project System Fundamentals

5. References Node in Solution Explorer "Fact is, we all have to tweak references daily." —*Lucian Wischik, Visual Basic Team Member*

Figure 2 shows how a typical Visual Basic 14 project looks in Solution Explorer.

What's new is the References node. This used to be hidden and you had to click Show All Files to see it—but that also showed lots of irrelevant files.

This previous behavior might have made sense 10 years ago when you'd start with a Windows Forms project and it would generally have the right set of references. But it's a reality of modern development nowadays that the References node is used frequently—especially to manage NuGet references. It's a small but handy convenience to be able to find it easily in Solution Explorer.

6. Shared Projects "This is really just nice tooling on top of linked files that makes them easier to work with." —*Windows Developer MVP Morten Nielsen*

Suppose you want to share code between two or more projects. It's a common-enough situation, for example when maintaining both Windows Presentation Foundation (WPF) and Windows Phone versions of an app. The goal is always the same: maximize code reuse, so, for example, a bug fix you make for one project will automatically benefit the other project.

In the past, you could choose between two techniques: use linked files to share common source code, or rearchitect your shared code into a Portable Class Library to share a common binary. Now Visual Basic 14 allows a third, powerful technique: Shared Projects.

Why would you use Shared Projects? The task of sharing code is deep and challenging with no good "one-size-fits-all" solution. Portable Class Libraries are a good, clean solution, but they force you to architect your code so the common code never calls *into* the WPF or Phone projects; it only calls into system APIs that are present on both WPF and Phone. Shared Projects are easier to use because they don't require this rearchitecting.

To set up a shared project, right-click on your solution and select Add | New Project | VB | Shared Project. Next, right-click on each project's Reference node in turn and choose Add | Shared Projects. A Shared Project is just a collection of the source files, XAML files, images and other assets that will be included in each project that references it.

For each of your projects you can also set up My Project | Compile | Advanced Compile Options | Custom Constants with custom constants—WPF and PHONE, for example. Then, in your shared code, you can call into project-specific APIs like this:

```
#If WPF Then
' nothing needed
#elseif PHONE Then
ShowBatteryStatus()
#End If
```

7. 50 Percent Faster Compile Times "50 percent is no joke." —*.NET MVP Sam Harwell*

The Visual Basic compiler used to be written in C++. For Visual Basic 14, the team has rewritten it completely in VB, which has made it considerably faster! Here are two comparisons:

- A large solution build (1.3 million lines of code) went from 68 seconds to 41 seconds.
- A cold solution load (a Windows Store app) went from 6.7 seconds to 4.6 seconds

That's a considerable time saving. It's great when your mind can stay on track rather than meandering in the delay between "finish coding" and "press F5 to hit a breakpoint."

The 50 percent performance increase might come as a surprise to people who think that C++ is faster than VB. The truth is

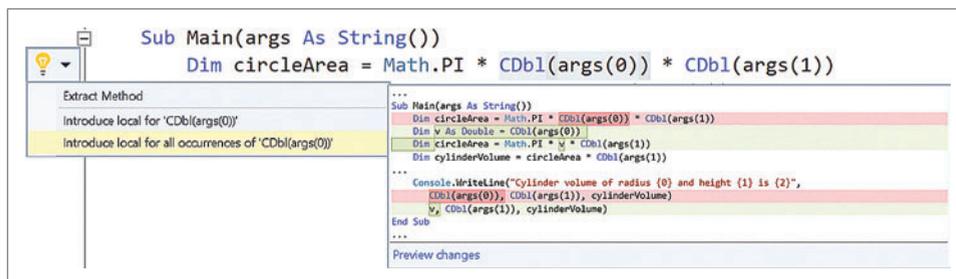


Figure 1 Visual Basic 14 Now Has Built-in Refactoring

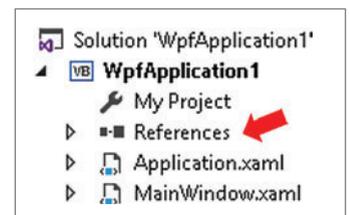


Figure 2 The References Node Is Now Shown in Solution Explorer

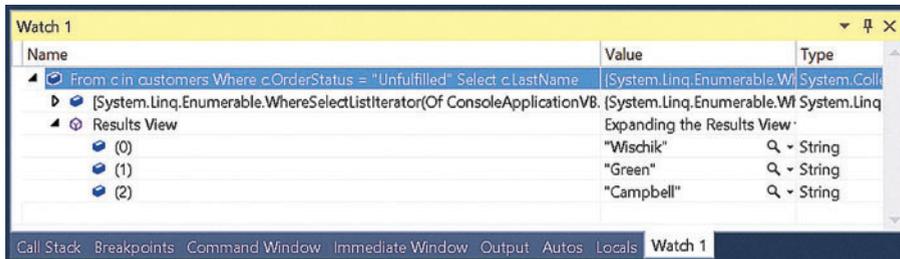


Figure 3 Lambdas and LINQ Expressions in the Watch Window

that algorithms, data structures and concurrency are where you get real speed benefits. The performance boost from rewriting in VB comes from many places: rethinking the data structures; being able to express algorithms more cleanly and refactor more safely; using async and the threadpool; using the Visual Studio profiling tools to discover CPU and memory-allocation hotspots; and using analyzers to detect simple .NET performance gotchas like unnecessary boxing.

8. Lambdas and LINQ Expressions in a Watch Window

“Awesome!”—*Visual Studio Uservoice User Marco Senn-Haag*

LINQ and lambdas are a great way to summarize data. One of the places this is needed most is at debug time, in the Watch and Immediate windows. In the past, any use of LINQ or lambdas here generated an error:

Evaluation of lambda expressions is not valid in the debugger.

Now, as you can see in **Figure 3**, it just works! For instance, if you're at a breakpoint where you have a collection called “customers,” you can get a quick read on the collection by writing this in the watch window:

```
From c In customers Where c.OrderStatus = "Unfulfilled" Select c.LastName
```

Did you know you can use the Immediate Window without even launching your program? For instance, if you've just written a module with a function GetName, you can open the Immediate window (Debug | Windows | Immediate) and type “? GetName()” and it will evaluate.

Visual Studio 2015 will also have better support for Edit and Continue such as in Async and Iterator methods, as well as in more common situations like inside of and around LINQ queries and lambdas, even allowing you to add a new query or lambda expression to an existing method. Although this didn't make it into the Visual Studio 2015 Preview, you'll be able to do all of this in the final release.

9. Better Error List

“The before and after really tell a great story.”—*Visual Basic Team Member Anthony D. Green*

The Error List in Visual Basic 14 has numerous practical improvements, ones that answer long-standing user requests (see **Figure 4**). Before, the Error List used to show fully qualified type names; now it shows only minimally qualified type

names so you can more easily read the error message. And it shows the error code, as well, which is handy for sorting by error code. Even better, the error code is a hyperlink to an Internet search, often more useful than a link to the MSDN documentation page. Also, you can filter each column in the error list like you do in Excel.

Sometimes when making big changes

it's easy to get your solution into a state where a lot of downstream code is broken. In the past, VB would show only the first 101 errors. This made it difficult to get an idea of just how widespread the break was, or to get an overview of what kinds of changes you'd have to make. Now, Visual Basic 14 shows all errors without limit. (This can be configured under Tools | Options | Text Editor | Basic | Advanced | Show diagnostics for closed files.)

Language Improvements

10. Null-Propagating Operators

“The constant need for null checking bites everyone both in productivity and in bugs.”—*.NET MVP Deborah Kurata*

Suppose you have a Customer class with an Address field that might legitimately be null, perhaps because your app doesn't require addresses to be typed in. Previously, any code that needed to do something with the address—like display it to the user—would need to be safeguarded with null checking to take this into account. Such null checks quickly get tedious. With Visual Basic 14 you can elegantly handle the possibility of a null like this, using the new ?. operator:

```
Console.WriteLine("{0} ({1})",
    customer.Name,
    customer.Address?.Country)
```

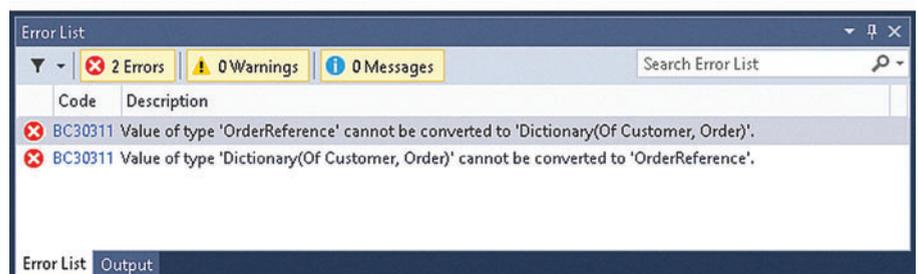
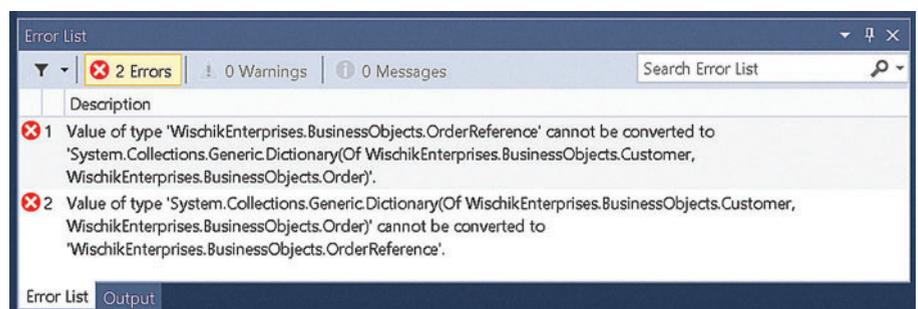


Figure 4 The Error List in Visual Studio 2015 (bottom) is more Readable and Versatile Than in Visual Studio 2013 (top)

The `?.` operator is simply shorthand for the common but cumbersome pattern of assigning to a temporary variable and then checking for null:

```
Dim _temp = customer.Address
Console.WriteLine("{0} ({1})",
    customer.Name,
    If(_temp Is Nothing, Nothing, _temp.Country))
```

You can use `?.` in a sequence and mix it with the regular dot operator, `a?.b.c?.d`, for example. It reads left-to-right. Any null value used by `?.` will just stop the sequence short and give the answer `Nothing`, and any null value used by `.` will throw a `NullReferenceException` as usual.

The `?.` operator is a null-conditional version of the `.` operator. There are null-conditional versions of most other operators, as well: indexing, `array?(i)`; delegate invocation, `delegate?(args)`; dictionary lookup, `dict?!key`; and XML axis properties, `xml?.@attr`, `xml?.<key>`, `xml?...<key>`.

You can use `?.` in other handy ways, too:

```
If customer?.Age > 50 Then ...
' If branch taken only if customer is non-null AND is older than 50
```

```
Dim name = If(customer?.Name, "blank")
' Pick a default name if customer is null
```

```
Dim first = customers?.FirstOrDefault()
' Only invoke this method if customers is non-null
```

11. Multiline String Literals “Is it sad that I’m so excited for multiline string literals?”—*Forum User Scaramouche*

This is how you used to write multiline strings, peppered with `vbCrLf`:

```
Dim json = "{ " & vbCrLf &
    " 'Name': 'Bad Boys'," & vbCrLf &
    " 'ReleaseDate': '1995-4-7T00:00:00'," & vbCrLf &
    " 'Genres': ['Action','Comedy']" & vbCrLf &
    "}"
```

Understandably, a common request has been to allow string literals that span multiple lines. Now, in Visual Basic 14, you can:

```
Dim json = "{
    'Name': 'Bad Boys',
    'ReleaseDate': '1995-4-7T00:00:00',
    'Genres': ['Action','Comedy']
}"
```

A related noteworthy feature—also commonly requested—is that you can now put comments within multiline statements. Previously, they weren’t allowed inside LINQ expressions like the following:

```
Dim q = From x In y ' This is a from clause
    Where x < z ' And this is the where
    Select x ' This select is redundant
```

12. String Interpolation “String interpolation makes for much simpler code, and makes the intent pop out.”—*Channel9 User Judah*

String interpolation is an easier way of writing strings with expressions in them, like so:

```
Dim s = $"hello {p.Name} you are {p.Height:0.00}m tall!"
```

This is simply shorthand for the following:

```
Dim s = String.Format("hello {0} you are {1:0.00}m tall", p.Name, p.Height)
```

String interpolation is often easier to write than an explicit call to `String.Format` because it saves you having to juggle the positional placeholders `{0}` and `{1}`. And, of course, there’s full colorization and IntelliSense for the expressions inside the holes. String interpolation works particularly well with programmatic strings, as in these examples:

```
Dim fn = $"C:\Documents\{folder}\{file}.ext)"
Dim url = $"http://{site}/{path}/{file}?search={query}"
```

As is normal for `String.Format`, this formats the string using the current culture. If you’re constructing a programmatic string that includes a floating point number, such as when you’re passing latitude and longitude to a Web service, you’ll most likely want `InvariantCulture` instead. This will be supported in Visual Studio 2015, but at press time the design wasn’t yet settled.

Note that string interpolation isn’t yet in Visual Studio 2015 Preview, but will be present in Visual Studio 2015 before its final.

13. NameOf “This will be such a help for so many scenarios I can think of.”—*Roslyn Forum Member ewwloyd*

The `NameOf` operator is a better way of embedding a string literal in your code, when that string literal refers to a name in your source code. Here’s one example:

```
Sub f(s As String)
    If s Is Nothing Then Throw New ArgumentNullException(NameOf(s))
End Sub
```

The `NameOf` operator isn’t evaluated at run time: It’s a compile-time constant, in this case the constant string “s.” The reason to use `NameOf(s)` is that it safeguards you against typos. For instance, if you rename the method parameter, then the `NameOf` argument will be renamed automatically. That wouldn’t have happened with just a string literal. Here’s another place where `NameOf` works well:

```
Private _age As Integer
Property Age As Integer
Get
    Return _age
End Get
Set
    _age = Value
    RaiseEvent PropertyChanged(
        Me, New PropertyChangedEventArgs(NameOf(Age)))
End Set
End Property
```

Note that `NameOf` isn’t yet in Visual Studio 2015 Preview, but will be present in Visual Studio 2015 before its final.

14. Open Source “We’re trying to engage the community. There are a lot of smart people out there. We’ll look at pull requests from the community just like we do our own ideas.”—*C#/Visual Basic Architect Anders Hejlsberg*

The final improvement is not in Visual Basic itself, but in the process of working with VB.

The source code of the VB compiler is now open source. So is the design process of the language itself. Each new feature proposal is made in the open, with full public scrutiny. The members of the Microsoft Visual Basic Language Design Team are essentially now stewards of the language. The team looks at proposals, considers them deeply, sees if there are unexpected gotchas or corner-cases, and determines if they meet the bar for inclusion in the language. The Language Design Meeting minutes are published openly. It’s truly an exciting time to be on the Visual Basic Language Design Team, and an exciting time to be a user of VB.

Wrapping Up

There are a lot of improvements in Visual Basic 14. This article has covered almost half of them. The general theme has been to make the existing VB work better in easy-to-use ways, without introducing difficult new concepts. For more information, check out roslyn.codeplex.com and blogs.msdn.com/vbteam. ■

LUCIAN WISCHIK is on the Visual Basic/C# Language Design Team at Microsoft, with particular responsibility for VB. Before joining Microsoft he worked in academia on concurrency theory and async. He’s a keen sailor and long-distance swimmer. Reach him at lwischik@microsoft.com.

THANKS to the following Microsoft technical experts for reviewing this article: *Dustin Campbell* and *Anthony D. Green*

We didn't invent the Internet...

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



TOOLS • COMPONENTS • ENTERPRISE ADAPTERS

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



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

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

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

connectivity
powered by

To learn more please visit our website →

www.nsoftware.com

Web-Based Test Case Management with TFS

Manoj Bableshwar

Application Lifecycle Management with Team Foundation Server (TFS) is all about leveraging an integrated toolset to manage your software projects, from planning and development through testing and deployment. As a core part of Team Foundation Server, the Test Hub enables you to create and run manual tests through an easy-to-use Web-based interface that can be accessed via all major browsers on any platform. In this article, I'll delve into the phases of manual testing—planning and creating tests, reviewing them with stakeholders, running the tests, and tracking the test progress of the team. I'll touch on different value propositions, such as the flexibility to customize workflows; end-to-end traceability; criteria-based test selection; change tracking and audit; sharing test steps and test data; stakeholder review; and most important, ease of use, especially for testers who've been using Excel-based frameworks

for manual testing. To access the Test Hub, you can navigate to it by clicking on the Test tab in on-premises TFS, just the way you access the Work tab to manage backlogs or the Build tab to monitor builds. Alternatively, you can sign up for a free Visual Studio Online (VSO) account at visualstudio.com and activate the 90-day account trial to try out Test Hub.

Plan Test Activity for the Sprint

Sprints or iterations are units of planning for teams that practice Agile or Scrum methodologies. It makes sense to plan test efforts for a sprint, just as it's done for user stories. To get started with test planning, create a test plan by providing a name and associating it with a team and a sprint. The test plan can have an owner and test cycle dates for out-of-band test activity such as a beta release sign-off or a user acceptance test cycle. Test plans in TFS are work items, so you get all the benefits of work items, such as change-tracking with work item history; permissions based on area paths; rich-text summary fields; file attachments and more. However, the most important benefit of work items is customization. Work item customization makes it possible to align the workflows and fields of artifacts used for tracking activities with the business processes used by the organization. This concept can be extended to better reflect the test activities practiced as part of your software development model, by customizing test plan work items. Moreover, the process of customizing test plan work items is similar to that of other work items, such as bugs or user stories. For example, the

This article discusses:

- Planning, creating, running and tracking manual tests on any platform
- End-to-end traceability of test activity across user stories, test cases and bugs
- Reviewing test scenarios with stakeholders, measuring feature quality and gauging if a feature is ready to ship

Technologies discussed:

Visual Studio Online, Team Foundation Server

default states of a test plan can be changed from Active and Inactive to, say, Authoring, Testing, or Archived. Additional user fields such as reviewers, approvers, sign-off owner, and so forth, needed for accountability or audit requirements, can be added to the test plan. As you integrate your processes into the test plan, you may want to restrict access to it, so that only certain people, such as team leads or test managers, have access to create and modify test plans. The Manage Test Plans permission can be used to moderate access to test plans at a user or team level.

Once you've set up a test plan, you'll be eager to create and run tests. But before that, it's important to think about the best way to

organize those tests to enable reuse and end-to-end traceability of test efforts. Test suites are artifacts that are contained in a test plan and enable grouping of test cases into logical units. Test suites are of three types: requirement-based test suites (RBS), query-based test suites (QBS) and static test suites. Static test suites work like folders to organize RBS and QBS. If you want to group test cases yourself, you can manually pick and add test cases to static test suites.

Once you've set up a test plan, you'll be eager to create and run tests.

Like test plans, test suites are work items, so all the customization benefits mentioned earlier apply to test suites. Some examples of custom fields for a test suite are summary fields describing instructions to set up the test application and fields to describe the nature of tests such as functional or integration, test complexity, and so on. Just as with test plans, you can moderate access to test suites at a user or team level with the Manage Test Suites permission. Changes to test cases contained in the suite, owner, state or other fields can be tracked in the test suite work item history.

End-to-end Traceability with Requirement-Based Suites

Requirement-based suites correspond to user stories (or product backlog items for scrum and requirements for CMMI-based projects) that the team is working on in the current sprint. The goal of creating an RBS by picking a user story is to enable traceability. Test cases created in an RBS are automatically linked to a user story, making it easy to find the scenarios covered to test the user story. Bugs, if any, that are filed while running these test cases are also linked to the user

The screenshot shows the Visual Studio Online interface for a test plan. The main area displays a table with columns for ID, Title, Step Action, Step Expected Result, Priority, and Tags. The table contains three test cases (983, 984, 985) with their respective steps and expected results.

ID	Title	Step Action	Step Expected Result	Priority	Tags
983	Escalate support ticket manually	Shared steps 987: Navigate to 'find open tickets' Enter CustomerID and tab out Select a ticket from list of customer tickets Click on Escalate and pick Level 2 Enter justification Click confirm	The ticket level changes to L2	2	E2E; ticket
984	Automatic escalation based on age	Shared steps 987: Navigate to 'find open tickets' Enter CustomerID and tab out Select a ticket from list of customer tickets that is mo... than 3 days old	Confirm that the level has changed to L2, with system generated justification as Age > 3	2	SystemValidated
985	L4 escalation support only if ticket older than 5 days	Shared steps 988: Login to fabrikamfiber.com Click on Tickets tab Click on find open tickets Enter CustomerID and tab out Select a ticket from list of customer tickets that is less than 5 days old Click on Escalate and pick Level 4	Error that says "Level 4 escalation permitted for tickets older than 5 days"	2	ticket

Figure 1 The Excel-Like Grid Can Be Used to Create Multiple Tests

story and the test case, thus providing end-to-end visibility of a user story, its test scenarios and open bugs. This helps you measure the quality and ship-readiness of a feature.

Criteria-Based Testing with Query-Based Suites

Regression-test coverage is as important as test coverage for new features. Teams typically set up regression-test coverage based on criteria—all priority 1 tests, all end-to-end scenario tests, all automated tests and so forth. Test Hub supports criteria-based testing with QBS; these suites are created by defining a query on test cases. Test cases that match the query criteria are automatically populated in the QBS, without any need to manually refresh the QBS. QBS can also be used in other scenarios, such as tracking test cases for bugs that are being fixed in the current sprint.

Creating Test Cases with an Excel-Like Grid

Test cases are the basic units of testing, each containing test steps that describe a set of actions to be performed, and expected results that describe what has to be validated at each test step. Each test step can have an optional attachment, for example, a screenshot that illustrates the output. Like test plans and test suites, test cases are work items, so all benefits of work item customization apply to test cases, as well.

There are two ways to create test cases. The first option is to use the test case work item form, which lets you create one test case at a time. The second option, and the one that really lets you breeze through creating test cases, is the Excel-like grid shown in Figure 1. The grid resonates very well with manual testers, who, typically, would've written and tested their test cases in Excel. With the grid, testers can create multiple test cases at a time, fluently typing test titles, steps, and expected results while navigating the grid with tabs, arrows, and the Enter key. It's a simple experience to insert, delete, cut, copy and paste rows. What's more, the grid can display all test case fields, such as state, tags, automation status and so

on, plus these fields can be bulk-marked for multiple test cases. If you have an intermittent Internet connection or are just more comfortable writing test cases in Excel, you're welcome to do that. Just copy and paste all the test cases you've written in Excel into the grid and save them to populate them into the system. In fact, if your team is just adopting the TFS Test Hub for testing, the grid can help you import your test cases from Excel. Check out the Test Case Migrator Plus utility at tcmimport.codeplex.com for advanced import requirements from Excel.

Share Test Steps and Test Data

Some test scenarios need specific test data as input to be meaningfully tested. Also, it makes sense to repeat tests with different variants of test data, for example, valid and invalid input sets or different combinations of items in a shopping basket. Parameters can be used to associate a test case with test data. With mature test teams that cover large and complex test scenarios, it's quite possible that many test cases use similar test data to drive testing. Shared parameters can help you consolidate and centrally manage such test data. You can also import test data from Excel and use it to drive tests through shared parameters.

Just as with the test data, it's possible the test steps are common across multiple test cases, for example the steps to log into an application or navigate to a form. Such common test steps can be consolidated into shared steps. The advantage of using shared steps is that a change, such as an updated application URL or an additional authentication step while logging in, can be updated in the shared step. Changes to shared parameters or shared steps will reflect across all referenced test cases instantly.

Review Tests with Stakeholders

Before running tests, it's a good idea to share the tests with stakeholders, such as product managers or business analysts, to solicit their comments. In cross-division or cross-organization development and test teams, such as outsourced test projects, a formal signoff may be required before proceeding with test execution. To share tests with stakeholders for review, you can export a test plan or a bunch of test suites by e-mail or print them to PDF or hard copy. The output of the e-mail dialog can be edited before sending it to stakeholders. You can also copy and paste to Word documents when stakeholders are required to respond with inline review comments.

Running Tests with the Web-Based Test Runner

To prepare the team to run tests, the test lead can assign tests to team members. The owner of a test case and the tester of a test case can be different people; the test lead has the flexibility to shuffle testers or even take the help of vendors to have tests executed. The most valuable feature of the Web-based Test Runner, which is used to run manual tests, is its cross-platform support. Because the Test Runner is browser-based, you can run it on any platform that supports any major browser—Internet Explorer, Chrome, Firefox and Safari.

The Test Runner presents the test steps and expected results in a narrow window, making it easy to read and execute the steps on the application being tested (see **Figure 2**). Image attachments that were created while writing the test case are visible and can be zoomed into. If your test case is driven by test data, then each row of parameter values included in the test case will correspond to one iteration of the test.

A test can have different outcomes—Passed, Failed, Blocked and

Not Applicable. The Blocked state can be used when tests are waiting on an external dependency, such as a bug fix, and Not Applicable is useful when a test doesn't apply to the current feature—a service release, for example. As you walk through validating the test steps, you mark them passed or failed. For failed steps, you can jot down comments for issues you observed while testing. You can report the failure to developers by creating a bug, right in the context of the Test Runner session. The bug is auto-populated with all the steps performed before you encountered the issue. The bug can also be updated with additional comments and screenshots before filing it. The bug is linked to the test case that was run while filing it and the requirement being tested, thus enabling end-to-end traceability. On the other hand, if you find that the discrepancy between the

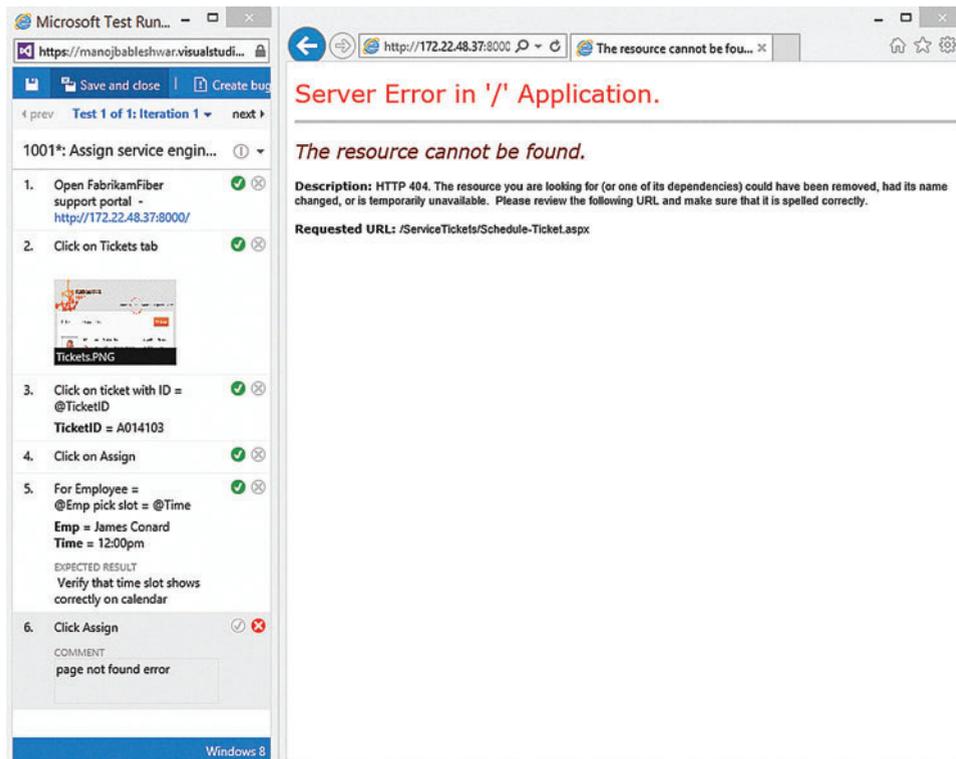


Figure 2 Web-Based Test Runner



NCache Goes Open Source!

Extreme Performance Linear Scalability

For .NET & Java Apps

(Microsoft Azure Supported)

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

In-Memory Distributed Cache

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

ASP.NET Optimization in Web Farms

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

Runtime Data Sharing

- Powerful event notifications for pub/sub data sharing
- Continuous Query events



FREE Download

sales@alachisoft.com

US: +1 (925) 236 3830

www.alachisoft.com

expected results and the application is because the application was recently updated, you can fix the test case inline, while it's running. If you're in a really long test session running many tests and need to take a break, you can pause the tests and resume them later. If you find that a test is failing for you, and wish to find out when it last passed or which team member got to execute it successfully, looking at the recent results of the test case will answer those questions.

While the Test Runner helps you walk through each test step of a test case in detail, the bulk-mark feature helps you pass or fail multiple tests at once. If you're validating high-level test scenarios highlighted by the test case title, but not actually walking through detailed test steps, you can quickly mark each test's outcome, without launching the Test Runner. The bulk-mark feature is particularly helpful when a large number of tests have been executed offline and their status has to be reflected back in the system.

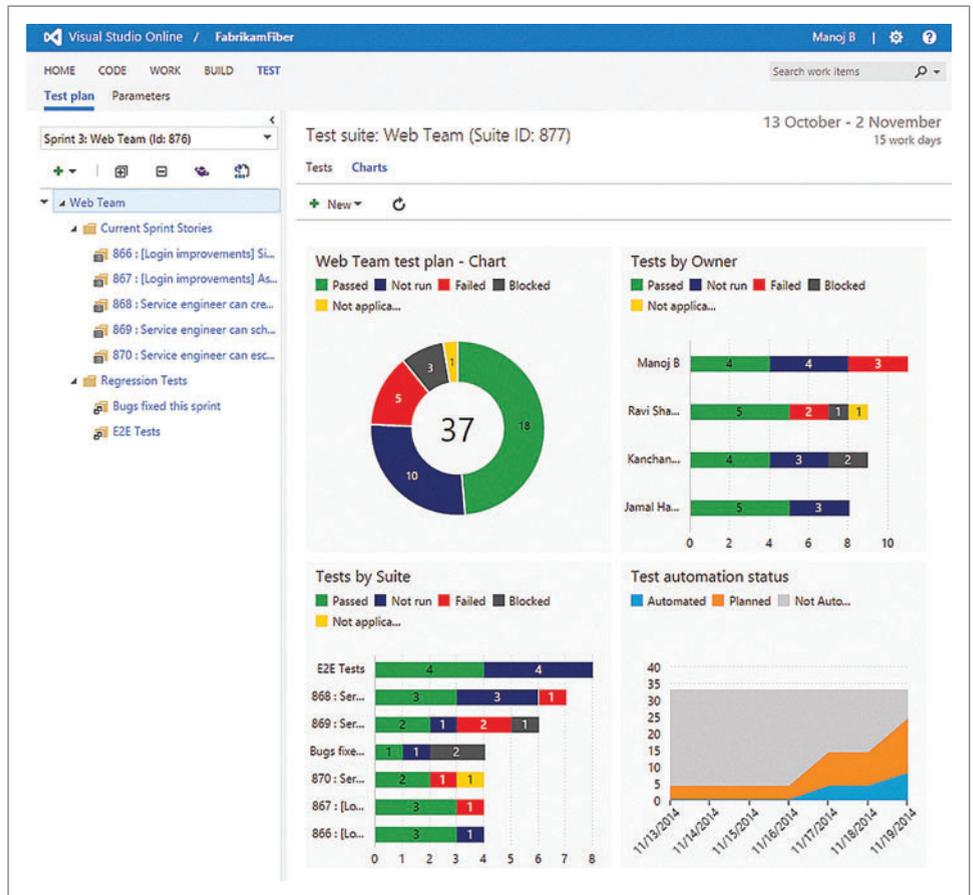


Figure 3 Tracking Test Results

Track Test Progress with Charts

“Is my feature ship-ready?” “Is my team on track to complete testing this sprint?” “Are all the test cases I planned for this sprint ready to run?” These are some of the questions in which test leads, test managers and stakeholders are interested. The Test Hub lets you create a rich set of charts to help answer such questions (see Figure 3). Charts come in two sets: test case charts that can be used to track the progress of test authoring activities, and test result charts that can be used to track test execution activities. And these charts can be different kinds of visualizations—pie, column, stacked bar, pivot table and so forth. Test case fields, such as owners, state, priority and the like can be used as pivots for test case charts. Test result charts come with the test suite, outcome, tester, run by, priority and more as pivots. For example, to find the test status of user stories, you can create a stacked bar chart with test suite and outcome as pivots for all the requirements-based suites being tested in the current sprint. These charts can either be created for a bunch of test suites or for a test plan to roll up information for the entire test plan. You can also share the insights with stakeholders by pinning the charts to the homepage. Finally, all the charts display real-time metrics, without any lag or processing delay.

Wrapping Up

The Test Hub isn't just for manual testers. It's a tool that product owners and business analysts can use to gauge how their features

measure up against the acceptance criteria. The grid can be used to keep track of acceptance criteria for requirements, and can later be used for sign-off. To summarize, the Test Hub offers:

- Customization of workflows with test plan, test suite and test case work items.
- End-to-end traceability from requirements to test cases and bugs with requirement-based test suites.
- Criteria-based test selection with query-based test suites.
- Excel-like interface with the grid for easy test case creation.
- Reusable test steps and test data with shared steps and shared parameters.
- Sharable test plans, test suites and test cases for reviewing with stakeholders.
- Browser-based test execution on any platform.
- Real-time charts for tracking test activity.

Test Hub provides an easy yet comprehensive way to test the user stories you plan to release in a sprint. Test Hub is available on-premises with TFS, as well as in the cloud with Visual Studio Online. Get started with a free 90-day trial right away at visualstudio.com. To see Test Hub in action, watch the demo at aka.ms/WebTCMDemo. ■

MANOJ BABLESHWAR is a program manager with the Visual Studio Online team at Microsoft. His team ships Manual Testing Tools to Visual Studio Online.

THANKS to the following Microsoft technical expert for reviewing this article: Ravi Shanker

Switch to Amyuni PDF



Create & Edit PDFs in .Net - ActiveX - WinRT

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



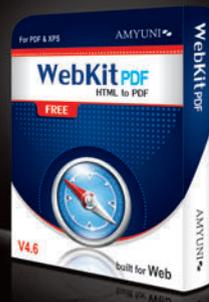
Complete Suite of Accurate PDF Components

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



Advanced HTML to PDF & XAML

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



High Performance PDF Printer For Desktops and Servers



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



AMYUNI

All development tools available at

www.amyuni.com

All trademarks are property of their respective owners. © Amyuni Technologies Inc. All rights reserved.

USA and Canada

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

Europe

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

Hadoop Made Easier for Microsoft Developers

Omid Afnan

This year Microsoft announced the availability of HDInsight, its Big Data cloud offering. With this service you're able to create Hadoop, HBase and now Storm clusters within Microsoft Azure. By using the Azure Management Portal, you can follow some simple steps to create a Hadoop cluster, associate Azure storage with it and choose the number of cores you need. With the click of a button, within minutes you have a cluster provisioned with the necessary computation resources and a 100 percent Apache Hadoop distribution up and running in the cloud. That's it: No hardware procurement, no sorting through Hadoop stacks, no need to learn how to install Hadoop. You're ready to run Big Data queries!

You may be saying: That's great ... now what? There are a number of common cases people are trying to address with Big Data. You or your organization may have some specific cases in mind, like Internet of Things (IoT) telemetry crunching, social sentiment analysis or user preference analysis. A case that's interesting to many of our customers is that of analyzing Web services or application logs. In this article I'm going to show you how this scenario

can be boot-strapped using HDInsight and how you can use the newly released Microsoft Azure HDInsight Tools for Visual Studio to simplify your development experience.

The Log Analysis Scenario

Most apps and services produce a large trail of data intended for tracking usage, understanding failures or managing the execution environment. Data streams such as error logs, software quality management (SQM) output, click-stream instrumentation and Web server logs can easily accumulate at the rate of gigabytes or terabytes a week. With the exception of debugging application failures or service outages, these logs are often left untouched due to their sheer size and the semi-structured nature of their contents.

However, many organizations are beginning to tackle this problem in order to discover quality issues more quickly, understand customer usage better and, ultimately, drive up customer engagement and satisfaction. A common base case is the analysis of Web server logs to derive two kinds of information:

- Frequency of particular errors happening on the Web site
- Traffic patterns into or out of the Web site

Parsing and aggregating this log data is an activity that allows a high degree of parallelism. The ideal would be to have subsets of records farmed out to individual servers that can parse, transform and summarize those records. Then you'd merge these partial results to create the final aggregated data. This is, of course, exactly what the MapReduce engine in Hadoop does and the Hive Query Language (HiveQL) allows you to express your target aggregations

This article discusses:

- Creating Hive queries in HDInsight
- Developer tooling for Big Data scenarios
- Analyzing Web server logs

Technologies discussed:

Microsoft Azure, Visual Studio, HDInsight, Hadoop

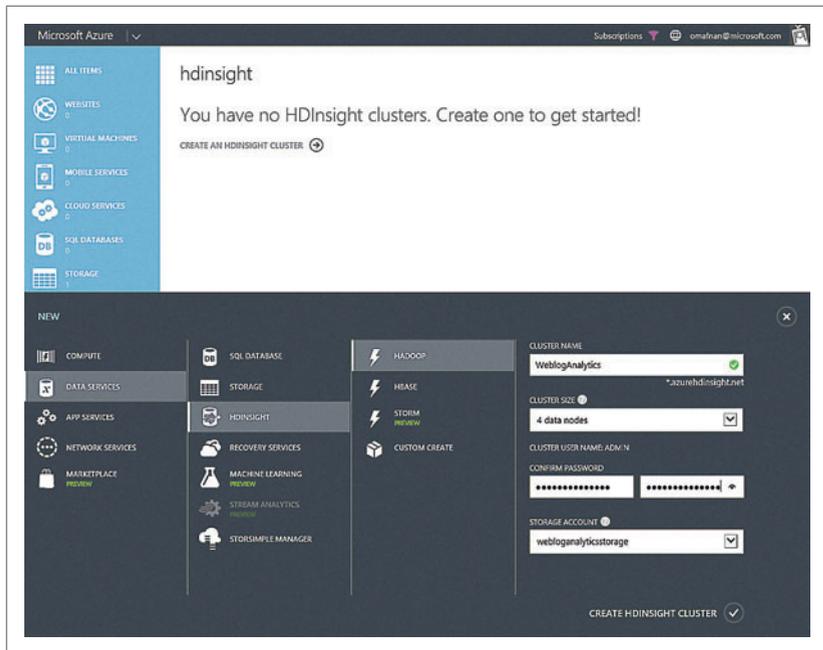


Figure 1 Creating a Hadoop Cluster in the Microsoft Azure Management Portal

in a SQL-like query syntax. As your data grows larger with increased usage, this approach scales effectively to keep overall processing time under control.

Programming against Hadoop MapReduce is made significantly easier with high-level languages like HiveQL. The HDInsight tools in Visual Studio bring the management of this kind of code into a standard framework familiar to all developers. The tool also exposes information related to the Hadoop cluster and the data artifacts in ways that are easy to understand. In the rest of this article I'll walk through a simple log analysis case and highlight how to use the language and the tool to get to a working solution quickly.

Getting Set Up

I'm going to start by creating an HDInsight cluster for running my queries. The cluster is needed only when running queries and can be deleted when not in use, or it can be left running (in which case you continue to incur charges). I'll put my persistent data on Azure storage, which is created separately and can be used with many different clusters over time. **Figure 1** shows where this is done in the Azure Management Portal.

After the cluster is created it will show up in the Management Portal, where you can view or delete it as needed. Note that there's also a Query Console, which you can open from the Management Portal to submit simple queries directly from the browser.

At this point you should have already installed the Azure SDK 2.5, which includes the HDInsight Tools for Visual Studio. When you navigate to the File | New | Project screen, you'll find a new category for HDInsight projects. Like other Visual Studio projects, the Hive Application

project type enables the easy creation of the related files needed for specifying a working application in a given language—in this case HiveQL. You'll also notice a Hive sample project that contains the code example presented in this article. Using the Hive project lets you easily group, save, modify and share related queries.

Getting the Data

Now, in Visual Studio, open Server Explorer to browse Azure services and objects in your subscription. Use this to navigate to the Azure | HDInsight node where you can find your recently created Hadoop cluster (called WeblogAnalytics in this example). Expand this node to find two children: Hadoop databases and the default blob storage you associated with the account at creation time. The blob storage now has a default container with the same name as your cluster, and Hadoop applications can read and write to this container and its subfolders using the URIs provided.

You can now copy or load your data into this container, and then simply reference the data by supplying the correct URI. If your Web site is located outside of Azure, use AzCopy to move the desired logs to this storage container, as needed. If you have existing data in Azure, you can access that data without further copying, which is much simpler if you're using Azure Websites to host your application. In this case, at cluster creation time, choose the storage account containing your Web site data as the associated storage account for your cluster. You have even more flexibility to add multiple storage containers using the HDInsight custom create option. In these cases you can easily access the required files with a reference such as:

```
wasb://container@storage.blob.core.windows.net/example/data/filename.ext
```

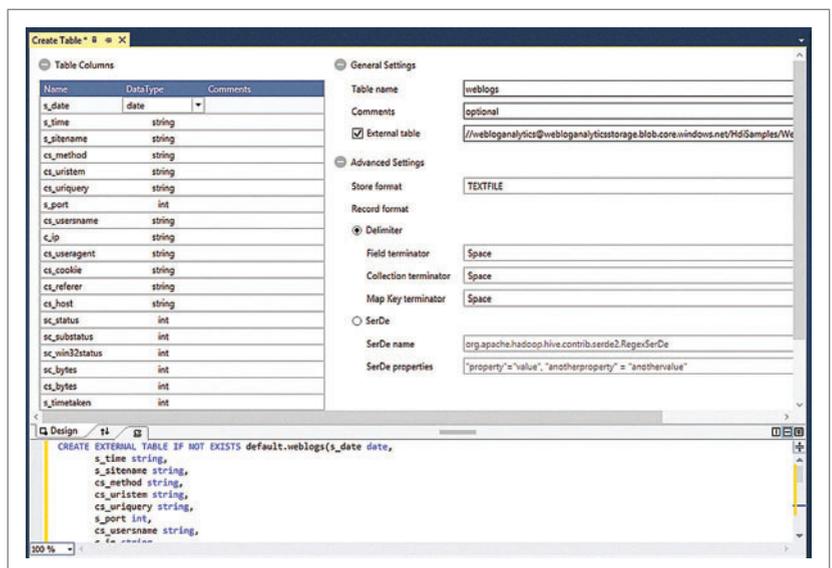


Figure 2 Creating a Hive Table Definition

Figure 3 Inserting Calculated Values

```
DROP TABLE IF EXISTS ClientErrors;
--create table ClientErrors for storing errors users experienced and
their frequencies
CREATE EXTERNAL TABLE ClientErrors(sc_status int, cs_referer string,
cs_page string, cnt int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';

--populate table ClientErrors with data from table weblogs
INSERT OVERWRITE TABLE ClientErrors
SELECT sc_status,
cs_referer,
concat(cs_uristem,'?', regexp_replace(cs_uriquery,
'X-ARR-LOG-ID=[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}', '')) cs_page,
count(distinct c_ip) as cnt
FROM weblogs
WHERE sc_status >=400 and sc_status < 500
GROUP BY sc_status,
cs_referer,
concat(cs_uristem,'?', regexp_replace(cs_uriquery,
'X-ARR-LOG-ID=[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}', ''))
ORDER BY cnt;
```

For this article I'm going to use some log data that's available as part of the sample code in HDInsight. You can navigate to the HDISamples folder in the default container where you'll find a set of folders containing samples, including the file:

```
HDISamples/webloganalyticsstorage/webloganalytics/HdiSamples/
WebsiteLogSampleData/SampleLog/909f2b.log
```

Once the raw data is available, you can turn it into a Hive table that can be referenced in other Hive queries as a rowset from which you can select rows. To do this you create it as an EXTERNAL table, which is a meta-data operation in Hadoop and doesn't change the data on disk. The only step you have to take is to define the column types and name them so you can refer to the data in the familiar SQL-style fashion. This is an important aspect of the Big Data model—that you can do a late binding between the format of your data and the data itself. There's no need to create schemas up front, or worry about cleansing your data at load time. In fact, there's no checking that the rows in your data adhere to any schema you define. Instead, it's your job to build your queries such that different versions of data or missing data is handled as your business logic requires.

Again, you can use the Visual Studio IDE to simplify this task. **Figure 2** shows the kind of table creation tool you may have seen in SQL Server tools. You can simply define table columns by adding entries to the Table Columns area, providing a name, data type and additional comments. Beyond a name, nothing else is needed to create a basic, empty table definition. However, in this case, I want to apply this definition to an existing file, so I use the External table property to specify the URI mentioned earlier.

Figure 4 Creating the Table and Inserting Data

```
DROP TABLE IF EXISTS RefersPerDay;
--create table RefersPerDay for storing references from external Web sites
CREATE EXTERNAL TABLE IF NOT EXISTS RefersPerDay(year int, month int, day int,
cs_referer string, cnt int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
--populate table RefersPerDay with data from the weblogs table
INSERT OVERWRITE TABLE RefersPerDay
SELECT year(s_date), month(s_date), day(s_date),
cs_referer, count(distinct c_ip) as cnt
FROM weblogs
WHERE sc_status >=200 and sc_status <300
GROUP BY s_date, cs_referer
ORDER BY cnt desc;
```

Metadata operations are achieved by running DDL statements in the Hadoop cluster. The create table dialog shows the DDL generated, which in this case works out to:

```
CREATE EXTERNAL TABLE IF NOT EXISTS weblogs (s_date date, s_time string,
s_sitename string, cs_method string, cs_uristem string,
cs_uriquery string, s_port int, cs_username string, cs_ip string,
cs_useragent string,
cs_cookie string, cs_referer string, cs_host string, sc_status int,
sc_substatus int,
sc_win32status int, sc_bytes int, cs_bytes int, s_timetaken int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ' '
STORED AS TEXTFILE LOCATION 'wasb://webloganalytics@
webloganalyticsstorage.blob.core.windows.net/HdiSamples/
WebsiteLogSampleData/SampleLog/'
TBLPROPERTIES ('skip.header.line.count'='2');
```

The DDL includes a couple of extra items. In this example, the sample file is using spaces to separate column values in each line, so I specify the field delimiter to be a space using the ROW FORMAT clause. Hive also allows you to set some specific metadata values related to a table. In the last line of the DDL I'm indicating that the text file contains two rows of header information that can be skipped. This DDL can be cut and pasted into another HiveQL script, or you can push the submit button to kick off the table creation. This results in a task being sent to the Hadoop cluster that will create the table. In Visual Studio you'll see a job status window, which you can refresh to track the task's completion status. Once the task is done, you can use Server Explorer to see the new table, browse its definition or view the first 100 rows (from the context menu of the database).

Cooking Data into Information

One of the key things you'll want to know is how well your Web site is working. A starting point might be to look at how often users are getting client errors on their HTTP requests. You can take a look at the different kinds of errors received by different source referrers, group them by error type and list them by most impacted (highest error count). You'll want to keep these results, so let's put them in a new Hive table, which you can do using a CREATE EXTERNAL TABLE statement. This time you'll populate the table by inserting calculated values using a select statement. **Figure 3** shows what the code looks like.

You'll notice I'm doing some cleanup in the data. For the cs-uri-query field, I simply want to see any query string that was passed in with the HTTP request, but don't care about the X-ARR-LOG-ID information stored in the logs. So, I simply replace that portion of the string with a blank. Of course, as this is a declarative query statement and not a procedure, it's not possible to specify reuse of the transformation I use for the cs-uri-query field. However, I can count on the query execution runtime to optimize this situation.

I'd further like to look at the top referrers for my Web site. To do this I can simply select the cs_referer field from the IIS logs and count the number of referrals by day and referrer. Once again, I want to keep this data for further use, so I create another Hive table called RefersPerDay and store the query results there. **Figure 4** shows how to do the table creation and data insertion.

All of the code in **Figure 3** and **Figure 4** can be put into a single Hive QL file. You can now create a new Hive application project, enter this code and run the queries. The Hive project allows you to save your query for later use. These examples are quite simple,

Data Quality Tools for Developers



A better way to build in data verification

Since 1985, Melissa Data has provided the tools developers need to enhance databases with clean, correct, and current contact data. Our powerful, yet affordable APIs and Cloud services provide maximum flexibility and ease of integration across industry-standard technologies, including .NET, Java, C, and C++. Build in a solid framework for data quality and protect your investments in data warehousing, business intelligence, and CRM.

- Verify international addresses for over 240 countries
- Enhance contact data with phone numbers and geocodes
- Find, match, and eliminate duplicate records
- Sample source code for rapid application development
- Free trials with 120-day ROI guarantee

Melissa Data.
Architecting data quality success.



Address
Verification



Phone
Verification



Email
Verification



Geocoding



Matching/
Dedupe



Change of
Address

MELISSA DATA[®]

| United States | United Kingdom | Germany | India |

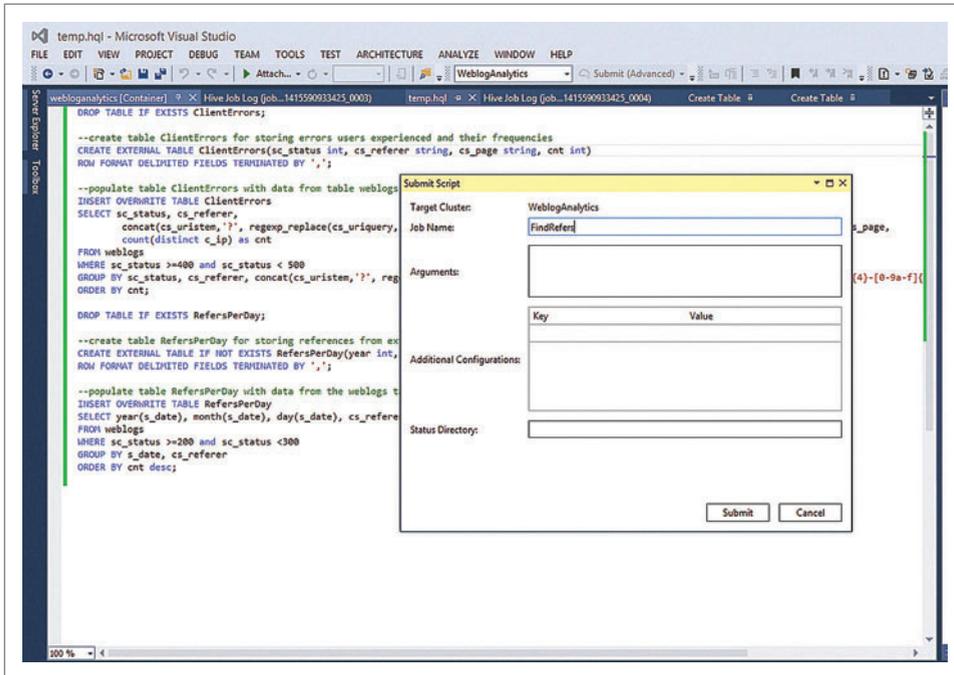


Figure 5 Submitting a Query

but they're based on the actual format of IIS logs. Therefore, they should work against your actual server logs and serve as a basis for creating more complex analyses.

Of course, running the queries means submitting them to your Hadoop cluster in Azure. To do this, you have to choose the cluster where you want to submit the job. The HDInsight toolbar allows you to select your cluster from a list of clusters available across your

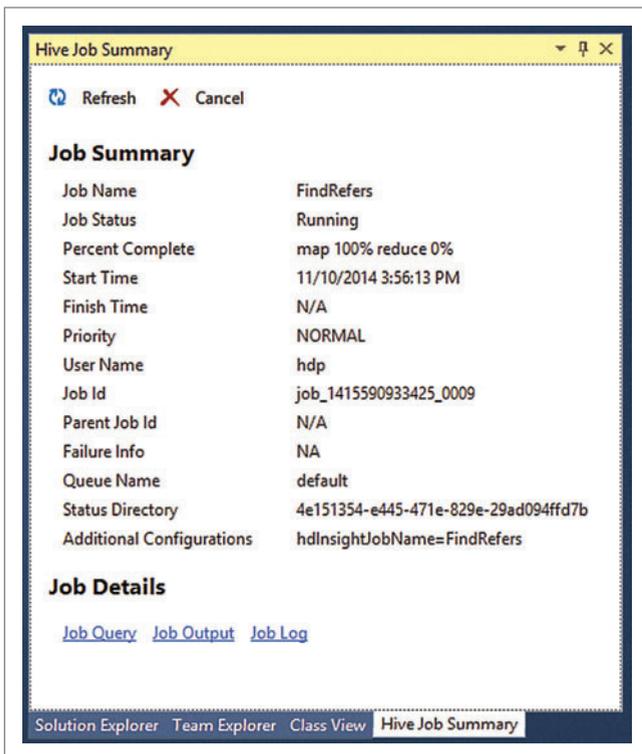


Figure 6 The Hive Job Summary Window

subscriptions. As you can see in **Figure 5**, I selected the Weblog-Analytics cluster. Choosing Submit will send the query to that cluster. Here I'm showing an advanced submit, where you can add a friendly name for the resulting job that will appear in the cluster. This makes it easier to track running queries. You can also add further arguments for the Hive processor.

As soon as the query is submitted, you'll see a job summary window like the one in **Figure 6**, indicating the current state of your query as it's initialized, starts to run, and eventually completes or fails. Your Hive queries are turned into MapReduce tasks that can be fanned out across a large number of nodes for execution. This is where the inherent power of the Hadoop processing model kicks in to allow

massive scaling. In the job summary view, you can see the job go through different percentages of the map and reduce phase until it completes all queries that were submitted together.

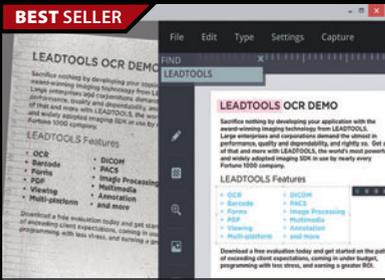
When the job finishes, you can review the associated query and look at job output and exceptions or errors through the links under Job Details. Of course, the query put summarized data into new tables and you can go back to Server Explorer to look at those tables and their content. You can also run new queries—ones that use the data in these new tables for further summarizing or aggregation.

Wrapping Up

With HDInsight you can quickly and easily access large amounts of data you've generated or transferred to the cloud. You can spin up Hadoop clusters to process the data, accessing the data from anywhere in Azure. Hive allows you to specify the structure of your data and to evolve with the changes that occur in your data format over time. HDInsight Tools for Visual Studio further allows you to explore your Hadoop clusters and data in intuitive ways. More important, it allows you to create Hive query projects and manage them like any other code, helping you move from simple, working queries to complex information processing over time. Aggregate data created using HDInsight is best visualized through tools such as Power BI, which offer direct connection to HDInsight sources. Other capabilities, like machine learning, can also be used in conjunction with these tools to extend the analytic possibilities. Watch this space for more on solving Big Data problems using Azure services. ■

OMID AFNAN is a principal program manager in the Azure Big Data team working on implementations of distributed computation systems and related developer tool-chains. He lives and works in China. Reach him at omafnan@microsoft.com.

THANKS to the following technical experts for reviewing this article:
Duc Le and Tony Murphy



LEADTOOLS Document Imaging SDKs V19 from \$2,995.00 SRP

Add powerful document imaging functionality to desktop, tablet, mobile & web applications.

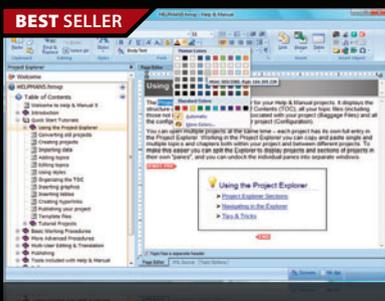
- Universal document viewer & conversion framework for PDF, Office, CAD, TIFF & more
- OCR, MICR, OMR, ICR and Forms Recognition supporting structured & unstructured forms
- PDF & PDF/A Create, Load, Save, View, Edit
- Barcode Detect, Read, Write for UPC, EAN, Code 128, Data Matrix, QR Code, PDF417
- Zero-footprint HTML5/JavaScript UI Controls & Web Services



ComponentOne Studio Enterprise 2014 v3 from \$1,315.60

.NET Tools for the Professional Developer: Windows, HTML5/Web, and XAML.

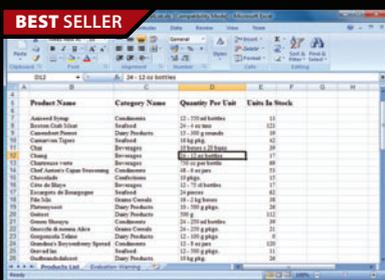
- Hundreds of UI controls for all .NET platforms including grids, charts, reports and schedulers
- A line of HTML5 and JavaScript products for enterprise application development
- Built in themes and an array of designers for creating custom themes and styling
- 40+ Windows 8.1 & Windows Phone 8.1 controls and Universal Windows app support
- All Microsoft platforms supported, Visual Studio 2013, ASP.NET, WinForms, WPF & more



Help & Manual Professional from \$583.10

Easily create documentation for Windows, the Web and iPad.

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



Apose.Total for .NET from \$2,449.02

Every Apose .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

Protect Your Online Services with Big Data and Machine Learning

Alisson Sol, Don Ankney and Eugene Bobukh

There are currently several methods for protecting online services, from the Security Development Lifecycle to operational processes for rapid incident response. One of the primary assets of online services is usually overlooked—the Big Data created by request logs and operational event monitoring. This article will explore usage data processing and machine learning (ML) techniques to improve security, based on the experiences of protecting online assets in the Microsoft Applications & Services Group (ASG), including services such as Bing, Bing Ads and MSN.

Most online services create several streams of logged data. While there's no standard taxonomy for the kinds of measurements you can store about a service, when you're exploring that data seeking security issues, you can broadly categorize it as usage data or operational data. Usage data includes any logged value regarding

use of the service by its target audience. A common example is a log entry for requests made to a Web site:

```
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port
cs-username c-ip cs(User-Agent) cs(Referer) sc-status sc-substatus sc-
win32-status time-taken
2014-10-27 20:46:57 127.0.0.1 GET /search q=elect
ion+results&form=PRUSEN&mkt=en-us 80 - 127.0.0.1
Mozilla/5.0+(Windows+NT+6.4;+WOW64;+Trident/7.0;+Touch;+rv:11.0)+like+Gecko
- 200 0 0 5265
```

This type of log entry contains data not only about the requested resource, but also the client browser, return code and time taken to complete the request. More sophisticated services may enrich usage data with derived information such as geolocation or application-specific information like user identification (for logged-in users). There would be no usage data without actual users, except perhaps for testing and monitoring agents.

Operational data refers to server and service operational measurements. This includes CPU utilization or temperature, disk space, network transfer rate, application exceptions, memory faults, and similar factors logged as soon as a server is turned on and a service started. In modern datacenters, log information typically includes not only computing devices, but also aspects such as air-conditioning measurements, presence of personnel and visitors in zones containing sensitive data, doors being opened and closed, and similar information required by operational security standards.

The code samples in this article will focus on processing usage data. However, you could apply most of the principles outlined and demonstrated here to identify vulnerabilities using operational

This article discusses:

- Protecting online services from attacks
- Analyzing usage data to determine nature of attacks
- Running test attacks against an attack surface

Technologies discussed:

Microsoft Bing, Microsoft Azure, Windows PowerShell

Code download available at:

msdn.microsoft.com/magazine/msdnmag0115

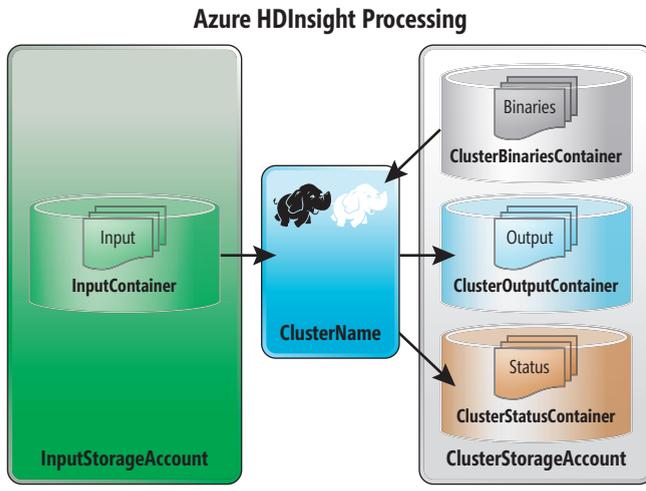


Figure 1 The Azure HDInsight Processing Environment

data. You can also improve your chances of identifying security incidents by correlating usage data with operational data.

Attacking Endpoints

The pace of changes for a large online service makes it hard to protect using only typical Security Development Lifecycle practices, such as code reviews and static analysis tools. Thousands of changes are committed every month. And there are often at least a few hundred experiments “in flight” at any given point. These experiments present new features to select users to gather feedback before widespread release. Besides following good development practices and having penetration test teams constantly trying to pinpoint vulnerabilities, it’s important to automate as much of the vulnerability discovery as possible.

Near the end of 2014, Microsoft Bing services generated hundreds of terabytes of usage data on a daily basis, logging up to hundreds of billions of requests. It’s safe to assume some of these requests were actually attacks, trying to identify or exploit vulnerabilities. A typical query to Bing is made sending to the service a URL request:

```
http://www.bing.com/search?q=election+results&form=PRUSEN&mkt=en-us
```

In this example, the user is searching for “election results.” There are two other parameters in the URL that identify the Web form that originated the request and the market setting (in this case,

Figure 2 Windows PowerShell Script XML Configuration File

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration>
  <SubscriptionName>Your-SubscriptionName</SubscriptionName>
  <ClusterName>Your-ClusterName</ClusterName>
  <ClusterStorageAccountName>Your-ClusterStorageAccountName
  </ClusterStorageAccountName>
  <ClusterBinariesContainer>Your-ClusterBinariesContainer
  </ClusterBinariesContainer>
  <MapperBinary>UsageDataMapper.exe</MapperBinary>
  <ReducerBinary>UsageDataReducer.exe</ReducerBinary>
  <ClusterOutputContainer>Your-ClusterOutputContainer</ClusterOutputContainer>
  <ClusterStatusContainer>Your-ClusterStatusContainer</ClusterStatusContainer>
  <InputStorageAccountName>Your-InputStorageAccountName
  </InputStorageAccountName>
  <InputStorageAccountKey>Your-InputStorageAccountKey</InputStorageAccountKey>
  <InputContainer>Your-InputContainer</InputContainer>
  <DeployBinaries>true</DeployBinaries>
  <DeployFlavor>Release</DeployFlavor>
  <JobTimeOut>3600</JobTimeOut>
</Configuration>
```

indicating the user language is English and in the United States). You may understand this as a call to the “search” application within the Bing domain, with parameters q, form and mkt. All such requests are expressed in a canonical format, like so:

```
http://www.bing.com/search?q=[]&form=[]&mkt=[]
```

There are other applications within the Bing domain answering similar requests. A request asking for “election results” in the video format would be:

```
http://www.bing.com/videos?q=election+results&form=PRUSEN&mkt=en-us
```

As online services grow, new applications and features are added dynamically—some for convenience and others for compatibility. Different formats are often allowed for the same request. Bing videos would also accept this request:

```
http://videos.bing.com/?q=election+results&form=PRUSEN&mkt=en-us
```

Assuming you could derive from the usage logs a list with all canonical requests to a service, you could then probe for vulnerabilities trying to inject known malicious payloads into the parameter values. For example, an intruder could use the following request to verify if the Bing video application is vulnerable to cross-site scripting (XSS) in the query parameter:

```
http://www.bing.com/videos?q=<script>alert("XSS")
</script>&form=PRUSEN&mkt=en-us
```

An intruder scanning for vulnerabilities will also test the responses for malicious payloads injected into other parameters, in all possible combinations. Once a vulnerability is found, an attack can be launched. Attack URLs are usually included in spam messages, in hopes a small percentage of users will carelessly click on the links. Some of those users may even suspect URLs containing JavaScript keywords. However, encoding the requests makes it more difficult to promptly identify attacks:

```
http://www.bing.com/videos?&form=PRUSEN&mkt=en-us&q=%3CscRipt%3Ealert(%2
2XSS%22)%3C%2FscriPt%3E
```

You can write an application that accepts the list of canonical requests to a service as input, injecting malicious payloads for each kind of possible vulnerability and detecting from the service answer if the attack succeeded. You can find the code for individual “detectors” for several kinds of vulnerabilities (XSS, SQL injection and open redirects) online. In this article, we’ll focus on finding the “attack surface” for online services from the usage data logs.

Processing Environment

Weblogs are usually distributed across several machines, making sequential log file reads extremely efficient (even better if the files are partitioned across different storage devices in a distributed file system). That makes processing Weblogs a great application for the MapReduce framework.

For this example, we’ll place Weblogs in Microsoft Azure Blobs under the same container called InputContainer. As a processing platform, we’ll use Azure HDInsight Streaming MapReduce jobs. There’s good information already available online on how to set up and configure HDInsight clusters. The code explained in this article will generate binaries you should place in a container accessible to the HDInsight cluster, referred to as ClusterBinariesContainer. As code executes and processes input, it will create output in another container called the ClusterOutputContainer, along with status information saved to the ClusterStatusContainer. A visualization of the Azure HDInsight processing configuration is shown in **Figure 1**.

Figure 3 Function to Extract Key and Values from Log Line

```
private static string ExtractKeyAndValuesFromLogLine(string inputLogLine)
{
    StringBuilder keyAndValues = new StringBuilder();

    string[] inputColumns = inputLogLine.Split(DataFormat.MapperInputColumnSeparator);
    string uriReference = inputColumns[DataFormat.MapperInputUriReferenceColumn];
    string uriQuery = inputColumns[DataFormat.MapperInputUriQueryColumn];
    string parameterNames = ExtractParameterNamesFromQuery(uriQuery);

    // Key = uriReference + separator + parameterNames
    keyAndValues.Append(uriReference);
    keyAndValues.Append(DataFormat.ReferenceFromQuerySeparator);
    keyAndValues.Append(parameterNames);

    keyAndValues.Append(DataFormat.MapperOutputColumnSeparator);
    keyAndValues.Append(DataFormat.OneOccurrence);

    return keyAndValues.ToString();
}
```

You need to replace the placeholder names in **Figure 1** with values for your specific configuration. You can set these in a configuration file. The Windows PowerShell script that will create and execute the HDInsight job will read the XML configuration file shown in **Figure 2**. After configuring the file, you'll most likely execute the script for usage data analysis from within an Azure PowerShell prompt, having properly configured your Azure account to have the authorization to access the storage and compute services (see `Get-AzureAccount` and `Add-AzureAccount` cmdlets help).

Map to Canonical Requests

Getting the attack surface for the online services using the MapReduce processing environment consists of creating a mapper application to extract the URLs from the Weblogs and transform them into canonical form. That value becomes the key for the reducer, which will then eliminate duplicates. That's the same principle used in the sample word count application available for HDInsight. Removing any comments and validation code, the following code demonstrates the main entry point for the mapper application:

```
public static void Main(string[] args)
{
    Console.SetIn(new StreamReader(args[0]));

    string inputLogLine;
    while ((inputLogLine = Console.ReadLine()) != null)
    {
        string outputKeyAndValues =
            ExtractKeyAndValuesFromLogLine(inputLogLine);

        Console.WriteLine(outputKeyAndValues);
    }
}
```

This code goes through every input line and extracts the unique key, as well as any complementary values relevant to the problem being solved. For example, if you were seeking the most common user queries, the key would be the value passed for the query parameter. Raw log lines appear as follows:

```
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port
cs-username c-ip cs(User-Agent) cs(Referer) sc-status sc-substatus sc-
win32-status time-taken
2014-10-27 20:46:57 127.0.0.1 GET /search q=elect
ion+results&form=PRUSEN&mkt=en-us 80 - 127.0.0.1
Mozilla/5.0+(Windows+NT+6.4;+WOW64;+Trident/7.0;+Touch;+rv:11.0)+like+Gecko
- 200 0 0 5265
```

Columns `cs-uri-stem` and `cs-uri-query` have the relevant information you need to have parsed to get the canonical form of the

Figure 4 Function to Get Just Parameter Names from Query

```
private static string ExtractParameterNamesFromQuery(string query)
{
    StringBuilder sb = new StringBuilder();

    // Go through each parameter adding to output string
    string[] nameValuePairs = query.Split(DataFormat.ParametersSeparator);
    Array.Sort(nameValuePairs, StringComparer.InvariantCultureIgnoreCase);
    List<string> uniqueParameterNames = new List<string>();
    foreach (string nameValuePair in nameValuePairs)
    {
        int indexOfSeparatorParameterNameFromValue =
            nameValuePair.IndexOf(DataFormat.ParameterNameFromValueSeparator);
        string paramName = nameValuePair;

        paramName = nameValuePair.Substring(0,
            indexOfSeparatorParameterNameFromValue);
        uniqueParameterNames.Add(paramName);
        sb.Append(paramName);
        sb.Append(DataFormat.ParameterNameFromValueSeparator);
        sb.Append(DataFormat.OneOccurrence);
        sb.Append(DataFormat.ParametersSeparator);
    }
    return sb.ToString();
}
```

Figure 5 Reducer Main Loop

```
public static void Main(string[] args)
{
    string currentKey, previousKey = null;
    int count = 0;
    Console.SetIn(new StreamReader(args[0]));

    string inputLine;
    while ((inputLine = Console.ReadLine()) != null)
    {
        string[] keyValuePair =
            inputLine.Split(DataFormat.ReducerInputColumnSeparator);
        currentKey = keyValuePair[0];
        if (currentKey != previousKey)
        {
            Console.WriteLine(DataFormat.ReducerOutputLineFormat, previousKey, count);
            count = 1;
            previousKey = currentKey;
        }
        else count++;
    }

    Console.WriteLine(DataFormat.ReducerOutputLineFormat, previousKey, count);
}
```

request (the sample code doesn't include multiple hosting processing). The function to extract key and values from each log line is outlined in **Figure 3**.

The only missing logic relates to extracting the parameter names from the query column. Code to perform the task is shown in **Figure 4**. The input for that function—the previously provided sample line—would be a string like this:

```
q=election+results&form=PRUSEN&mkt=en-us
```

The canonical form used in the sample code will remove the parameter values, sort the parameter names and transform this into a still valid query string:

```
form=1&mkt=1&q=1&
```

Sorting the parameter names helps avoid duplication, because Web requests don't depend on parameter order. The placeholder used for the parameter value is "1," instead of "[]" because it's shorter. It may also be used for other things like counting the number of times parameters appear in all combinations of request parameters, as shown in **Figure 4**.

Reduce the Attack Surface

The mapping code sequentially reads the Weblog lines, then outputs a key and value for each. MapReduce has a “combine” phase, which assembles all records with the same key for processing by the reducer code. If an input log had several lines doing search queries, by now those produce identical output:

```
search?form=1&mkt=1&q=1& 1
search?form=1&mkt=1&q=1& 1
search?form=1&mkt=1&q=1& 1
```

Figure 5 has the outline for the reducer code. It reads input lines and splits those into key and values. It keeps a counter until the key changes, and then outputs the result.

You could easily modify the code and script provided in this article for other purposes. Change function ExtractKeyAndValuesFromLogLine to have the parameter values as the keys and you'd have a useful distribution of value frequency. In its current form, the output will be a list with the attack surface, showing the normalized application path and frequency of requests:

```
search?form=1&mkt=1&q=1& 3
video?form=1&mkt=1&q=1& 10
```

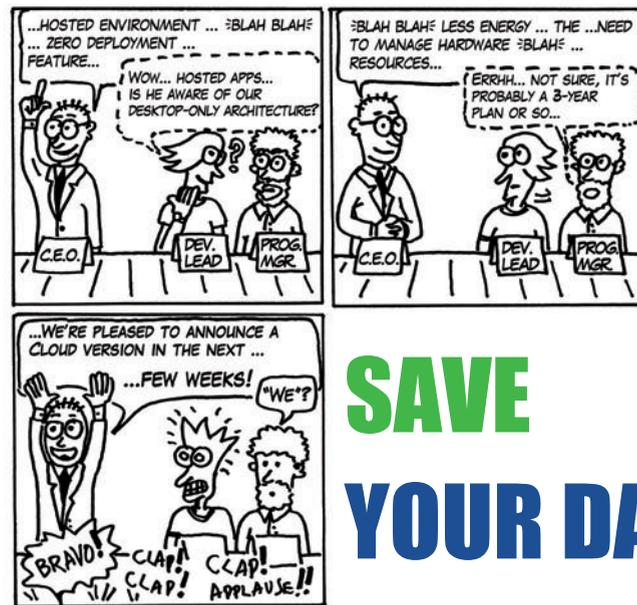
Understand Service Traffic

The attack surface will already be valuable to help you understand what's happening with your service, even if you don't perform proactive penetration testing to expose vulnerabilities. Although request frequency changes per hour, per day of the week, and with other seasonal factors, the normal behavior over time changes with feature releases or even coordinated attacks. For example, Bing receives hundreds of billions requests per day. The attack surface list typically has hundreds of thousands of values per day. Not all such paths are even expected. Figure 6 summarizes what's found on the attack surface on a typical day. The first row indicates the top 10 canonical paths amount to 89.8 percent of the usual request traffic. The next top 10 paths add up to 6.3 percent of the request count (or 96.1 percent for the top 20). Those are really the top applications for the service. Everything else amounts to less than 4 percent. Such a request pattern is very different for a site with syndicated content, like MSN.com.

It's particularly interesting to note that about two-thirds of the requests are to unique paths. Some of those requests may come from attackers trying to probe for application parameters that might trigger certain functionality. Yet the very nature of online services generates a lot of such traffic. Links to your service stored a few years back

may still be activated by humans or automated processes. While seeking an attacker, you may uncover the need to have a compatibility mode for old URLs, and automatically redirect those to new versions of the application. That's a good business result.

Developing the attacking application is a journey you should take with care. Assuming your detectors are all perfect, you'll now impose a load onto your service that needs to be properly throttled. You need to avoid creating a denial-of-service attack or affect performance for real users. It's also essential to avoid numerous false positives. If that



SAVE YOUR DAY!



Use CodeFluent Entities

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

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

Renato Xavier - Colombaroli - Brazil

* Source : <http://visualstudiogallery.msdn.microsoft.com/B6299BBF-1E11-436D-B618-66E8C16A8410>

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

 **CodeFluent Entities**
tools for developers, by developers

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

happens, incident reports from the attacking service will soon be ignored.

Learn from Service Data

ML lets you automate several processes that would be difficult to do by directly coding instructions or rules. For example, it would be hard to conceive the code for a computer vision application that would detect a person is in front of a camera. After labeling thousands of images with depth information, however, the Kinect team at Microsoft was able to “train” an ML module to do this with sufficient accuracy. The labeled depth images indicating not only human presence, but also body position, enabled the learning process.

An attack service that generates requests of known categories (XSS, SQL injection and so on) automates an important part of the process to use ML methods for evaluating online traffic. It generates a large body of synthetic ground truth. Check the usage logs to easily identify all such attack requests known to be made at a certain time by the attack service. They’re now mixed with user requests, for which there isn’t yet a known classification (normal or malicious request).

To create a stratification of the data that truly matches the traffic hitting the service, you have to understand the nature of that traffic. Generating an experimental sample with 0.1 percent of the usage data for a service receiving 100 billion requests a day still results in 100 million requests. Only synthetic data can help create such an initial ground truth.

Assuming you have high-quality ground truth and adequate tools, the iterative cycle for the learning process for an ML solution to evaluate user requests is outlined in **Figure 7**. Starting with synthetic data in the ground truth, you can make experiments and kick off a training process in the ML module to classify requests (into categories such as normal, XSS, SQL injection and so on) or make a regression (indicating the confidence a request belongs to one or more of the categories). You could then deploy this ML

Figure 6 Typical Distribution of Request Paths for Bing in 2013

Canonical Paths	Percentage
Top 10	89.8
Top 20	96.1
Paths with <= 1,000 requests	99.9
Paths with <= 100 requests	99.6
Paths with <= 10 requests	97.6
Paths with = 1 request	67.5

module as part of a solution and start receiving evaluation requests. The output is then subject to a scoring process, which will indicate whether the ML module correctly identified the requests (true positives and true negatives), missed suspicious requests (false negatives) or generated a false alert (false positive).

If the initial experiments produced a good enough ML module based on synthetic data, that module should be fairly accurate with few incorrectly evaluated actual user requests. You

can then properly label those that were wrongly evaluated and add them to the ground truth. A few more experiments and training should now generate a new ML module with restored accuracy. As you carefully iterate this process, the initial synthetic data becomes a smaller part of the ground truth used in the training process, and iterations of the ML module get better at accurately evaluating user requests. For additional validation, you can use the ML module for offline applications examining usage logs and identifying malicious requests. After sufficient development, you can deploy the ML module online to evaluate requests in real time and prevent attacks from ever hitting back-end applications.

Wrapping Up

While you should continue to follow solid development processes (including the Security Development Lifecycle), you should also assume your online service may be under attack at any point. Usage logs can provide you with insightful information about how such attacks are occurring. Knowing your attack surface will help you proactively attack your service to identify and close vulnerabilities before they’re exploited. Building that attack service then creates synthetic ground truth, enabling the use of ML techniques to train an ML module to evaluate service requests. Building the attack service is not a trivial task, but the immediate and long-term business results more than justify the investment. ■

Learning Cycle

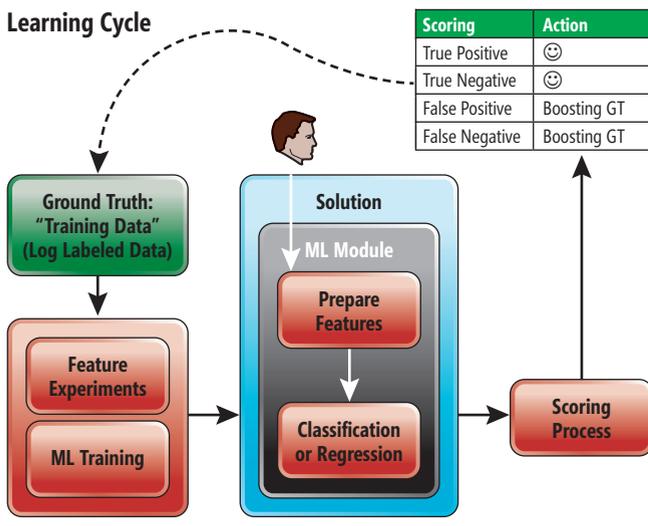


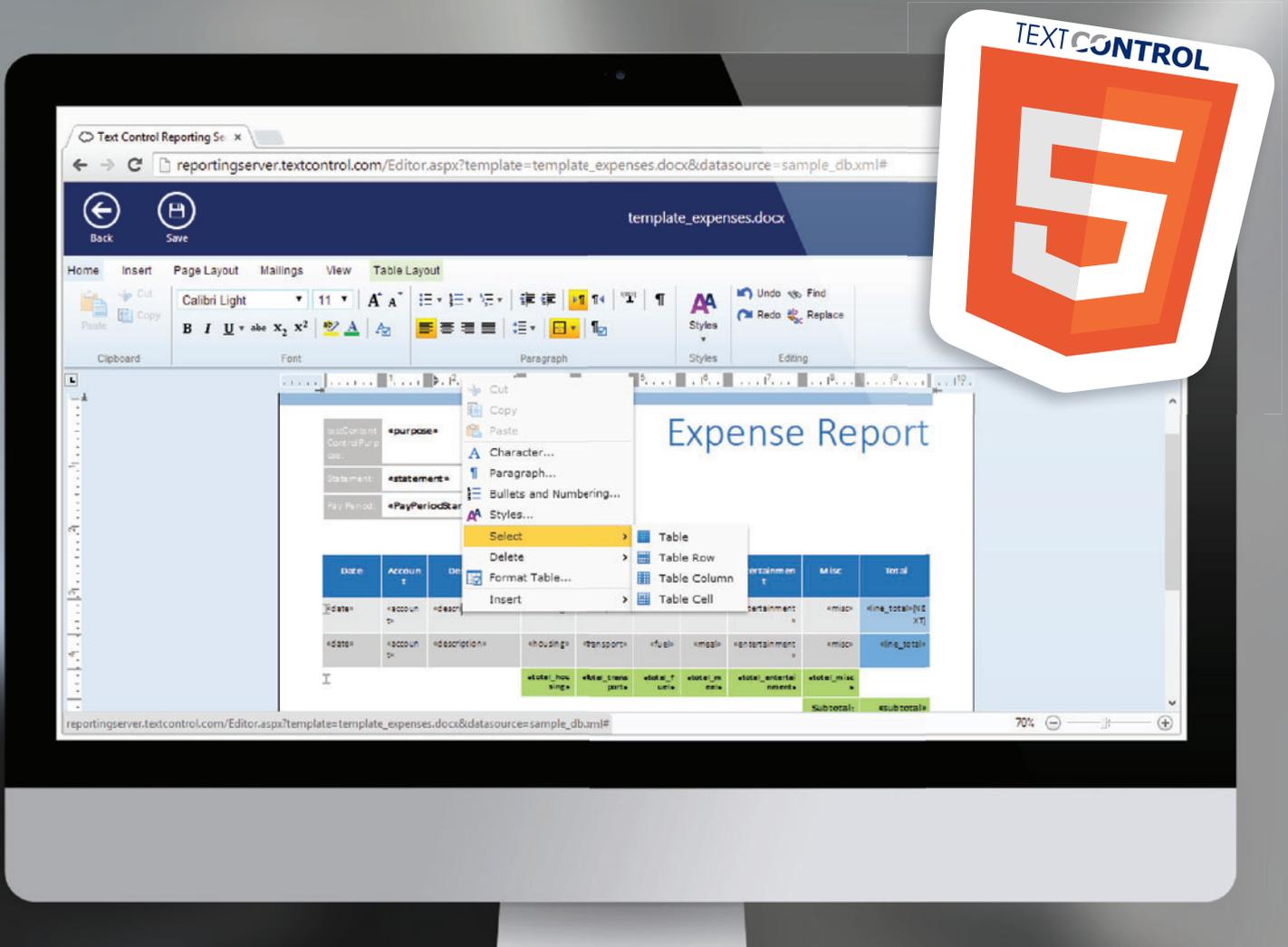
Figure 7 Learning Cycle to Create a Machine Learning-Based Solution

ALISSON SOL is a principal architect for Microsoft. He has many years of software development experience, with a focus on image processing, computer vision, ERP and business intelligence, Big Data, machine learning and distributed systems. Prior to starting at Microsoft in 2000, he cofounded three software companies, published several technical papers and recorded several patent applications. Read his blog at AlissonSol.com/blog.

DON ANKNEY is a senior security researcher in the Microsoft Information Platform Group where he works on applying Microsoft machine learning investments to the services security space. He was a founding member of Black Lodge Research, an educational non-profit focused on security, and regularly teaches secure development techniques at regional meet-ups, workshops and un-conferences.

EUGENE BOBUKH is a senior security/data program manager at Microsoft. Focusing on applying scientific approach to security problems, he conducted security testing and performed security reviews for more than 200 Microsoft releases since 2000, including .NET, Silverlight, and Internet Explorer. Some of Bobukh’s work is described at blogs.msdn.com/b/eugene_bobukh.

THANKS to the following Microsoft technical experts for reviewing this article: Barry Markey and Viresh Ramdatmisier



Cross-browser, cross-platform document and template editing

EDIT MS WORD DOCUMENTS IN ANY BROWSER

The first true WYSIWYG, HTML5-based Web editor and reporting template designer for ASP.NET.



Give your users an MS Word compatible editor to create powerful reporting templates anywhere - in any browser on any device.

Download your 30-day trial version and test a live demo today:

www.textcontrol.com/html5



the new
TEXT CONTROL

All trademarks or registered trademarks are property of their respective owners.



Visual Studio Live!

is hittin' the open road on the ultimate code trip! For 22 years, we have helped tens of thousands of developers navigate the .NET Highway, featuring code-filled days, networking nights and the best in independent training:

- Multi-track Events
- Focused, Cutting-edge .NET Education
- Covering the Hottest Topics
- Relevant and Useable Content
- Expert Speakers, Including Many Microsoft Instructors
- Interactive Sessions
- Discuss Your Challenges
- Find Actionable Solutions

MAKE PLANS TO JOIN US IN 2015!



Redmond

Code Home
August 10 – 14
Microsoft HQ
Redmond, WA

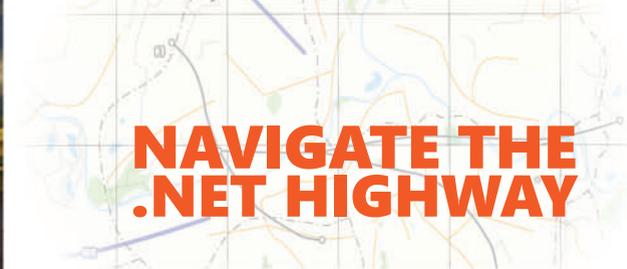
JUST
ADDED!

San Francisco

Code By The Bay
June 15 – 18
The Fairmont
San Francisco, CA

Las Vegas

Code on the Strip
March 16 – 20
Bally's Hotel & Casino
Las Vegas, NV



NAVIGATE THE .NET HIGHWAY

New York

The Code That Never Sleeps
September 28 – October 1
NY Marriott at Brooklyn
Bridge Brooklyn, NY

NOW WITH
6 Locations
to Choose
From!



Austin

Don't Mess with Code
June 1 - 4
Hyatt Regency
Austin, TX



Orlando

Code in the Sun
PART OF LIVE! 360
Tech Events with Perspective

SharePoint Live!
SQL Server Live!
Modern Apps Live!
TechMentor

November 16 – 20
Loews Royal Pacific Resort
Orlando, FL

Occasionally Connected Data in Cross-Platform Mobile Apps

Kevin Ashley

Most mobile apps need to stay offline for some periods of time, and mobile users expect their apps to work gracefully in both connected and disconnected states. Hundreds of millions of mobile device users might not be aware of needs that the apps have with regard to online and offline states; they simply expect the apps to work under any circumstances. In this article I'll show you ways to enable your mobile app to work in both states and synchronize data gracefully with the cloud in Windows, iOS and Android using cross-platform Xamarin tooling and Microsoft Azure Mobile Services.

As a mobile app developer myself, I encountered the need to synchronize offline data early on. For my Winter Sports ski app (winter-sports.co) and Active Fitness activity tracking app (activefitness.co), it's somewhat expected that you can't get a decent connection on the slopes or on the run. So those apps need to be able to synchronize data collected offline, without significant impact on battery life and reliability. In other words, the apps need to work efficiently in any conditions.

There's more than meets the eye when considering persistent storage. To start, there are multiple approaches to synchronizing, meaning it can be done from within an app or as a background

process in the OS. In addition, there are different types of data stores, such as semistructured data from sensors and potentially relational data stored in SQLite. It's also important to implement a conflict resolution policy so data loss and degradation are minimized. Finally, there are a wide variety of data formats to manage, such as binary, JSON, XML and custom.

Mobile apps typically store several types of data. Structured data, such as JSON or XML, is typically used for local settings, local files or caching. In addition to storing data in files, you also have a choice of storage engines, such as SQLite, to store and query data. Mobile apps may also store blobs, media files and other types of large binary data. For these types of data, I'll demonstrate techniques that make data transfer more reliable on occasionally connected devices. I'll provide an overview of several technologies. For example, instead of focusing strictly on offline sync for structured data, I'll show you a broader picture of both structured and unstructured (blob) data synchronization techniques. I'll also use a cross-platform approach everywhere throughout these examples.

The Cross-Platform Approach

Because it's becoming increasingly popular to connect sensor-enabled devices to the cloud, I included device sensor data in my project to demonstrate various methods to sync it with the cloud. I'll discuss three scenarios: offline data sync, manual data synchronization, and synchronizing large media and binary data. The accompanying code sample is completely cross-platform, with 100 percent reusability for Android, iOS and Windows. To achieve that, I used Xamarin.Forms, cross-platform XAML/C# tooling that works well on iOS, Android and Windows and is getting integrated with Visual Studio tools (see the Microsoft Channel 9 video, "Cross-Platform Mobile Development Using Visual Studio," at bit.ly/1xyct02).

In the code sample are two classes that manage cross-platform data models: `SensorDataItem` and `SensorModel`. This approach may be

This article discusses:

- Using cross-platform functionality
- Offline synchronizing of structured data
- Manually synchronizing serialized data
- Synchronizing unstructured data with the cloud

Technologies discussed:

Microsoft Azure Mobile Services, Xamarin

Code download available at:

bit.ly/11yZyhN

used by many sports and fitness tracking apps, such as Active Fitness, or apps that need to synchronize structured data from local storage with the cloud. I added latitude, longitude, speed and distance to the `SensorDataItem` class as an example of data collected by sensors, such as GPS, to illustrate the idea. Of course, the data structure in the actual app might be more complicated—and include dependencies—but my example will give you an idea of the concept.

Synchronizing Structured Data with Offline Sync

Offline sync is a powerful new feature in Azure Mobile Services. You can reference the Azure Mobile Services package in your Visual Studio project using NuGet. More important, it's also supported in cross-platform apps with the new version of the Azure Mobile Services SDK. That means you can use this feature in your Windows, iOS and Android apps that occasionally need to connect to the cloud and synchronize their states.

I'll start with a few concepts.

Sync Table This is a new object in the Azure Mobile Services SDK created to distinguish tables that support synchronization from “local” tables. Sync tables implement the `IMobileServiceSyncTable<T>` interface and include additional “sync” methods, such as `PullAsync`, `PushAsync` and `Purge`. If you want to synchronize your offline sensor data with the cloud, you need to use sync tables instead of standard tables. In my code sample, I initialize my sensor data sync table by using the `GetSyncTable<T>` call. In the Azure Mobile Services portal, I created a regular table called `SensorDataItem` and added the code in **Figure 1** to my client initialization (you can download the full source code from bit.ly/11yZyhN).

The Synchronization Context This is responsible for synchronizing data between local and remote stores. Azure Mobile Services ships with `SQLiteStore`, which is based on the popular `SQLite` library. The code in **Figure 1** does a few things. It checks whether a synchronization context has already been initialized and, if not, it creates a new instance of the `SQLite` store from the local .db file, defines the table based on the `SensorDataItem` class and initializes the store. To handle pending operations, the synchronization context uses a queue, accessible via the `PendingOperations` property. The synchronization context provided by Azure Mobile Services is also “smart” enough to distinguish update operations happening in the local store. Synchronization is done automatically by the system, so you don't have to manually and unnecessarily call into the cloud to persist data. This is good because it lowers traffic and increases device battery life.

The Push Operation This lets you explicitly synchronize data between the local store and the cloud store by pushing local data to the server. It's important to point out that in the current version of the Azure Mobile Services SDK, you need to invoke push and pull explicitly to synchronize the context. The push operation executes on the whole synchronization context to help you preserve relationships between tables. For example, if I have relationships between tables, my first insert will give me an `Id` of the object, and subsequent inserts will preserve referential integrity:

```
await client.SyncContext.PushAsync();
```

The Pull Operation This lets you explicitly synchronize data by pulling data from a remote store into the local store. You can use LINQ

to specify a subset of data, or any OData query. Unlike push operations, which happen on an entire context, pull is executed on the table level. If items are pending in the synchronization queue, those items are pushed first—before pull operations execute—to prevent data loss (yet another benefit from using Azure Mobile Services for data synchronization). In this example, I pull data that has a nonzero speed (collected by my GPS sensor, for example), stored earlier in the server:

```
var query = sensorDataTable.Where(s => s.speed > 0);
await sensorDataTable.PullAsync(query);
```

The Purge Operation This clears data specified from local and remote tables, causing synchronization. Similar to pull, you can use LINQ to specify a subset of data, or any OData query. In this example, I purge data that has zero distance (which might also come from my GPS sensor) from my tables:

```
var query = sensorDataTable.Where(s => s.distance == 0);
await sensorDataTable.PurgeAsync(query);
```

Appropriately Handling Conflicts This is an important part of your data synchronization strategy when devices come online and offline. Conflicts will happen, and the Azure Mobile Services SDK provides ways to deal with conflicts. For conflict resolution to work, I enabled the `Version` property column on the `SensorDataItem` object. In addition, I created the `ConflictHandler` class, which implements the `IMobileServiceSyncHandler` interface. When you need to resolve a conflict, you have three options: keep the client value, keep the server value or abort the push operation.

Figure 1 Leveraging the MobileServiceSQLiteStore Object for Synchronization

```
// Initialize the client with your app URL and key
client = new MobileServiceClient(applicationURL, applicationKey);
// Create sync table instance
todoTable = client.GetSyncTable<SensorDataItem>();
// Later in code
public async Task InitStoreAsync()
{
    if (!client.SyncContext.IsInitialized)
    {
        var store = new MobileServiceSQLiteStore(syncStorePath);
        store.DefineTable<SensorDataItem>();
        await client.SyncContext.InitializeAsync(store,
            new MobileServiceSyncHandler ());
    }
}
```

Figure 2 Data Structure for Data Synchronization

```
public class SensorDataItem
{
    public string Id { get; set; }

    [Version]
    public string Version { get; set; }

    [JsonProperty]
    public string text { get; set; }

    [JsonProperty]
    public double latitude { get; set; }

    [JsonProperty]
    public double longitude { get; set; }

    [JsonProperty]
    public double distance { get; set; }

    [JsonProperty]
    public double speed { get; set; }
}
```

In my example, check the ConflictHandler class. When it's initialized, in the constructor I set it with one of three conflict resolution policies:

```
public enum ConflictResolutionPolicy
{
    KeepLocal,
    KeepRemote,
    Abort
}
```

Depending on the method, each time a conflict happens, in the ExecuteTableOperationAsync method, I automatically apply my conflict resolution policy. When I initialize my synchronization context, I pass my ConflictHandler class to the synchronization context with my default conflict resolution policy:

```
await client.SyncContext.InitializeAsync(
    store,
    new ConflictHandler(client, ConflictResolutionPolicy.KeepLocal)
);
```

To read more on conflict resolution, see the MSDN sample, "Azure Mobile Services - Handling Conflicts with Offline WP8," at bit.ly/14FmZan and the Azure documentation article, "Handling Conflicts with Offline Data Sync in Mobile Services," at bit.ly/1zA01e0.

Synchronizing Serialized Data Manually

Before Azure Mobile Services started offering offline synchronization, developers had to implement data synchronization manually. So, if you develop an app that occasionally needs to synchronize data and you don't use the Azure Mobile Services offline sync feature, you can manually do it (although I strongly recommend looking at the offline sync feature). You can use either direct object serialization (such as a JSON serializer) into files, or data storing engines such as SQLite. The main difference between the offline sync mechanism and manual synchronization is that you need to do most of the work yourself in the latter. One of the methods

Figure 3 Download Operation (Cross-Platform Code)

```
public static async Task<bool> DownloadFileAsync(
    IFolder folder, string url, string fileName)
{
    // Connect with HTTP
    using (var client = new HttpClient())
    // Begin async download
    using (var response = await client.GetAsync(url))
    {
        // If ok?
        if (response.StatusCode == System.Net.HttpStatusCode.OK)
        {
            // Continue download
            Stream temp = await response.Content.ReadAsStreamAsync();
            // Save to local disk
            IFile file = await folder.CreateFileAsync(fileName,
                CreationCollisionOption.ReplaceExisting);

            using (var fs =
                await file.OpenAsync(PCLStorage.FileAccess.ReadAndWrite))
            {
                // Copy to temp folder
                await temp.CopyToAsync(fs);
                fs.Close();
                return true;
            }
        }
        else
        {
            Debug.WriteLine("NOT FOUND " + url);
            return false;
        }
    }
}
```

to detect whether data has been synchronized is by using the Id property of any object in your data model. For example, see the SensorDataItem class used in my earlier example—note the Id and Version fields—shown in **Figure 2**.

When a record is inserted into a remote database, Azure Mobile Services automatically creates an Id and assigns it to the object, so the Id will be a non-null value when the record has been inserted, and null when the record has never been synchronized with the database:

```
// Manually synchronizing data
if (item.Id == null)
{
    await this.sensorDataTable.InsertAsync(item);
}
```

Synchronizing deletes and updates manually is a far more challenging and involved process and is beyond the scope of this article. If you're looking for a comprehensive synchronization solution, look at the offline sync features of the Azure Mobile Services SDK. Of course, this example is simple compared with real-life scenarios, but if you're looking to implement manual data synchronization, this can give you an idea on where to start. Certainly, because the Azure Mobile Services SDK offers a tested, well-thought-out solution for synchronizing data, I recommend trying the offline sync approach, especially in apps that require a solid, tested method to keep local and remote data in sync.

Transferring Binary Data, Photos and Media to the Cloud

In addition to structured data, apps often need to synchronize unstructured or binary data or files. Think of a mobile photo app or an app that needs to upload a binary file to the cloud, such as a photo or video. Because I explore this subject in a cross-platform context, different platforms have different capabilities. But are they really that different? Synchronizing blob data can be achieved in multiple ways, such as with an in-process service or by leveraging an out-of-process, platform-specific background transfer service. To manage downloads, I also provided a simple TransferQueue class, based on ConcurrentQueue. Every time I need to submit a file for upload or download, I add a new Job object to the queue. This is also a common pattern in the cloud, where you put unfinished work in a queue and then have some other background process reading from the queue and completing the work.

In-Process File Transfer Sometimes you need to transfer files directly from within your app. This is the most obvious way of handling blob transfers, but as mentioned earlier, it has disadvantages. To protect the UX, the OS puts a cap on an app's use of bandwidth and resources. However, this assumes the user is using the app to work. In the case of occasionally disconnected apps, it may not be the best idea. The upside of transferring files directly from an app is that it has full control over data transfer. With full control, an app can take advantage of shared access signature methods to manage uploads and downloads. You can read about the advantages in the Microsoft Azure Storage Team Blog post, "Introducing Table SAS (Shared Access Signature), Queue SAS and Update to Blob SAS," at bit.ly/1t1Sb94. Although not all platforms provide this functionality built in, if you're willing to use a REST-based approach, you can certainly take advantage of SAS keys from Azure Storage Services. The downside to transferring files directly from an app

WPF lives!



➔ **XCEED Business Suite for WPF**

The essential set of WPF controls for all your line-of-business solutions. Includes the industry-leading **Xceed DataGrid for WPF**.
A total of 85 tools!

is twofold. First, you must write more code. Second, the app must be running, potentially draining battery life and limiting the UX. The best solutions leverage some of the elegance of built-in data synchronization techniques.

I provided source code for a basic upload and download operation in a cross-platform Xamarin app in `BlobTransfer.cs` (see accompanying code download). This code should work on iOS, Android and Windows. In order to use platform-independent file storage, I used the PCLStorage NuGet package (Install-Package PCLStorage), which allows me to abstract file operations on iOS, Android and Windows.

To initiate an in-process transfer, I call my `TransferQueue.AddInProcessAsync` method:

```
var ok = await queue.AddInProcessAsync(new Job {
    Id = 1, Url = imageUrl, LocalFile = String.Format("image(0).jpg", 1)});
```

This schedules a typical in-process download operation, defined in the `BlobTransfer` object as shown in **Figure 3**.

Of course, if you wish to upload a file, you can do it in-process by using the method shown in **Figure 4**.

Out-of-Process File Transfer Using OS-Specific Transfer Service Downloading and uploading using built-in file transfer services has many advantages. Most platforms offer a service that can transfer large files (both uploading and downloading) autonomously as a background service. You should leverage such services whenever possible because they run out of process—that is, your app isn't capped on the actual data transfer, which may be fairly expensive in terms of resources consumed. Also, your app doesn't have to stay in memory all the time to transfer files, and the OS usually provides a conflict resolution (retry) mechanism to restart uploads and downloads. Other advantages include less code to write, the app doesn't need to be active (the OS manages its own queue of uploads and downloads) and the app is more memory/resource efficient. The challenge, however, is that this method requires platform-specific implementation: iOS, Windows Phone and so on have their own background transfer implementations.

Conceptually, a reliable upload for files in a mobile app using an OS-specific out-of-process service looks similar to the in-app implementation. But the actual upload/download queue management is outsourced to the OS transfer service. For Windows Phone Store apps and Windows Store apps, developers can use

Figure 4 Upload Operation (Cross-Platform Code)

```
public static async Task UploadFileAsync(
    IFolder folder, string fileName, string fileUrl)
{
    // Connect with HTTP
    using (var client = new HttpClient())
    {
        // Start upload
        var file = await folder.GetFilesAsync(fileName);
        var fileStream = await file.OpenAsync(PCLStorage.FileAccess.Read);
        var content = new StreamContent(fileStream);
        // Define content type for blob
        content.Headers.Add("Content-Type", "application/octet-stream");
        content.Headers.Add("x-ms-blob-type", "BlockBlob");
        using (var uploadResponse = await client.PutAsync(
            new Uri(fileUrl, UriKind.Absolute), content))
        {
            Debug.WriteLine("CLOUD UPLOADED " + fileName);
            return;
        }
    }
}
```

`BackgroundDownloader` and `BackgroundUploader` objects. For iOS 7 and higher, `NSURLSession` provides `CreateDownloadTask` and `CreateUploadTask` methods to initiate downloads and uploads.

Using my earlier example, I now need to call my out-of-process method to invoke a call using an OS-specific background transferring service. In fact, because the service is handled by the OS, I'll schedule 10 downloads to demonstrate the app isn't blocking and the execution is handled by the OS (in this example, I used the iOS background transferring service):

```
for (int i = 0; i < 10; i++)
{
    queue.AddOutProcess(new Job { Id = i, Url = imageUrl,
        LocalFile = String.Format("image(0).jpg", i) });
}
```

For an example of a background transfer service for iOS, check `BackgroundTransferService.cs`. In iOS you first need to initialize a background session with `CreateBackgroundSessionConfiguration` (note this only works on iOS 8 or later):

```
using (var configuration = NSURLSessionConfiguration.
    CreateBackgroundSessionConfiguration(sessionId))
{
    session = NSURLSession.FromConfiguration(configuration);
}
```

Then you can submit a long upload or download operation and the OS will handle it independently from your app:

```
using (var uri = NSURL.FromString(url))
using (var request = NSURLRequest.FromUrl(uri))
{
    downloadTask = session.CreateDownloadTask(request);
    downloadTask.Resume();
}
```

You also need to think of a queuing mechanism to reliably upload and download blobs.

Sample Code and Next Steps

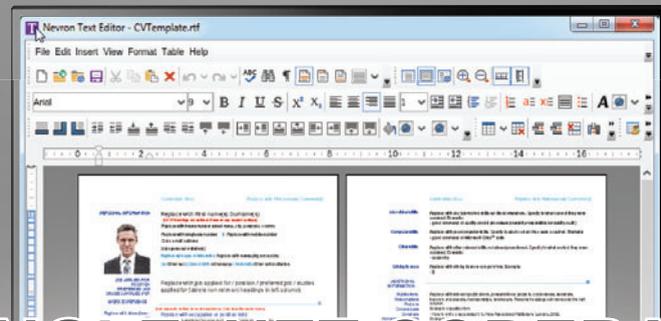
All sample code for this article is available on GitHub at bit.ly/11yZyhN. To use this source code, you can use Visual Studio with Xamarin or Xamarin Studio, which is available from xamarin.com. The project uses cross-platform Xamarin.Forms and the Azure Mobile Services library with offline sync. For the next steps, it would be interesting to see an out-of-process service added to community libraries, such as Xamarin Labs, as well as queuing capabilities and conflict resolution similar to what's currently provided for structured data in the Azure Mobile Services Offline Sync SDK.

To recap, Microsoft Azure Mobile Services provide a powerful and efficient way to synchronize offline data. You can use these services in a cross-platform scenario on Windows, Android and iOS. Microsoft also provides easy-to-use native SDKs that work on every platform. You can improve your app's reliability in disconnected scenarios by integrating these services and adding offline sync to your apps. ■

KEVIN ASHLEY is an architect evangelist at Microsoft. He's a coauthor of "Professional Windows 8 Programming" (Wrox, 2012) and a developer of top apps and games, most notably *Active Fitness* (activefitness.co). He often presents on technology at various events, industry shows and Webcasts. He works with startups and partners, advising on software design, business and technology strategy, architecture, and development. Follow him at his blog at kevinashley.com and on Twitter at twitter.com/kashleytwit.

THANKS to the following Microsoft technical experts for reviewing this article: Greg Oliver and Bruno Terkaly

Write Once, Run Everywhere



SINGLE .NET CODEBASE

Nevron Open Vision is the first suite of .NET controls for building cross-platform user interfaces and applications from a single code base.

Nevron Open Vision includes:

 **NOV WIDGETS** for .NET

 **NOV TEXT EDITOR** for .NET

 **NOV GAUGE** for .NET

 **NOV BARCODE** for .NET

COMING SOON:

 **NOV CHART** for .NET

 **NOV DIAGRAM** for .NET

 **NOV GRID** for .NET

Learn more at www.nevron.com today

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

BYOD Challenge: Connect Mobile Devices to the Enterprise

Greg Oliver and Bruno Terkaly

There are still many concerns surrounding the Bring Your Own Device (BYOD) movement in corporate environments and how Azure Mobile Services can help developers. A core value of Mobile Services is that it democratizes identity and lowers the cost and effort to provide secure access to protected corporate resources. In this article, we'll take a closer look at what it takes to support more complex scenarios when connecting mobile applications to enterprise resources.

We'll start by defining the identity stakeholders and look at the similarities and differences with respect to employees, trading partners and customers. We'll also look at software architectures that provide secure connectivity for browser and mobile device users accessing on-premises services and sites. Later in the article, we'll demonstrate these techniques using an iOS app that connects to a Mobile Services back end, as well as an on-premises service endpoint.

Not All Access Is Equal

There are several important goals of this effort. The first is to provide access to on-premises resources with a minimum of custom code or infrastructure changes, including existing network configurations. Another goal is to manage these usage scenarios to ensure control and visibility, while being agile and flexible.

Not all users who need access behind the corporate firewall are created equal. And not all data behind the corporate firewall needs the same level of protection. Some employees need greater access to protected information. Employees like field salespersons need access to the most up-to-date information on product prices and

inventory levels. They may even need access to accounting data on their customers, such as recent payments on account. They probably don't, however, need access to human resources applications.

Trading partners have their own specialized needs. Trusted business partners will frequently federate identity across an extranet. Trading partners typically manage their own identity infrastructure to authenticate users and will use "claims" to moderate access to a company's protected resources. Thus, either the partner or the company will use trust policy to map the incoming claims to claims understood by a protected resource, such as an internal Web application.

There are two other less common identity stakeholders. Customers often need access to their account information, something like PayPal or Netflix. A prospect is effectively a potential customer. He might have given up credit card information, but hasn't really bought anything yet. All these types of users share one common aspect: They all want to bring and use their iOS, Android and Windows Phone devices to work at the enterprise.

Enterprise Security Management

At least five software stacks play a role in securing mobile devices in the enterprise. Here, we'll take a closer look at Hybrid Connections, Azure Service Bus, Azure Active Directory Application Proxy and Microsoft Azure API Management.

It would be simple to have employees present their credentials to Active Directory through a Web UI, get an access token and start working, but the world of identity is more nuanced and complex. The notion that "one size fits all" is simply not true. Employees are typically forced to use a corporate identity provider.

Most enterprises have brought identity in-house, typically leveraging Active Directory, which mostly exposes an LDAP-based protocol. In this new mobile, cloud-enabled world, enterprises need an HTTP-based communication protocol for mobile devices. Microsoft has created such an identity system in Azure called Azure Active Directory (Azure AD). One of the great features of Azure AD is that enterprises can sync identities to the cloud using directory synchronization. Read more about that process at bit.ly/1ztaB9Z.

This article discusses:

- Supporting complex mobile enterprise apps
- Multiple approaches to mobile identity management
- Managing different categories of contacts

Technologies discussed:

Microsoft Azure, Azure Mobile Services

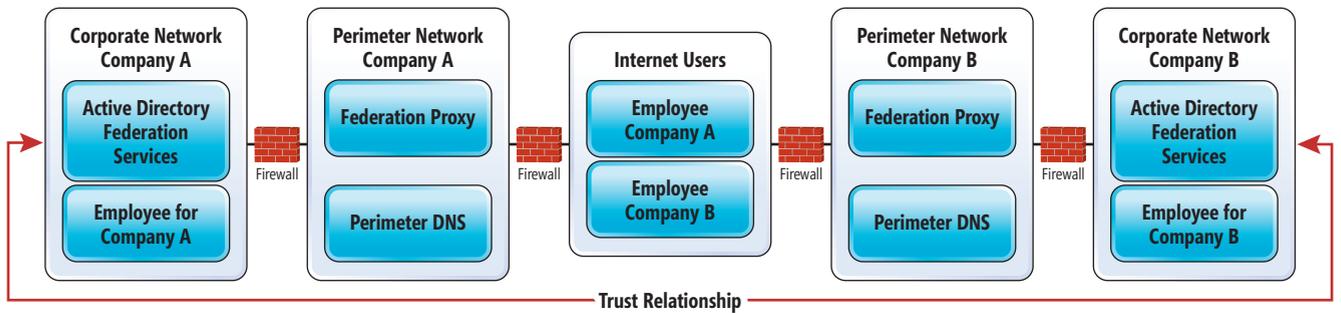


Figure 1 Traditional Identity Model for Active Directory Federation

There are a couple of other nice features in Azure AD. Azure AD provides the convenience of single sign-on (SSO). This lets an employee access protected resources over time without repeatedly being forced to log in and provide credentials. The access token is stored locally as a cookie on the employee's device. You can limit access by using token expiration techniques.

Azure AD also supports OAuth 2.0, perhaps the most popular open standard for authentication. This provides client applications secure delegated access to server resources on behalf of a resource owner. Using OAuth 2.0, Azure AD can authenticate a user and issue either a SAML 2.0 or JWT token. Read more about how tokens work at bit.ly/1pDc0rg.

Trading Partners

The traditional way companies have managed authentication across each other's IT infrastructures has been through federation trust. This involves establishing a trust between the federation services of the two companies (see Figure 1). This supported secure Web-based transactions with business partners. However, it's not as straightforward as it sounds, and can indeed present challenges.

The partner needs to set up a trust relationship between their Active Directory and the other enterprise's Active Directory. This lets an employee from one company authenticate through their own Active Directory and then swap authentication tokens with the other company's Active Directory in order to gain access to an app in the other network. This approach requires a lot of coordination and manual workflows for IT administrators.

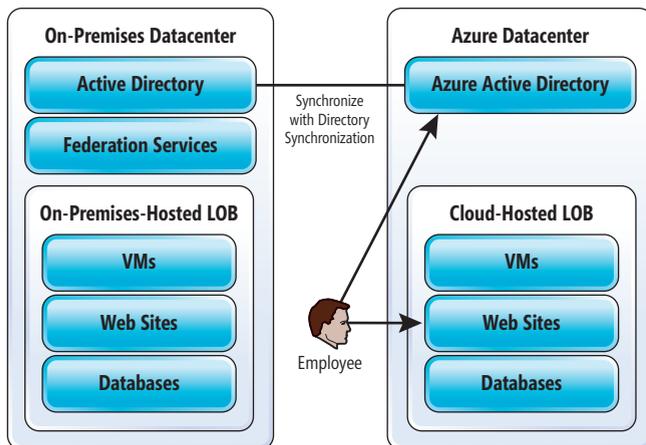


Figure 2 A Modern Approach to Identity Management

Figure 2 depicts a more modern approach to identity management. Generally, a copy of employee identities are hosted in Azure, although some identities may reside in Azure alone. Maintenance is kept at a minimum because the original copy of most identities is managed on-premises. A background synchronization process keeps the cloud copy of identities fully synchronized. Users can connect directly to Azure AD in a datacenter in the cloud and then get redirected to the appropriate application.

For mobile app developers, you'll find a number of libraries on GitHub you can use to take advantage of Azure AD (bit.ly/1qmeipz).

Android, iOS, the Microsoft .NET Framework, JavaScript, Node.js and more are all supported. For example, the iOS sample shows you how to obtain an interactive authorization code to work with a work account. You can tie this work account to an Active Directory server running in your datacenter or live completely in the cloud with Azure AD. On the back end, you could use either a REST-based Node.js API or .NET Web API.

Customer Access

You don't usually think of customer-facing sites and services having access to protected company resources. However, this is more common than you think. For example, Wells Fargo might want to expose a customer's financial data across the Web via a secure Web site or mobile app, while Netflix might want to provide secure access to protected resources such as current account balance or payment history. This becomes especially important for customers using mobile devices.

In some scenarios, it even makes sense to leverage social identity providers, such as Microsoft Account, Facebook, Google or Twitter. This approach was demonstrated in detail in the November Azure Insider article (msdn.microsoft.com/magazine/dn818496), in which we wrote a native iOS application that uses Twitter as the identity provider.

The assumption here is that anyone can download your app or access the Web site. There's a publicly available service tier. Prospects typically have an unauthenticated experience, and see only publicly available information. Even in this case, there may be some need to access on-premises resources. Your service could provide an unauthenticated tier of service or respond to a set of default credentials.

Secure Connectivity Tools

Identity aside, there are several tools that can help securely connect users to resources in enterprise networks. One such example is Azure Service Bus Relay (see Figure 3), which provides an elegant way to

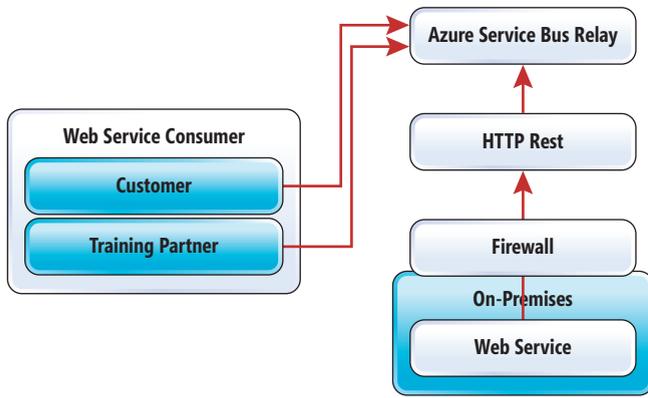


Figure 3 The Azure Service Bus Relay

facilitate access to protected resources. You can think of the Azure Service Bus Relay as a cloud-hosted service acting as a gatekeeper to broker the connection between a client and corporate resources. The Service Bus Relay is authenticated and only opens outbound ports. It supports Topics and Queues, as well, so you could implement a form of distributed messaging. You could even incorporate Event Hubs if you needed to ingest large amounts of events.

The advantage of this approach is there are no configuration changes required to the on-premises networks. That includes provisioning users or changing firewall settings, because both the client and back-end service use outbound connectivity to port 80 or 443. In other words, no additional ports need to be opened to incoming traffic and no proxies are required.

Application Proxy Approach

The second approach to making protected resources available outside the firewall is to use the Azure AD Application Proxy service, currently in Preview. This provides secure access to an organization's internal Web apps and services through HTTP and HTTPS Web protocols.

The big benefit of this approach is users can enjoy SSO and a native application experience and still use an unmanaged, non-domain-joined device. In a previous article, we discussed the concept of a “workplace join,” which is a lightweight version of a fully domain-joined device (bit.ly/1tE7dRR). This supports those BYOD scenarios where users want to retain control of their personal device.

Azure AD Application Proxy provides tight integration with Azure AD, supporting SSO, Software-as-a-Service and on-premises applications, multifactor authentication and device registration. The connector bridges the resource to the application proxy and, therefore, the mobile application. These connectors essentially route traffic and support redundancy, scale and multiple sites.

An Application Proxy provides a few advantages. First, you can implement fine-grained control at an application, device, user and location level. Existing client apps don't need modification and the devices themselves don't need to be modified. It also supports multifactor authentication (MFA), which can provide an extra level of protection for highly confidential resources. MFA provides several supporting services, like fraud alerts, IP whitelists, and mobile phone or SMS as a second authentication factor.

The third advantage of an Application Proxy is it provides isolation between cellular networks and on-premises networks. Back-end services are never directly exposed because the Application Proxy itself lives in Azure AD and the connector reaches out directly from the protected resource. Finally, an Application Proxy protects against denial-of-service attacks, giving you control over throttling, queuing and session management.

API Management

A third approach to providing access to protected resources is Azure API Management, which also behaves as a proxy to your on-premises resources. This lets you expose Web services implemented within your firewall to the outside world. To make this option work, you'll need to make some modifications to your on-premises Web server, as well as create and configure an API in the Azure portal. You can author your API with either .NET Web API or Node.js.

First, we'll consider authenticating with Azure API Management—basic authentication to provide outside access to an on-premises IIS-based Web server. The first step to enabling this scenario is to make some changes to your on-premises IIS Web server and enable basic authentication. Within IIS, you can choose the authorization icon and drill down to basic authentication to add a user. Azure API Management will access on-premises IIS with this user profile.

The next approach you can use with Azure API Management is shared secret authentication. This means the API Management proxy will need to store a secret key. It will place that key in the HTTP header before trying to connect to the on-premises back-end Web service. It should be no surprise the back-end Web service will need to pull the secret key from the HTTP header and process the authorization request.

You can implement shared secret authentication by going to the Azure Management Portal, navigating to the API Management section and selecting Policies from the menu. Here, you'll add a policy, which is nothing more than a secret key sent to your on-premises Web server. The policy definition file that you'll add is an XML document.

Hybrid Connections

The fourth and arguably simplest approach to granting access to protected resources is using Hybrid Connections (see Figure 4), a feature of the Azure BizTalk Services currently in Preview. Hybrid Connections leverages Service Bus Relay technology to create a

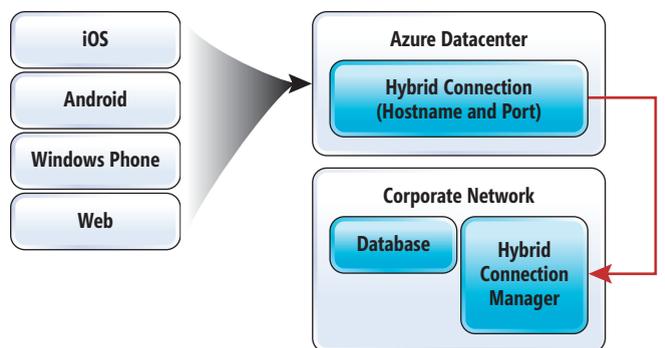


Figure 4 Microsoft Azure BizTalk Services—Hybrid Connections

simple, secure and effective connection between on-premises process and an Azure Mobile Services service or an Azure Web Sites site. You can use Hybrid Connections to connect to any on-premises resource that uses TCP or HTTP for connectivity, including databases and ERP solutions such as SQL, Oracle and SAP.

A requirement for this is the on-premises process must be accessible via a static TCP port, such as SQL Server on port 1433. You can use any framework to connect to the resource, including Node.js, Java or the .NET Framework. Most of the configuration requirements of Service Bus Relay are abstracted away by Hybrid Connections. There are no required changes to the source code of the on-premises process.

To use Hybrid Connections, configure a small amount of metadata in the Azure Management Portal. Install an agent on the server that hosts your process (SQL Server, MySQL, your Web service and so on). There's a Service Bus Relay connection under the covers, using an SAS key for authentication. You can rotate both service-side and client-side keys independently from within the portal. Because it uses Service Bus Relay, no incoming endpoints or proxies are required.

If you prefer more simplicity in your set up and developer efforts, Hybrid Connections might be your solution. Better yet, because it's available in the free tier of Azure BizTalk Services, you can use it cost-effectively, only paying for egress bandwidth.

Figure 5 Client Code Leverages Protected Resources via a Hybrid Connection

```
// Use the SQL Server connection
try
{
    SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder();
    builder["Data Source"] = "myserver";
    builder["Integrated Security"] = false;
    builder["Initial Catalog"] = "mydb";
    builder["User ID"] = "greg";
    builder["Password"] = "S3cr3t";

    int count = 0;
    string query = "SELECT COUNT(*) FROM mytable";
    using (SqlConnection conn = new SqlConnection(builder.ConnectionString))
    using (SqlCommand cmd = new SqlCommand(query, conn))
    {
        conn.Open();
        cmd.CommandType = System.Data.CommandType.Text;
        count = Convert.ToInt32(cmd.ExecuteScalar());
    }
    string results = string.Format("Count of records in mytable is {0}", count);
    Services.Log.Info(results);
}
catch (Exception ex)
{
    Services.Log.Error(ex.Message);
}

// Use the Web service connection
try
{
    const string URL = "http://mydevbox:31106/";
    HttpClient client = new HttpClient();
    client.BaseAddress = new Uri(URL);

    client.DefaultRequestHeaders.Accept.Clear();
    client.DefaultRequestHeaders.Accept.Add(
        new MediaTypeWithQualityHeaderValue("application/json"));

    string response = await client.GetStringAsync("api/values/5");
    Services.Log.Info("Accessing the web service succeeded.");
}
catch (Exception ex)
{
    Services.Log.Error(ex.InnerException.Message);
}
```

Now for a Demonstration

As promised, we'll now demonstrate some of the techniques outlined here, using a native iOS app plus its AMS back end as the platform. We'll add an on-premises Web service and an on-premises SQL Server using Hybrid Connections. In the interests of conserving space and utilizing already-existing resources, please refer to the AMS team's tutorial at bit.ly/1mK1LQF. In the directions you'll find branches to additional tutorial resources (bit.ly/1vFiPuv and bit.ly/1xLWuuF) to build out the complete solution, which consists of the following:

- A Web API back end installed as an AMS service and registered with an Azure AD tenant
- A native iOS app registered with the same Azure AD tenant
- Bi-directional linkage within the tenant between the Web API and the AMS service

There are several steps that involve creating metadata in the Azure Management Portal, while others involve using native development tools (Visual Studio, Xcode, Android and more) to create the actual apps. When complete, you should have a working system that requires you to authenticate into your iOS app by entering Azure AD credentials. Then it will let you do CRUD operations on To Do data. Add the code in **Figure 5** to the mobile service code.

In the connection information for SQL Server and the addressing of the Web service (a simple template Web service), only the name of the Hybrid Connection is involved.

Wrapping Up

Identity is becoming increasingly important in the world of BYOD. Businesses are under increasing pressure to support mobile devices in the workplace. It's made more complex by the fact that we live in a hybrid world. Key resources are located in the cloud and on-premises. Authentication may occur in the cloud or on-premises or both. Protected resources might include business apps, services running on-premises or cloud-hosted Web services. With so many variables, you need multiple flexible approaches.

For full details of this implementation, check the Go Live on Azure blog post at bit.ly/1rVbtCp, and for a Java-based treatment of the topic, bit.ly/1zW7XpZ. ■

GREG OLIVER joined the Microsoft Azure ISV organization in 2010. He spends most of his time helping companies with their migration plans and implementations. Most recently, he has been working with a popular consumer games company with their migration from a competing cloud provider. Prior to joining Microsoft, he was a technology partner in a startup company.

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

THANKS to the following Microsoft technical expert for reviewing this article:
Santosh Chandwani

Understanding TypeScript

Peter Vogel

In many ways, it's useful to think of TypeScript on its own merits. The TypeScript language specification refers to TypeScript as “a syntactic sugar for JavaScript.” That's true and probably an essential step in reaching to the language's target audience—client-side developers currently using JavaScript.

And you do need to understand JavaScript before you can understand TypeScript. In fact, the language specification (you can read it at bit.ly/1xH1m5B) often describes TypeScript constructs in terms of the resulting JavaScript code. But it's equally useful to think of TypeScript as a language on its own that shares features with JavaScript.

For example, like C#, TypeScript is a data-typed language, which gives you IntelliSense support and compile-time checking, among other features. Like C#, TypeScript includes generic and lambda expressions (or their equivalent).

But TypeScript, of course, is not C#. Understanding what's unique about TypeScript is as important as understanding what TypeScript shares with the server-side language you're currently

using. The TypeScript type system is different (and simpler) than C#. TypeScript leverages its understanding of other object models in a unique way and executes inheritance differently than C#. And because TypeScript compiles to JavaScript, TypeScript shares many of its fundamentals with JavaScript, unlike C#.

The question then remains, “Would you rather write your client-side code in this language or in JavaScript?”

TypeScript Is Data-Typed

TypeScript doesn't have many built-in data types you can use to declare variables—just string, number and Boolean. Those three types are a subtype of the any type (which you can also use when declaring variables). You can set or test variables declared with those four types against the types null or undefined. You can also declare methods as void, indicating they don't return a value.

This example declares a variable as string:

```
var name: string;
```

You can extend this simple type system with enumerated values and four kinds of object types: interfaces, classes, arrays and functions. For example, the following code defines an interface (one kind of object type) with the name ICustomerShort. The interface includes two members: a property called Id and a method called CalculateDiscount:

```
interface ICustomerShort
{
    Id: number;
    CalculateDiscount(): number;
}
```

This article discusses:

- Using TypeScript to develop apps
- The similarities and differences of TypeScript and JavaScript
- How TypeScript works with JavaScript

Technologies discussed:

JavaScript, C#, TypeScript

As in C#, you can use interfaces when declaring variables and return types. This example declares the variable `cs` as type `ICustomerShort`:

```
var cs: ICustomerShort;
```

You can also define object types as classes, which, unlike interfaces, can contain executable code. This example defines a class called `CustomerShort` with one property and one method:

```
class CustomerShort
{
  FullName: string;
  UpdateStatus( status: string ): string
  {
    ...manipulate status...
    return status;
  }
}
```

Like more recent versions of C#, it's not necessary to provide implementation code when defining a property. The simple declaration of the name and type is sufficient. Classes can implement one or more interfaces, as shown in **Figure 1**, which adds my `ICustomerShort` interface, with its property, to my `CustomerShort` class.

Like C#, TypeScript is a data-typed language.

As **Figure 1** shows, the syntax for implementing an interface is as simple in TypeScript as in C#. To implement the interface's members you simply add members with the same name instead of tying the interface name to the relevant class' members. In this example, I simply added `Id` and `CalculateDiscount` to the class to implement `ICustomerShort`. TypeScript also lets you use object type literals. This code sets the variable `cst` to an object literal containing one property and one method:

```
var cst1 = {
  Age: 61,
  HaveBirthday(): number
  {
    return this.Age++;
  }
};
```

This example uses an object type to specify the return value of the `UpdateStatus` method:

```
UpdateStatus( status: string ): { status: string; valid: boolean }
{
  return {status: "New",
    valid: true
  };
}
```

Figure 1 Add an Interface to a Class

```
class CustomerShort implements ICustomerShort
{
  Id: number;
  FullName: string;
  UpdateStatus(status: string): string
  {
    ...manipulate status...
    return status;
  }
  CalculateDiscount(): number
  {
    var discAmount: number;
    ...calculate discAmount...
    return discAmount;
  }
}
```

Figure 2 This Class Implements the Proper Interface

```
class CustomerShort implements ICustomerShort
{
  Id: number;
  FullName: string;
  CalculateDiscount( discountedAmount:
    ( discountClass: string, multipleDiscounts: boolean ) => number ): number
  {
    var discAmount: number;
    ...calculate discAmount...
    return discAmount;
  }
}
```

Besides object types (class, interface, literal and array), you can also define function types that describe a function's signature. The following code rewrites `CalculateDiscount` from my `CustomerShort` class to accept a single parameter called `discountAmount`:

```
interface ICustomerShort
{
  Id: number;
  CalculateDiscount( discountAmount:
    ( discountClass: string, multipleDiscount: boolean ) => number ): number
}
```

That parameter is defined using a function type that accepts two parameters (one of string, one of boolean) and returns a number. If you're a C# developer, you might find that the syntax looks much like a lambda expression.

A class that implements this interface would look something like **Figure 2**.

Like the recent versions of C#, TypeScript also infers the data-type of a variable from the value to which the variable is initialized. In this example, TypeScript will assume the variable `myCust` is of type `CustomerShort`:

```
var myCust= new CustomerShort();
myCust.FullName = "Peter Vogel";
```

Like C#, you can declare variables using an interface and then set the variable to an object that implements that interface:

```
var cs: ICustomerShort;
cs = new CustomerShort();
cs.Id = 11;
cs.FullName = "Peter Vogel";
```

Finally, you can also use type parameters (which look suspiciously like generics in C#) to let the invoking code specify the data type to be used. This example lets the code that creates the class set the datatype of the `Id` property:

```
class CustomerTyped<T>
{
  Id: T;
}
```

This code sets the datatype of the `Id` property to a string before using it:

```
var cst: CustomerTyped<string>;
cst = new CustomerTyped<string>();
cst.Id = "A123";
```

To isolate classes, interfaces and other public members and avoid name collisions, you can declare these constructs inside modules much like C# namespaces. You'll have to flag those items you want to make available to other modules with the `export` keyword. The module in **Figure 3** exports two interfaces and a class.

To use the exported components, you can prefix the component name with the module name as in this example:

```
var cs: TypeScriptSample.CustomerShort;
```

Or you can use the TypeScript import keyword to establish a shortcut to the module:

```
import tss = TypeScriptSample;
...
var cs:tss.CustomerShort;
```

TypeScript Is Flexible About Data Typing

All this should look familiar if you're a C# programmer, except perhaps the reversal of variable declarations (variable name first, data type second) and object literals. However, virtually all data typing in TypeScript is optional. The specification describes the data types as "annotations." If you omit data types (and TypeScript doesn't infer the data type), data types default to the any type.

TypeScript doesn't require strict datatype matching, either. TypeScript uses what the specification calls "structural subtyping" to determine compatibility. This is similar to what's often called "duck typing." In TypeScript, two classes are considered identical if they have members with the same types. For example, here's a CustomerShort class that implements an interface called ICustomerShort:

```
interface ICustomerShort
{
  Id: number;
  FullName: string;
}

class CustomerShort implements ICustomerShort
{
  Id: number;
  FullName: string;
}
```

Here's a class called CustomerDeviant that looks similar to my CustomerShort class:

```
class CustomerDeviant
{
  Id: number;
  FullName: string;
}
```

Virtually all data typing in
TypeScript is optional.

Thanks to structural subtyping, I can use CustomerDeviant with variables defined with my CustomerShort class or ICustomerShort interface. These examples use CustomerDeviant interchangeably with variables declared as CustomerShort or ICustomerShort:

```
var cs: CustomerShort;
cs = new CustomerDeviant();
cs.Id = 11;

var csi: ICustomerShort;
csi = new CustomerDeviant();
csi.FullName = "Peter Vogel";
```

This flexibility lets you assign TypeScript object literals to variables declared as classes or interfaces, provided they're structurally compatible, as they are here:

```
var cs: CustomerShort;
cs = {Id: 2,
      FullName: "Peter Vogel"
};

var csi: ICustomerShort;
csi = {Id: 2,
      FullName: "Peter Vogel"
};
```

Figure 3 Export Two Interfaces and One Class

```
module TypeScriptSample
{
  export interface ICustomerDTO
  {
    Id: number;
  }
  export interface ICustomerShort extends ICustomerDTO
  {
    FullName: string;
  }
  export class CustomerShort implements ICustomerShort
  {
    Id: number;
    FullName: string;
  }
}
```

This leads into TypeScript-specific features around apparent types, supertypes and subtypes leading to the general issue of assignability, which I'll skip here. Those features would allow CustomerDeviant, for example, to have members that aren't present in CustomerShort without causing my sample code to fail.

TypeScript Has Class

The TypeScript specification refers to the language as implementing "the class pattern [using] prototype chains to implement many variations on object-oriented inheritance mechanisms." In practice, it means TypeScript isn't only data-typed, but effectively object-oriented.

In the same way that a C# interface can inherit from a base interface, a TypeScript interface can extend another interface—even if that other interface is defined in a different module. This example extends the ICustomerShort interface to create a new interface called ICustomerLong:

```
interface ICustomerShort
{
  Id: number;
}

interface ICustomerLong extends ICustomerShort
{
  FullName: string;
}
```

The ICustomerLong interface will have two members: FullName and Id. In the merged interface, the members from the interface appear first. Therefore, my ICustomerLong interface is equivalent to this interface:

```
interface ICustomerLongPseudo
{
  FullName: string;
  Id: number;
}
```

A class that implements ICustomerLong would need both properties:

```
class CustomerLong implements ICustomerLong
{
  Id: number;
  FullName: string;
}
```

Classes can extend other classes in the same way one interface can extend another. The class in **Figure 4** extends CustomerShort and adds a new property to the definition. It uses explicit getters and setters to define the properties (although not in a particularly useful way).

TypeScript enforces the best practice of accessing internal fields (like id and fullName) through a reference to the class (this). Classes

can also have constructor functions that include a feature C# has just adopted: automatic definition of fields. The constructor function in a TypeScript class must be named `constructor` and its public parameters are automatically defined as properties and initialized from the values passed to them. In this example, the constructor accepts a single parameter called `Company` of type `string`:

```
export class CustomerShort implements ICustomerShort
{
  constructor(public Company: string)
  { }
}
```

Because the `Company` parameter is defined as `public`, the class also gets a public property called `Company` initialized from the value passed to the constructor. Thanks to that feature, the variable `comp` will be set to “PH&VIS,” as in this example:

```
var css: CustomerShort;
css = new CustomerShort( "PH&VIS" );
var comp = css.Company;
```

Declaring a constructor’s parameter as `private` creates an internal property it can only be accessed from code inside members of the class through the keyword `this`. If the parameter isn’t declared as `public` or `private`, no property is generated.

TypeScript isn’t only data-typed, but effectively object-oriented.

Your class must have a constructor. As in C#, if you don’t provide one, one will be provided for you. If your class extends another class, any constructor you create must include a call to `super`. This calls the constructor on the class it’s extending. This example includes a constructor with a `super` call that provides parameters to the base class’ constructor:

```
class MyBaseClass
{
  constructor(public x: number, public y: number ) { }
}
class MyDerivedClass extends MyBaseClass
{
  constructor()
  {
    super(2,1);
  }
}
```

TypeScript Inherits Differently

Again, this will all look familiar to you if you’re a C# programmer, except for some funny keywords (`extends`). But, again, extending a class or an interface isn’t quite the same thing as the inheritance mechanisms in C#. The TypeScript specification uses the usual terms for the class being extended (“base class”) and the class that extends it (“derived class”). However, the specification refers to a class’ “heritage specification,” for example, instead of using the word “inheritance.”

To begin with, TypeScript has fewer options than C# when it comes to defining base classes. You can’t declare the class or members as non-overrideable, abstract or virtual (though interfaces provide much of the functionality that a virtual base class provides).

There’s no way to prevent some members from not being inherited. A derived class inherits all members of the base class, including public

Figure 4 Properties Defined with Getters and Setters

```
class CustomerShort
{
  Id: number;
}

class CustomerLong extends CustomerLong
{
  private id: number;
  private fullName: string;

  get Id(): number
  {
    return this.id
  }
  set Id( value: number )
  {
    this.id = value;
  }
  get FullName(): string
  {
    return this.fullName;
  }
  set FullName( value: string )
  {
    this.fullName = value;
  }
}
```

and private members (all public members of the base class are overrideable while private members are not). To override a public member, simply define a member in the derived class with the same signature. While you can use the `super` keyword to access a public method from a derived class, you can’t access a property in the base class using `super` (though you can override the property).

TypeScript lets you augment an interface by simply declaring an interface with an identical name and new members. This lets you extend existing JavaScript code without creating a new named type. The example in **Figure 5** defines the `ICustomerMerge` interface through two separate interface definitions and then implements the interface in a class.

Classes can also extend other classes, but not interfaces. In TypeScript, interfaces can also extend classes, but only in a way that involves inheritance. When an interface extends a class, the interface includes all class members (public and private), but without the class’ implementations. In **Figure 6**, the `ICustomer` interface will have the private member `id`, public member `Id` and the public member `MiddleName`.

The `ICustomer` interface has a significant restriction—you can only use it with classes that extend the same class the interface

Figure 5 The `ICustomerMerge` Interface Defined Through Two Interface Definitions

```
interface ICustomerMerge
{
  MiddleName: string;
}
interface ICustomerMerge
{
  Id: number;
}

class CustomerMerge implements ICustomerMerge
{
  Id: number;
  MiddleName: string;
}
```

extended (in this case, that's the Customer class). TypeScript requires that you include private members in the interface to be inherited from the class that the interface extends, instead of being reimplemented in the derived class. A new class that uses the ICustomer interface would need, for example, to provide an implementation for MiddleName (because it's only specified in the interface). The developer using ICustomer could choose to either inherit or override public methods from the Customer class, but wouldn't be able to override the private id member.

Classes can extend other classes in the same way one interface can extend another.

This example shows a class (called NewCustomer) that implements the ICustomer interface and extends the Customer class as required. In this example, NewCustomer inherits the implementation of Id from Customer and provides an implementation for MiddleName:

```
class NewCustomer extends Customer implements ICustomer
{
  MiddleName: string;
}
```

This combination of interfaces, classes, implementation and extension provides a controlled way for classes you define to extend classes defined in other object models (for more details, check out section 7.3 of the language specification, "Interfaces Extending Classes"). Coupled with the ability of TypeScript to use information about other JavaScript libraries, it lets you write TypeScript code that works with the objects defined in those libraries.

TypeScript Knows About Your Libraries

Besides knowing about the classes and interfaces defined in your application, you can provide TypeScript with information about other object libraries. That's handled through the TypeScript declare keyword. This creates what the specification calls "ambient declarations." You may never have to use the declare keyword yourself because you can find definition files for most JavaScript libraries

Figure 6 An Extended Class with All Members

```
class Customer
{
  private id: number;

  get Id(): number
  {
    return this.id
  }
  set Id( value: number )
  {
    this.id = value;
  }
}

interface ICustomer extends Customer
{
  MiddleName: string;
}
```

on the DefinitelyTyped site at definitelytyped.org. Through these definition files, TypeScript can effectively "read the documentation" about the libraries with which you need to work.

"Reading the documentation," of course, means you get data-typed IntelliSense support and compile-time checking when using the objects that make up the library. It also lets TypeScript, under certain circumstances, infer the type of a variable from the context in which it's used. Thanks to the lib.d.ts definition file included with TypeScript, TypeScript assumes the variable anchor is of type HTMLAnchorElement in the following code:

```
var anchor = document.createElement( "a" );
```

The definition file specifies that's the result returned by the createElement method when the method is passed the string "a." Knowing anchor is an HTMLAnchorElement means TypeScript knows the anchor variable will support, for example, the addEventListener method.

The TypeScript data type inference also works with parameter types. For example, the addEventListener method accepts two parameters. The second is a function in which addEventListener passes an object of type PointerEvent. TypeScript knows that and supports accessing the cancelBubble property of the PointerEvent class within the function:

```
span.addEventListener("pointerenter", function ( e )
{
  e.cancelBubble = true;
})
```

In the same way that lib.d.ts provides information about the HTML DOM, the definition files for other JavaScript provide similar functionality. After adding the backbone.d.ts file to my project, for example, I can declare a class that extends the Backbone Model class and implements my own interface with code like this:

```
class CustomerShort extends bb.Model implements ICustomerShort
{
}
```

If you're interested in details on how to use TypeScript with Backbone and Knockout, check out my Practical TypeScript columns at bit.ly/1BRh8NJ. In the new year, I'll be looking at the details of using TypeScript with Angular.

There's more to TypeScript than you see here. TypeScript version 1.3 is slated to include union datatypes (to support, for example, functions that return a list of specific types) and tuples. The TypeScript team is working with other teams applying data typing to JavaScript (Flow and Angular) to ensure TypeScript will work with as broad a range of JavaScript libraries as possible.

If you need to do something that JavaScript supports and TypeScript won't let you do, you can always integrate your JavaScript code because TypeScript is a superset of JavaScript. So the question remains—which of these languages would you prefer to use to write your client-side code? ■

PETER VOGEL is a principal with PH&V Information Services, specializing in Web development with expertise in SOA, client-side development and UI design. PH&V clients include the Canadian Imperial Bank of Commerce, Volvo and Microsoft. He also teaches and writes courses for Learning Tree International and writes the Practical .NET column for VisualStudioMagazine.com.

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

DEVELOPED FOR INTUITIVE USE

DynamicPDF—Comprehensive PDF Solutions for .NET Developers

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



DynamicPDF

WWW.DYNAMICPDF.COM



TRY OUR PDF SOLUTIONS FREE TODAY!

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

ceTe software

Adding Application Data to the Windows Phone 8.1 Map Control

Keith Pijanowski

In my November article on the mapping capabilities of Windows Phone 8.1, I introduced the Windows Phone 8.1 Map control and the Mapping Services API. I showed how to use the Map control to add mapping functionality to a phone application, for example, adding images to mark a specified location, calculating the geo-coordinates of a specified address (geocoding), determining the address of a specified geo-coordinate (reverse geocoding), and providing driving and walking directions.

This article will show how to get application data into the Map control by adding controls to the Map control using XAML, and binding application data to those controls. If your application will be used offline, as well, the underlying data that's used by the Map control

can also be downloaded for offline use. This allows the map control to fit in with the offline functionality of your app. I'll also show how to take the Map control offline and keep the offline data up-to-date.

Adding Controls to the Map Control Using XAML

In my previous article, I used code to add an ellipse control to a map in order to circle a specific location on the map. That's a good approach if you need complete control. However, if you wish to get creative and annotate a map with many different types of controls, using code can be clunky. Adding controls to the map using XAML is more efficient. Also, the XAML approach is much easier when you need to map a collection of locations within a map using data binding.

First I'll show how to add controls using XAML and how to bind application data to these controls. Then I'll delve further into data binding and show how to data bind a collection of locations to the Map control.

Adding Basic XAML Controls There are two ways to add controls to the Map control in order to add application data to a map: You can add the controls as children of the Map control or you can use the MapItemsControl to contain the controls.

Figure 1 shows how to add controls as children of the Map control. Because Children is the default content property of the Map control, it doesn't need to be explicitly specified in the XAML markup. **Figure 2** shows an alternate approach to adding controls to the Map control, which makes use of the MapItemControl to contain added controls.

This article discusses:

- Adding controls to the Map control using XAML
- Binding application data to controls
- Scaling controls
- Downloading maps for offline use
- Updating downloaded maps

Technologies discussed:

Windows Phone 8.1, Visual Studio 2013 Update 3

Code download available at:

msdn.microsoft.com/magazine/msdnmag0115

If you use the `MapItemControl` technique shown in **Figure 2**, be aware you won't be able to access the controls in your codebehind. IntelliSense will acknowledge your control names as valid variables, but at run time these variables will always be null and if you reference them you'll get a `NullReferenceException`. The `MapItemControl` is typically used to contain data templates that are used for binding a collection of objects to the Map control. (This will be discussed later in this article.) Therefore, if you're not binding to a collection of objects, it's better to add your controls as children of the Map control, as shown in **Figure 1**.

Data Binding The code in **Figure 1** makes use of data binding to set the `Location` property of both the image control and the text block. The `Text` property of the `TextBlock` also utilizes data binding. To quickly review, the `Location` property is an attached property of type `Geopoint`. It's used to specify where on the map the control will be placed. The `NormalizedAnchorPoint` attached property allows the location of the control to be fine-tuned. For example, you can use the `NormalizedAnchorPoint` property to center a control over a location, or place a control on the upper left of a location. Both of these properties were discussed in detail in my first article (msdn.microsoft.com/magazine/dn818495).

Figure 1 Adding Controls as Children of the Map Control

```
<Maps:MapControl
  x:Name="myMapControl" Grid.Row="1"
  MapServiceToken="{StaticResource MapServiceTokenString}" >

  <!-- Progress bar which is used while the page is loading. -->
  <ProgressBar Name="pbProgressBar" IsIndeterminate="True" Height="560"
    Width="350" />

  <TextBlock Name="tbMessage" Text="{Binding Message}"
    Maps:MapControl.Location="{Binding Location}"
    Maps:MapControl.NormalizedAnchorPoint="1.0,1.0"
    FontSize="15" Foreground="Black" FontWeight="SemiBold"
    Padding="4,4,4,4"
    Visibility="Collapsed"
  />

  <Image Name="imgMyLocation" Source="Assets/PinkPushPin.png"
    Maps:MapControl.Location="{Binding Location}"
    Maps:MapControl.NormalizedAnchorPoint="0.25, 0.9"
    Height="30" Width="30"
    Visibility="Collapsed" />

</Maps:MapControl>
```

Figure 2 Containing Controls Within a MapItemControl

```
<Maps:MapControl
  x:Name="myMapControl"
  MapServiceToken="{StaticResource MapServiceTokenString}"

  <Maps:MapItemsControl>

  <TextBlock Name="tbAddress" Text="{Binding Message}"
    Maps:MapControl.Location="{Binding Location}"
    Maps:MapControl.NormalizedAnchorPoint="1.0,1.0"
    Visibility="Collapsed" />

  <Image Name="imgMyLocation" Source="Assets/PinkPushPin.png"
    Maps:MapControl.Location="{Binding Location}"
    Maps:MapControl.NormalizedAnchorPoint="0.25, 0.9"
    Height="30" Width="30"
    Visibility="Collapsed"/>

  </Maps:MapItemsControl>

</Maps:MapControl>
```

Data binding allows XAML controls to get their values from underlying objects. An object created from the following class will be used to hold the values needed for the XAML controls shown in **Figure 1**:

```
public class LocationEntity
{
  public Geopoint Location { get; set; }
  public string Message { get; set; }
}
```

Data binding allows XAML controls to get their values from underlying objects.

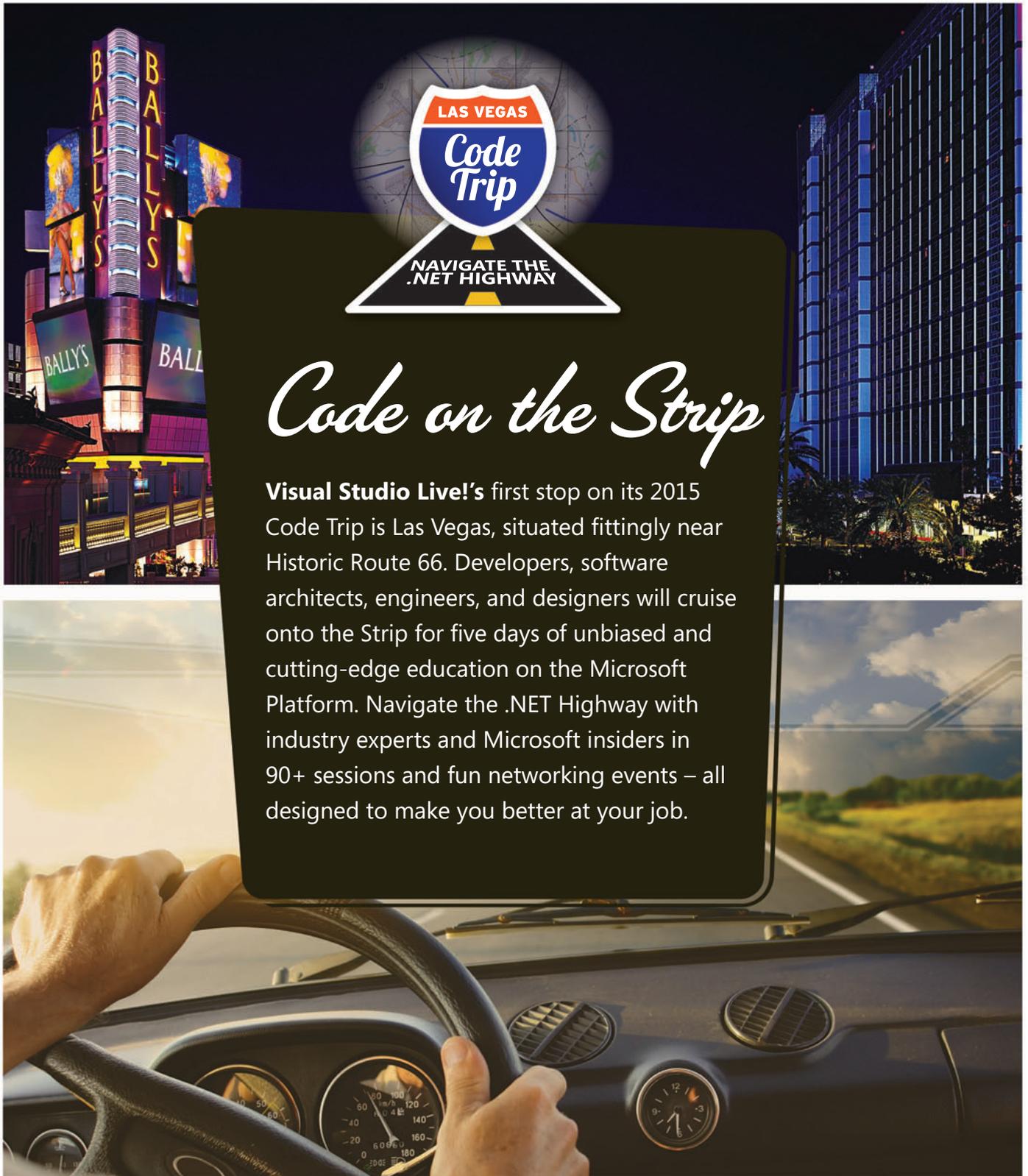
Figure 3 shows the complete `OnNavigatedTo` event for a page containing the controls in **Figure 1**. (If a page or view within an application requires a lot of setup, consider placing this code in the creation of a view model and running it asynchronously.) This code sets the device's current location into an instance of the `LocationEntity` class along with a short message. Notice that the `LocationEntity` object is set into the `DataContext` property of the Map control.

A Few Tips

In the code shown in **Figure 3**, the `OnNavigatedTo` event performs time-consuming work. Specifically, this event calls the `GetGeopositionAsync` function of a `Geolocator` object. This function gets the device's current location, which is needed before the image control and the `TextBlock` control can be properly positioned on the map. In situations like this, any controls added to the map as children via XAML are initially displayed in the upper-left corner of the map until they can be positioned on the map. This creates a poor UX. To correct this, simply set up any added controls as `Collapsed` within your XAML. The added controls can be made visible once the control locations are calculated and the object in which the controls are bound is set up. (Refer to **Figure 1** and the last few lines of code in **Figure 3**.)

Consider using the `DesiredAccuracyInMeters` property of the `Geolocator` object. Setting this property to 100 meters or less will get the most accurate data available. If the device has GPS capabilities, the GPS will then be used to determine the device's current location. If this value is greater than 100 meters, the `Geolocator` object will optimize for power and use non-GPS data, such as Wi-Fi signals. The 100 meter threshold could change as devices evolve, so set this property based on the needs of the application and not the underlying threshold, which triggers different behavior. Finally, the `DesiredAccuracyInMeters` property overrides anything set in the `Geolocator's DesiredAccuracy` property.

Also, consider using the overload of `GetGeopositionAsync`, which accepts a `maximumAge` parameter and a `timeout` parameter. The `maximum age` parameter is a `TimeSpan` that specifies the maximum acceptable age of cached location data. The `timeout` parameter is also a `TimeSpan` and will cause the `GetGeopositionAsync` function to throw an exception if determining the current location takes longer than specified.



Code on the Strip

Visual Studio Live!'s first stop on its 2015 Code Trip is Las Vegas, situated fittingly near Historic Route 66. Developers, software architects, engineers, and designers will cruise onto the Strip for five days of unbiased and cutting-edge education on the Microsoft Platform. Navigate the .NET Highway with industry experts and Microsoft insiders in 90+ sessions and fun networking events – all designed to make you better at your job.



NAVIGATE THE .NET HIGHWAY

TRACKS INCLUDE:

- Visual Studio / .NET Framework
- JavaScript / HTML5 Client
- Modern App Development
- ASP.NET
- Cross-Platform Mobile Development
- Database and Analytics
- Windows Client
- Cloud Computing
- Windows Phone
- SharePoint Cloud Business Apps



*Register NOW and
Save \$400!*

Use promo code VSLJAN4



CONNECT WITH VISUAL STUDIO LIVE!



twitter.com/vslive – @VSLive



facebook.com – Search “VSLive”



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



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

TURN THE PAGE FOR MORE EVENT DETAILS



vslive.com/lasvegas

Bally's Hotel & Casino

will play host to Visual Studio Live!, and is offering a special reduced room rate to conference attendees.



CONNECT WITH VISUAL STUDIO LIVE!

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

 [facebook.com](https://www.facebook.com/vslive) – Search “VSLive”

 [linkedin.com](https://www.linkedin.com/groups?gid=11111111) – Join the “Visual Studio Live” group!



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

AGENDA AT-A-GLANCE

ASP.NET		Cloud Computing	Cross-Platform Mobile Development	JavaScript / HTML5 Client
Visual Studio Live! Pre-Conference Workshops: Monday, March 16, 2015				
7:30 AM	9:00 AM	Pre-Conference Workshop Registration - Coffee and Morning Pastries		
9:00 AM	6:00 PM	M01 - Workshop: Native Mobile App Development for iOS, Android and Windows Using C# - <i>Marcel de Vries & Rockford Lhotka</i>		
6:45 PM	9:00 PM	Dine-A-Round		
Visual Studio Live! Day 1: Tuesday, March 17, 2015				
7:00 AM	8:00 AM	Registration - Coffee and Morning Pastries		
8:00 AM	9:00 AM	Keynote: To Be Announced		
9:15 AM	10:30 AM	T01 - An Overview of the Xamarin Programming Platforms - <i>Laurent Bugnion</i>	T02 - JavaScript for the C# (and Java) Developer - <i>Philip Japikse</i>	
10:45 AM	12:00 PM	T07 - Building Cross-Platform Applications with XAML, Xamarin, Xamarin.Forms and MVVM Light - <i>Laurent Bugnion</i>	T08 - Write Next Generation, Object Oriented JavaScript, with TypeScript. - <i>Rachel Appel</i>	
12:00 PM	1:30 PM	Lunch - Visit Exhibitors		
1:30 PM	2:45 PM	T13 - Microsoft - To Be Announced	T14 - It's the Top 10 of Cool HTML5 Features Every Developer Should Know Right Now - <i>Gill Cleeren</i>	
3:00 PM	4:15 PM	T19 - Creating Responsive Cross-Platform Native/ Web Apps with JavaScript and Bootstrap - <i>Ben Dewey</i>	T20 - I Just Met You, and "This" is Crazy, But Here's My NaN, So Call (me) Maybe? - <i>Rachel Appel</i>	
4:15 PM	5:30 PM	Welcome Reception		
Visual Studio Live! Day 2: Wednesday, March 18, 2015				
7:00 AM	8:00 AM	Registration - Coffee and Morning Pastries		
8:00 AM	9:00 AM	Keynote: To Be Announced - <i>Rockford Lhotka, CTO, Magenic</i>		
9:15 AM	10:30 AM	W01 - Cross-Platform Mobile Apps Using AngularJS and the Ionic Framework - <i>Ben Dewey</i>	W02 - Getting Started with jQuery and How it Works Together with ASP.NET WebForms and MVC - <i>Gill Cleeren</i>	
10:45 AM	12:00 PM	W07 - Building Cross-Platform Apps in C# - <i>Rockford Lhotka</i>	W08 - Build Data-Centric HTML5 Single Page Applications with Breeze - <i>Brian Noyes</i>	
12:00 PM	1:30 PM	Birds-of-a-Feather Lunch - Visit Exhibitors		
1:30 PM	2:45 PM	W13 - Building Your First Universal Application for Windows Phone and Windows Store - <i>Brian Peek</i>	W14 - Busy Developer's Guide to MEANJS - <i>Ted Neward</i>	
3:00 PM	4:15 PM	W19 - Getting Started with Windows Phone 8.1 Development - <i>Nick Landry</i>	W20 - Creating Angular Applications Using Visual Studio LightSwitch - <i>Michael Washington</i>	
4:30 PM	5:45 PM	W25 - Strike Up a Conversation with Cortana on Windows Phone - <i>Walt Ritscher</i>	W26 - JavaScript Patterns for the C# Developer - <i>Ben Hoelting</i>	
7:00 PM	9:00 PM	Visual Studio Live! Evening Event		
Visual Studio Live! Day 3: Thursday, March 19, 2015				
7:00 AM	8:00 AM	Registration - Coffee and Morning Pastries		
8:00 AM	9:15 AM	TH01 - From WPF to Universal Apps: A Guide Forward - <i>Rockford Lhotka</i>	TH02 - AngularJS for ASP.NET MVC Developers - <i>Miguel Castro</i>	
9:30 AM	10:45 AM	TH07 - UX Design Principle Fundamentals for Non-Designers - <i>Billy Hollis</i>	TH08 - Securing Angular Apps - <i>Brian Noyes</i>	
11:00 AM	12:15 PM	TH13 - XAML: Achieving Your Moment of Clarity - <i>Miguel Castro</i>	TH14 - Unit Testing JavaScript Applications - <i>Ben Dewey</i>	
12:15 PM	1:30 PM	Lunch		
1:30 PM	2:45 PM	TH19 - Tricks and Shortcuts for Amazing UI in XAML - <i>Billy Hollis</i>	TH20 - Building Single Page Web Applications Using JavaScript and the MVVM Pattern - <i>Ben Hoelting</i>	
3:00 PM	4:15 PM	TH25 - WPF Data Binding in Depth - <i>Brian Noyes</i>	TH26 - Busy Developer's Guide to SignalR and WebSockets - <i>Ted Neward</i>	
Visual Studio Live! Post-Conference Workshops: Friday, March 20, 2015				
7:30 AM	8:00 AM	Post-Conference Workshop Registration - Coffee and Morning Pastries		
8:00 AM	5:00 PM	F01 - Workshop: Hybrid Mobile Apps with Cordova, Angular, and Azure - <i>Brian Noyes</i>		

Speakers and sessions subject to change

Modern Apps LIVE!

MOBILE, CROSS-DEVICE & CLOUD DEVELOPMENT

Presented in partnership with **Magenic**

BONUS CONTENT! Modern Apps Live! is now a part of Visual Studio Live! Las Vegas at no additional cost!

NAVIGATE THE .NET HIGHWAY

SharePoint Cloud Business Apps

Database and Analytics

Visual Studio / .NET Framework

Windows Client

Windows Phone

Modern Apps Live!

(Separate entry fee required)

M02 - Workshop: Deep Dive into Visual Studio 2013, TFS and Visual Studio Online - Brian Randell

M03 - Workshop: UX Design Process Essentials - Steps and Techniques - Billy Hollis

M04 - Workshop: Modern App Technology Overview - Android, iOS, Cloud, and Mobile Web

T03 - What's Up with ASP.NET vNext (Who Moved My Cheese?) - Mark Michaelis

T04 - Cloud or Not, 10 Reasons Why You Must Know "Web Sites" - Vishwas Lele

T05 - New IDE and Editor Features in Visual Studio - Deborah Kurata

T06 - Defining Modern App Development

T09 - Microsoft - To Be Announced

T10 - Azure 10-10: Top 10 Azure Announcements in "T-10" Months - Vishwas Lele

T11 - Visual Studio Online: An Update Every Three Weeks - Brian Randell

T12 - Modern App Architecture

T15 - ASP.NET MVC For The WebForms Developer - Philip Japikse

T16 - Busy Developer's Guide to NoSQL - Ted Neward

T17 - Defensive Coding Techniques in C# - Deborah Kurata

T18 - ALM with Visual Studio Online (TFS) and Git

T21 - Slice Development Time With ASP.NET MVC, Visual Studio, and Razor - Philip Japikse

T22 - Busy Developer's Guide to Six Developer Cloud Platforms - Ted Neward

T23 - Automated Cross Browser Testing of Your Web Applications with Visual Studio CodedUI - Marcel de Vries

T24 - Reusing Business and Data Access Logic Across Platforms

W03 - ASP.NET vNext in All Its Glory - Adam Tuliper

W04 - T-SQL for Application Developers - Attendees Choose - Kevin Goff

W05 - Using Microsoft Application Insights to Implement a Build, Measure, Learn Loop - Marcel de Vries

W06 - Coding for Quality and Maintainability

W09 - Understanding the Power of ASP.NET MVC vNext - Ben Hoelting

W10 - SQL Server Reporting Services for Application Developers - Attendees Choose - Kevin Goff

W11 - Build It and Ship It with TFS and Release Management - Brian Randell

W12 - UX Design for Modern Apps

W15 - Improving Web Application Performance - Ben Hoelting

W16 - A Tour of Azure for Developers - Adam Tuliper

W17 - Programming C# 6.0 - Mark Michaelis

W18 - Applied UX: iOS, Android, Windows

W21 - To Be Announced

W22 - 10 Ways to Guarantee Your Azure Project Will Fail! - Michael Collier

W23 - Writing Asynchronous Code using .NET 4.5 and C# 5.0 - Brian Peek

W24 - Leveraging Azure Services

W27 - WCF & Web API: Can We All Just Get Along!?! - Miguel Castro

W28 - Intro to Azure API Management - Michael Collier

W29 - The Busy Developers Guide to Virtualization with Hyper-V - Brian Randell

W30 - Unit Testing Modern Apps

TH03 - Fiddler and Your Website - Robert Boedigheimer

TH04 - SQL Server in Azure Virtual Machines - Eric D. Boyd

TH05 - Managing the .NET Compiler - Jason Bock

TH06 - Analyzing Results with Power BI

TH09 - Web Essentials - Robert Boedigheimer

TH10 - Microsoft Azure SQL Database - SQL Server in the Cloud - Leonard Lobel

TH11 - Real World Scrum with Team Foundation Server 2013 & Visual Studio Online - Benjamin Day

TH12 - Building a Modern Android App with Xamarin

TH15 - Node.js for .NET Developers - Jason Bock

TH16 - Database Development with SQL Server Data Tools - Leonard Lobel

TH17 - Git for the Microsoft Developer - Eric D. Boyd

TH18 - Building an iOS and Android App with Xamarin.Forms

TH21 - Building Rich UI with ASP.NET MVC and Bootstrap - Walt Ritscher

TH22 - Creating Visual Studio Cloud Business Applications for SharePoint 2013 Using AngularJS - Michael Washington

TH23 - To Be Announced

TH24 - Building a Windows App

TH27 - ASP.NET - Rudiments of Routing - Walt Ritscher

TH28 - Cloud Business Apps and SharePoint Workflows - Michael Washington

TH29 - Zero to Hero: Untested to Tested with Visual Studio Fakes - Benjamin Day

TH30 - Building a Responsive Single Page App

(Separate entry fee required)

F02 - Workshop: SQL Server 2014 for Developers - Leonard Lobel

F03 - Workshop: Modern App Development In-Depth: iOS, Android, Windows, and Web

Figure 3 The OnNavigatedTo Event Creating Objects Needed for Data Binding

```
protected async override void OnNavigatedTo(NavigationEventArgs e)
{
    // Call the navigation helper base function.
    this.navigationHelper.OnNavigatedTo(e);

    // Get the user's current location so that it can be
    // used as the center point of the map control.
    Geolocator geolocator = new Geolocator();

    // Set the desired accuracy.
    geolocator.DesiredAccuracyInMeters = 200;

    // The maximum acceptable age of cached location data.
    TimeSpan maxAge = new TimeSpan(0, 2, 0);

    // Timeout used for the call to GetGeopositionAsync.
    TimeSpan timeout = new TimeSpan(0, 0, 5);

    Geoposition geoposition = null;
    try
    {
        geoposition = await geolocator.GetGeopositionAsync(maxAge, timeout);
    }
    catch (Exception)
    {
        // Add exception logic.
    }

    // Set up the LocationEntity object.
    LocationEntity locationEntity = new LocationEntity();
    locationEntity.Location = geoposition.Coordinate.Point;
    locationEntity.Message = "You are here";

    // Specify the location that will be at the center of the map.
    myMapControl.Center = locationEntity.Location;

    // Set the map controls data context so data binding will work.
    myMapControl.DataContext = locationEntity;

    // Zoom level.
    myMapControl.ZoomLevel = 15;

    // The map control has done most of its work. Make the controls visible.
    imgMyLocation.Visibility = Windows.UI.Xaml.Visibility.Visible;
    tbMessage.Visibility = Windows.UI.Xaml.Visibility.Visible;

    // Collapse the progress bar.
    pbProgressBar.Visibility = Windows.UI.Xaml.Visibility.Collapsed;
}
```

The previous two tips should substantially improve performance. However, in areas with poor Internet connectivity and on devices with low memory and slower CPUs, the Map control and the Mapping Services APIs may still need time, resulting in the user experiencing a delay. In such cases, the Map control will display a map of the globe at the current zoom level until a location is set into the Map control's center point property. This is also a poor UX because the user should be given some visual indication that work is occurring. In the XAML in **Figure 1**, a progress bar is added as a child of the Map control and is made visible. The progress bar is set up as indeterminate because the exact amount of time needed isn't known. Once the center property of the map is calculated and set into the Map control, the progress bar can be collapsed. Best practices for the use of indeterminate progress bars indicate they should always be placed at the very top of the page, and for most scenarios, I agree. However, when using the Map control, I prefer to see the progress bar displayed right across the center of the map. The Map control tends to grab the user's focus and for this reason a progress bar at the very top of the page might be overlooked. Also,

a progress bar shown across the Map control tells the user that it's the Map control doing work.

Data Binding a Collection

Binding a collection to the Map control is similar to binding a collection to a `ListBox`, `ListView`, `GridView` or any other control that displays collections. To illustrate this with the Map control, consider the following class:

```
public class Team
{
    public string TeamName { get; set; }
    public Uri ImageSource { get; set; }
    public string Division { get; set; }
    public string StadiumName { get; set; }
    public Geopoint StadiumLocation { get; set; }
    public Point NAP { get; set; }
}
```

An instance of this class could be used to hold basic information about a professional sports team. Notice that one of the properties is a `Geopoint` property that contains the location of the stadium used as the team's home field.

The XAML in **Figure 4** sets up the Map control such that a collection of `Team` objects can be bound to it and the location of each team's home field stadium will be marked with a small icon.

If you're familiar with collection binding in other controls, nothing in **Figure 4** should be unfamiliar. An `Image` control is contained within a `DataTemplate`. The `Image` control's attached `Location` property is bound to the `StadiumLocation` property of the `Team` object. The `Image` control also has an attached `NormalizedAnchorPoint` property, which is bound to a property that centers the image over the stadium location.

Note that when binding the Map control to a collection, the `NormalizedAnchorPoint` property can't be hardcoded. It must be bound to a value in the underlying object. If you try to set this property like this:

```
Maps:MapControl.NormalizedAnchorPoint="0.5,0.5"
```

the XAML editor will not like this syntax and the value will never get set. Microsoft is aware of this problem.

Figure 5 shows the Map control displaying the NFL stadium locations. (This screenshot is from the code sample that accompanies this article. The code sample contains all the ancillary code not shown here for the sake of brevity, such as instantiating the collection and properly formatting the control to fit nicely on a page.)

Figure 4 Setting Up the Map Control for Collection Binding

```
<Maps:MapControl
    x:Name="myMapControl" Grid.Row="1"
    MapServiceToken="{StaticResource MapServiceTokenString}" >

    <Maps:MapItemsControl x:Name="MapItems" >
        <Maps:MapItemsControl.ItemTemplate>
            <DataTemplate>
                <Image Source="{Binding ImageSource}"
                    Maps:MapControl.Location="{Binding StadiumLocation}"
                    Maps:MapControl.NormalizedAnchorPoint="{Binding NAP}"
                    Height="15" Width="15"
                    />
            </DataTemplate>
        </Maps:MapItemsControl.ItemTemplate>
    </Maps:MapItemsControl>

</Maps:MapControl>
```

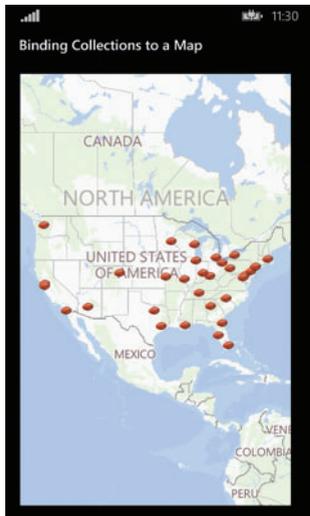


Figure 5 The Stadiums of the National Football League

certain distance involves determining how much distance is represented by a single pixel on the Map control. Here's the equation for determining the distance represented by a single pixel on the Map control:

```
const double BING_MAP_CONSTANT = 156543.04;
double MetersPerPixel =
    BING_MAP_CONSTANT * Math.Cos(latitude) / Math.Pow(2, zoomLevel);
```

The latitude used should come from the Map control's center property. It must be in radians, not degrees. Because latitude is typically expressed in degrees, you'll have to convert it to radians by multiplying the degree value by $\text{Pi}/180$ ($\text{Math.PI}/180$). The zoom level comes from the zoom level of the Map control and is a value from 1 to 20. On Windows Phone 8.1, it's expressed as a double.

A complete mathematical derivation of the equation shown here is beyond the scope of this article; however, a few comments are in order.

At first glance the equation may seem incorrect. Why does the latitude matter? Shouldn't the scale of a map simply be a function of the zoom level? The truth is that latitude does matter and the zoom level alone is insufficient when determining the scale of a map. Bing Maps, which is the back end for the Map control, makes use of the Mercator projection. The Mercator projection is a technique for projecting the details of the surface of a sphere (in this case the Earth) onto a square surface. This projection introduces inconsistencies in scale because the farther you go from the equator, the more stretching there is to maintain a square fit. A simple experiment can help you better understand the Mercator projection. Peel an orange and save all the pieces. When the orange is completely peeled, put

Scaling Controls

the pieces back together on a flat surface and try to get them to exactly fit into a square shape. It's impossible. The only way to get these pieces into a square shape is to stretch the pieces that are not on the "equator" of your orange. The closer you get to the "poles," the more stretching you need to do to get a square fit. This stretching creates a difference in scale as compared to locations close to the equator. The amount of stretching needed is a function of the distance from the equator. Hence, the dependency on latitude. As an example of this stretching in action on a Mercator projection of the Earth, consider the two cities of Quebec, Canada, and Key West, Fla. A completely zoomed-in view (Zoom Level of 20) of Quebec using the Map control will result in 100 pixels representing 33.5 feet. The same zoom level of Key West will result in 100 pixels representing 44.5 feet. The residents of Quebec enjoy a scale that's 11 feet better than the residents of Key West—a small consolation for the colder climate. For more information on the Mercator projection, check out the MSDN Library article, "Understanding Scale and Resolution," at bit.ly/1x1qEwC.

Controls in Windows Phone 8.1 are sized by specifying width and height in pixels. Therefore, scaling controls to represent a cer-

tain distance involves determining how much distance is represented by a single pixel on the Map control. Here's the equation for determining the distance represented by a single pixel on the Map control:

tain distance involves determining how much distance is represented by a single pixel on the Map control. Here's the equation for determining the distance represented by a single pixel on the Map control:

Figure 6 shows a TextBlock and a Rectangle added to a Map control to be used as a distance scale. Figure 7 shows the ZoomLevelChanged event and the logic needed to create a distance scale. (The code sample checks the RegionInfo.CurrentRegion.IsMetric property and shows metric values for users who are more familiar with the metric system.) Finally, Figure 8 shows a screenshot from the code sample—a distance scale added to a Map control.

Downloading Maps for Offline Use

Maps can be downloaded for offline use. This is handy for applications that need to provide mapping functionality when the device is without any form of Internet connectivity, such as while a user

Figure 6 Xaml TextBlock and Rectangle Used As a Distance Scale

```
<Maps:MapControl
    x:Name="myMapControl" Grid.Row="1"
    MapServiceToken="{StaticResource MapServiceTokenString}"
    ZoomLevelChanged="myMapControl_ZoomLevelChanged">
    <!-- Distance scale, which is located at the lower left of the Map control. -->
    <TextBlock Name="tbScale" Text="Scale Text" FontSize="15" Foreground="Black"
        Margin="24,530,24,6" Opacity="0.6" />
    <Rectangle Name="recScale" Fill="Purple" Width="100" Height="6"
        Margin="24,548,24,24" Opacity="0.6" />
</Maps:MapControl>
```

Figure 7 ZoomLevelChanged Event

```
private void myMapControl_ZoomLevelChanged(MapControl sender, object args)
{
    // Get the Map control's current zoom level.
    double zoomLevel = sender.ZoomLevel;

    // Use the latitude from the center point of the Map control.
    double latitude = sender.Center.Position.Latitude;

    // The following formula for map resolution needs latitude in radians.
    latitude = latitude * (Math.PI / 180);

    // This constant is based on the diameter of the Earth and the
    // equations Microsoft uses to determine the map shown for the
    // Map control's zoom level.
    const double BING_MAP_CONSTANT = 156543.04;

    // Calculate the number of meters represented by a single pixel.
    double MetersPerPixel =
        BING_MAP_CONSTANT * Math.Cos(latitude) / Math.Pow(2, zoomLevel);

    // Additional units.
    double KilometersPerPixel = MetersPerPixel / 1000;
    double FeetPerPixel = MetersPerPixel * 3.28;
    double MilesPerPixel = FeetPerPixel / 5280;

    // Determine the distance represented by the rectangle control.
    double scaleDistance = recScale.Width * MilesPerPixel;
    tbScale.Text = scaleDistance.ToString() + " miles";
}
```

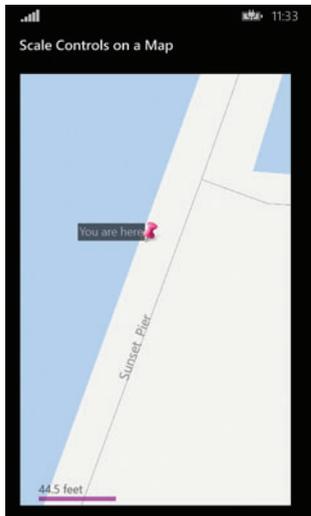


Figure 8 Distance Scale Added to a Map Control Showing Sunset Pier in Key West, Fla.

is driving on a long highway between urban areas or hiking deep in the woods.

Maps can't be downloaded programmatically without user consent because they take up a significant amount of storage. The user must opt in to each download by using a system-provided UX that allows needed maps to be selected and downloaded. This is accomplished using the static class `MapManager`, which belongs to the `Windows.Services.Maps` namespace. This class provides functions for both downloading maps and updating maps. The `ShowDownloadedMapsUI` function shown here will launch

the built-in Maps application and display the page shown in Figure 9:

```
MapManager.ShowDownloadedMapsUI();
```

The built-in maps application is named Maps and the page shown in Figure 9 can also be accessed by tapping the Download maps button from the Settings menu option. Because this function takes the user to another application (the built-in maps application), he will have to return to the original application manually.

The page shown in Figure 9 shows all the maps that have been previously downloaded. It also contains a menu option that allows you to delete any of the previously downloaded maps in case storage on the device needs to be freed or a map is no longer needed.

Tapping the add button (+) starts the map selection wizard, displaying a page that lets you specify the continent containing the map you wish to download. Once a continent is selected, the page displaying all countries within the selected continent will be displayed. You can select all the countries within a continent and



Figure 9 The UI for Downloading Maps

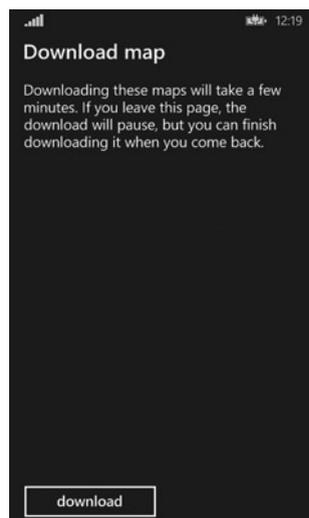


Figure 10 Downloading a Map

download maps for the entire continent, but this would require a large amount of storage. For example, all the maps for the United States require 3.4GB of storage. The entire continent of North and Central America would require much more.

If you select a large country that contains multiple regions or states, you'll see a page that lets you choose specific regions or states. You can select more than one region by using the Select button. When a region is specified, you'll be taken to the Download page shown in Figure 10. This same page will also be shown if you select a small country without any regions or states.

Before downloading any maps, it's a good idea to encourage users to check the available space on the device by going to the Settings application and selecting Storage sense. This will show the available space on the device, as well as the storage used by each application on the device in case space needs to be freed up. At the time of this writing, there are no WinRT APIs that allow you to programmatically determine the amount of storage used on a device or the amount of storage available.

It's also a good idea to download maps while connected to an unrestricted network, such as Wi-Fi. The code sample for this article shows how to check the current network connection and warn the user if they're on a metered connection.

Updating Downloaded Maps

Maps are occasionally changed or modified. New roads may be added to a city, street names may change, and regional borders are sometimes altered. Therefore, downloaded maps should be periodically updated. The `MapManager` static class contains a function named `ShowMapsUpdateUI` for updating downloaded maps:

```
MapManager.ShowMapsUpdateUI();
```

The `ShowMapsUpdateUI` function is similar to the `ShowDownloadedMapsUI` function in that it takes the user to the built-in maps application. When this function is called, all downloaded maps are checked to see if they need to be updated. If any previously downloaded maps require updating, the user will be told the size of the update and given the option to either cancel or proceed with the download. This same functionality is also available from the Settings menu option of the Maps application.

If your application encourages the downloading of maps by utilizing `ShowDownloadedMapsUI`, it's a best practice to also use `ShowMapsUpdateUI` so that downloaded maps can stay current.

Wrapping Up

In this article, I showed you how to add application data to the Map control for Windows Phone 8.1. This included adding XAML controls to a map, data binding and drawing controls to scale. I also showed how to take the Map control's underlying data offline so the Map control can be set up to work seamlessly within applications designed for offline use. ■

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

THANKS to the following Microsoft technical expert for reviewing this article: Mike O'Malley

facebook



Microsoft SharePoint 2010



Linked in



twitter

SEE THE WORLD AS A DATABASE

amazon web services

bing

amazon web services

Microsoft Excel 2010

Microsoft SQL Server

OData Open Data Protocol



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

Microsoft SQL Server

Linked in

SAP

OData Open Data Protocol

Salesforce



facebook



Microsoft SharePoint 2010

amazon web services



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

Work With Relational Data, Not Complex APIs or Services

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



Give RSSBus a try today and see what mean:

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

rssbus

INTEGRATION YOUR WAY

Cross-Platform Cloud Automation with JavaScript

Steven Edouard

Not everyone knows Microsoft Azure has the Cross-Platform Command-Line Interface (xplat-cli) for managing Azure Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) services, but it does. For years, Microsoft has advocated using Windows PowerShell as a scripting language, which worked well for provisioning cloud-based resources. But now that more developers are involved in deploying and moving applications to the cloud, it makes more sense to use JavaScript. The Azure Service Management REST APIs make this cross-platform implementation easier. The official Azure xplat-cli is implemented on top of Node.js, the same runtime often used to build high-performance Web sites and APIs with JavaScript.

Building on top of Node.js makes scripting truly cross-platform. It means you can run your scripts for managing virtual machines (VMs), SQL Databases, Web sites and virtual networks from any machine. This frees you to do automation scripts in a wider variety of languages and for a wider variety of client computers (see **Figure 1**).

There are many advantages to scripting cloud resource provisioning. For example, the Web portal isn't repeatable and requires human input. Another advantage is being able to source control the steps to provision and deploy your service using different specific configuration values. In this article, I'll walk through scripting the process of creating VMs associated with an Affinity Group and a virtual network using Node.js as our script automation environment.

This article discusses:

- How to automate configuration tasks with cross-platform scripting
- Configuring and deploying groups of VMs
- Confirming lists of provisioned VMs

Technologies discussed:

Microsoft Azure, Windows PowerShell, Node.js

The Cross-Platform CLI

Like the Azure PowerShell CLI, the cross-platform version offers powerful script ability to manage IaaS and PaaS services. This lets you describe your infrastructure with code, share it with your team via source control, and automate the creation of development, test, and production application deployments.

To get started with the xplat-cli, head over to Nodejs.org and install the appropriate Node.js package for your system. After installing Node, you can view the azure-cli package on npmjs.org/package/azure-cli.

The package listing gives you all the relevant documentation on the CLI, as well as the simple command line to install the package. This package is a CLI, so it makes sense to install it in the global Node.js package store by specifying `-g`, so you can make Azure

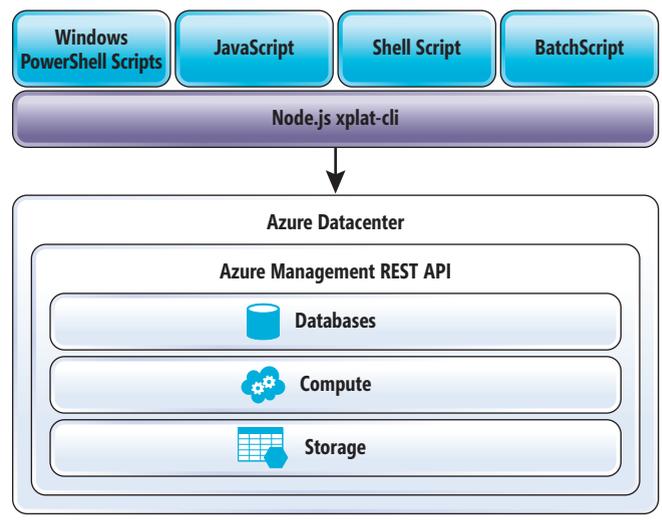


Figure 1 Supported Scripting Environments

If you browse closely on the `azure-cli` npm documentation (npmjs.org/package/azure-cli), you'll notice that there's a library dependent on `azure-cli` called `azure-scripty`. It turns out someone in the open source Node.js community has already decided to write a command-line adapter for the `azure-cli` for JavaScript. Because this library is a command-line wrapper, you can use all the same commands as on the command line in JavaScript. Here's an example that will list all of our VMs:

Figure 5 Create an Affinity Group

```

////////////////////////////////////
//Creates an affinity group if the specified one doesn't exist
////////////////////////////////////
function createAffinityGroup(cb) {
  console.log('Getting affinity groups...');
  scripty.invoke('account affinity-group list', function (err, result) {

    // Check for errors.
    if (err) {
      return cb(err);
    }
    console.log('Current affinity groups');
    // For debugging purposes, list out names of existing affinity groups.
    for (var i in result) {
      console.log(result[i].name);
    }
    // Get the name of the desired affinity group from the config.json file.
    var affinityGroup = nconf.get('affinity_group').name;
    var label = nconf.get('affinity_group').label;
    // Verify that affinity group hasn't already been created.
    for (var i in result) {
      if (result[i].name === affinityGroup && result[i].label === label) {
        // The affinity group to use in the config already exists.
        return cb();
      }
    }
    // Create affinity group because it doesn't exist.
    console.log('Specified affinity group ' + affinityGroup +
      ' does not exist, creating new one...');
    var cmd = {
      command: 'account affinity-group create',
      positional: [affinityGroup],
      location: '"West US"',
      label: label
    };
    scripty.invoke(cmd, function (err) {
      if (err) {
        cb(err);
      }
      return cb();
    });
  });
}

```

Figure 6 Create a Virtual Network

```

////////////////////////////////////
//Creates the config specified vnet, if it doesn't already exist
////////////////////////////////////
function createVirtualNetwork(cb) {
  console.log('Getting networks...');
  scripty.invoke('network vnet list', function (err, result) {

    if (err) {
      return cb(err);
    }

    var networkName = nconf.get('vnet_name');

    for (var i in result) {
      console.log(result[i].name);
    }

    // Check if VNet name listed in config.json exists.

    // If it doesn't, create the VNet.
  });
}

```

```

// Get the list of existing VMs on this subscription
scripty.invoke('vm list', function (err, result) {

  if (err) {
    return cb(err);
  }

  for (var i in result) {

    console.log(result[i].VMName);
  }
});

```

To get started, take a look at how you can make a batch of VMs. First, install the necessary packages:

```

# a command-line wrapper for the azure-cli
npm install azure-scripty
# a small library that helps manage asynchronous tasks
npm install async
# a library that makes using JSON configuration files easy
npm install nconf

```

To avoid hardcoding the configuration settings for staging, test and production, use a configuration file to keep the scripts flexible enough for different environments such as development, test and production. Call this configuration file `config.json` (see Figure 4).

You can use that configuration file to set up a developer environment, complete with three Ubuntu VMs that are part of the affinity group (`azuredemoaffinitygroup`) and a corresponding virtual network (`azuredvnet2`). This approach is declarative, meaning you only need to make changes to a configuration file to change the way your development environments are set up. You can also have additional versions for staging, test and production—and source control each.

The custom JavaScript code will parse the configuration file so the dependencies, such as affinity groups, will be provisioned with the correct names and values. Inside the `config.json` file, the script can read the desired name for the affinity group (`azuredemoaffinitygroup`). From there, it can query check to see if the affinity group has already been created by looping through the list of provisioned affinity groups that already exist in the datacenter (see Figure 5).

Figure 7 Create a List of Virtual Machines to Provision

```

scripty.invoke('vm list', function (err, result) {

  if (err) {
    return cb(err);
  }

  // Read the desired VM name from config.json.
  var baseName = nconf.get('vm_base_name');

  var vmNames = [];

  // Create the array of the computed VM names.
  for (var z = 0; z < count; z++) {
    vmNames.push(baseName + z.toString());
  }

  // Go through the list of existing VMs.
  for (var i in result) {

    for (var k in vmNames) {

      if (result[i].VMName === vmNames[k]) {
        // A VM we intend on creating already exists on this sub.
        // Remove it on the list of VMs to create.
        delete vmNames[k];
      }
    }
  }

  // vmNames now only contains the name of VMs that do not exist.
  // Create them.

```

Figure 8 Finish Provisioning Virtual Machines

```
var domainName = nconf.get('dns_name');
var userName = nconf.get('vm_username');
var password = nconf.get('vm_password');
var imageName = nconf.get('vm_image_name');
var vmCreationTasks = [];
var taskArguments = [];

for (var m in vmNames) {

  if (vmNames[m]) {
    var cmd = {
      command: 'vm create',
      positional: [vmNames[m], imageName, userName, password],
      'affinity-group': nconf.get('affinity_group').name,
      'virtual-network-name': nconf.get('vnet_name')
    }
    // Define the async task function that will create each VM.
    var task = function (args, cb) {
      console.log('Creating vm ' + vmNames[args[0]]);
      scripty.invoke(args[1], function (err) {

        if (err) {
          console.log('Vm creation failed: ' + vmNames[args[0]]);
          return cb(err);
        }

        console.log('vm creation of ' + vmNames[args[0]] + ' successful!');
        cb();
      });
    };
    // Bind this function to this context and pass the index and VM create command.
    task = task.bind(this, [m, cmd]);
    vmCreationTasks.push(task);
  }
}

// Execute each VM creation task using the async library.
async.series(vmCreationTasks, function (err) {
  if (err) {
    return cb(err);
  }
  console.log('All VMs created successfully!');
  cb();
})
```

Notice how the `azure-scripty` library lets you easily parse output from the `azure-cli` via callback function result argument. It also lets you create CLI commands by declaratively specifying the command, positional and named parameters.

The pattern in **Figure 6** is the same one you use to create your virtual network and uses the `azure-scripty` library in the same way.

Some of the code in **Figure 6** has been omitted for brevity, but the same pattern exists. The needed resource is specified in the configuration file. You'll need to provision it if it doesn't exist.

You can create a batch of VMs once you've created the dependencies (the affinity groups and networks). The number of VMs you'll create is specified in the `vm_count` field in the `config.json` configuration file. Following the same pattern, first list the current VMs on the subscription, check if any of the VMs you'll create already exist and create only the ones that don't exist.

Figure 7 will follow a simple algorithm to name the VMs using the `vm_base_name` field in the `config.json` and append the numbers 0 – (`vm_count` – 1) to the end of the VM name. Given the current `config.json`, create `sedouardmachine0`, `sedouardmachine1` and `sedouardmachine2` because none of those machines already exist on the subscription.

Figure 7 confirms the list of non-existing VMs in the `vmNames` array. **Figure 8** will use the `async` library to manage the asynchronous tasks that will go off and create the VMs.

Figure 8 creates a task for each VM you need to create. Then you need to add those tasks to an array, `vmCreationTasks`. Each task function is bound to the necessary arguments in order to call the Azure `azure-cli` with the correct VM name. Afterward, use the `async` library to run each task in series and execute the callback with a null or an error parameter indicating if the VMs were successfully created. **Figure 9** confirms the output of the script and lists out all the VM that have been created.

Each of your VMs has been successfully created. You can even check back to your portal to confirm that the machines have been created and are part of the specified virtual network. Use the GitHub repository at bit.ly/azure-xplat-cli to get started scripting your Azure setup using Node.js.

Wrapping Up

The Azure `xplat-cli` lets you completely automate the process of configuring and deploying Azure resources in portable scripts you can use on virtually any OS. It also lets you use source control and configuration files to simplify sharing. You could have multiple versions of `config.json`, each one representing a deployment model for test, staging and production.

You can also recreate the deployment process quickly and efficiently. The forthcoming Azure Resource Manager Microsoft will release in 2015 will go above and beyond what's illustrated here, but this approach is virtually identical to what the Azure Resource Manager will do—which is declaratively specify how resources are provisioned in the Azure datacenter. ■

STEVEN EDOUARD is a developer evangelist at Microsoft. Before that, he worked on the .NET runtime team as a software test engineer delivering products like the Microsoft .NET Framework 4.5 and .NET Native Compilation. Now his passion resides in exciting people on the limitless potentials of cloud computing services through engaging technical demonstrations, online content and presentations.

THANKS to the following Microsoft technical expert for reviewing this article: Bruno Terkaly

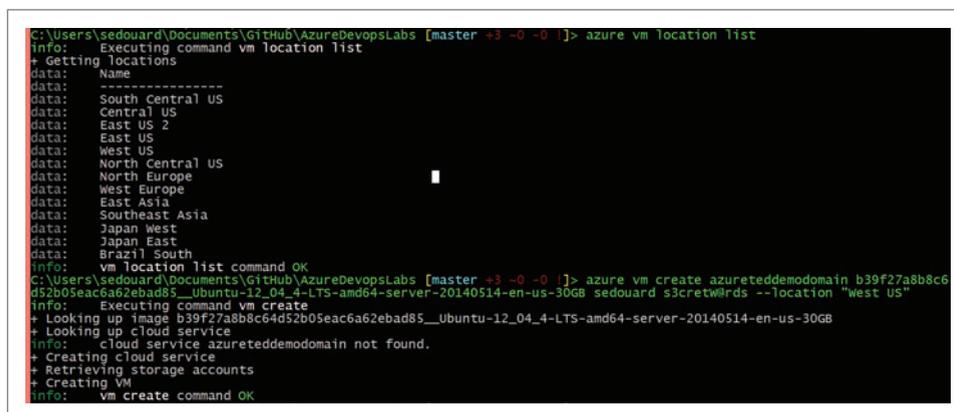


Figure 9 The Final List of Virtual Machines



NAVIGATE THE .NET HIGHWAY

2 Locations to Choose from in June!

Visual Studio Live! is hitting the open road on the ultimate code trip to help you navigate the .NET Highway. In June, we're pullin' into **Austin, TX, June 1 – 4** and then headed to **San Francisco, CA, June 15 – 18.**

Both of these great locations will feature four days of .NET developer training sessions, workshops and networking events – all designed to make you better at your job.

Code with industry experts, get practical answers to your challenges, and immerse yourself in what's to come on the .NET horizon. The full agenda and speaker line-up will be available for both events in early 2015. Stay tuned!





Austin

JUNE 1-4, 2015

HYATT REGENCY, AUSTIN, TX
vslive.com/austin



TRACKS INCLUDE:

- Visual Studio / .NET
- JavaScript/HTML5
- ASP.NET
- Windows 8.1/WinRT
- Cross-Platform Mobile Development
- Database and Analytics
- Cloud Computing
- SharePoint/Office



San Francisco

JUNE 15-18, 2015

THE FAIRMONT, SAN FRANCISCO, CA
vslive.com/sf

REGISTER NOW AND SAVE \$300!



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

Use promo code VSLCODE2

vslive.com

Join us on the Ultimate Code Trip in 2015!



FOLLOW US

twitter.com/vslive - @VSLive

[facebook.com](https://www.facebook.com/vslive) - Search "VSLive"

[linkedin.com](https://www.linkedin.com/groups?gid=11111111) - Join the "Visual Studio Live" group!



Logistic Regression Classification with Multi-Swarm Optimization

One of the most fundamental forms of machine learning is logistic regression (LR) classification. The goal of LR classification is to create a model that predicts a variable that can take on one of two possible values. For example, you might want to predict which of two candidates a voter will select ("Smith" = 0, "Jones" = 1) based on the voter's age (x1), sex (x2) and annual income (x3).

If Y is the predicted value, an LR model for this problem would take the form:

$$z = b_0 + b_1(x_1) + b_2(x_2) + b_3(x_3)$$

$$Y = 1.0 / (1.0 + e^{-z})$$

Here, b0, b1, b2 and b3 are weights, which are just numeric values that must be determined. In words, you compute a value z that is the sum of input values times b-weights, plus a b0 constant, then feed the z value to the equation that uses math constant e. It turns out that Y will always be between 0 and 1. If Y is less than 0.5, you conclude the output is 0 and if Y is greater than 0.5 you conclude the output is 1.

For example, suppose a voter's age is 32, sex is male (-1), and annual income in tens of thousands of dollars is 48.0. And suppose b0 = -9.0, b1 = 8.0, b2 = 5.0, and b3 = -5.0. Then z = -9.0 + (8.0)(32) + (5.0)(-1) + (-5.0)(48.0) = 2.0 and so Y = 1.0 / (1.0 + e^{-2.0}) = 0.88.

Because Y is greater than 0.5, you'd conclude the voter will pick candidate 1 ("Jones"). But where do the b-weight values come from? Training an LR classifier is the process of finding the values for the b-weights. The idea is to use training data that has known output values and then find the set of b values so that the difference between the computed output values and the known output values is minimized. This is a math problem that's often called numerical optimization.

There are about a dozen major optimization algorithms used in machine learning. For logistic regression classification training, two of the most common algorithms are called iterative Newton-Raphson and L-BFGS. In this article, I present a technique called multi-swarm optimization (MSO). MSO is a variation of particle swarm optimization (PSO). In MSO, a virtual particle has a position that corresponds to a set of b-weight values. A swarm is a collection of particles that move in a way inspired by group behavior such as the flocking

```

file:///C:/LogisticWithMulti/bin/Debug/LogisticWithMulti.EXE
Begin Logistic Regression Classification with Multi-Swarm demo
Generating 10000 artificial data items with 5 features
Data generation weights:
4.5249 6.3465 5.3605 1.1632 -5.8793 1.1777
Done
Creating train (80%) and test (20%) matrices
Done
Training data:
[ 0] -6.82 -6.85 1.00 -4.14 7.11 0.00
[ 1] 5.40 -1.86 8.00 -0.78 -0.75 1.00
[ 2] 1.77 2.49 -4.15 -0.41 5.45 1.00
[ 3] -3.75 5.32 6.85 2.36 4.07 1.00
[7999] -6.66 -8.50 4.52 -4.76 1.17 0.00
Test data:
[ 0] 4.49 1.94 3.09 2.72 -5.71 1.00
[ 1] 6.69 1.86 -3.00 -4.65 -0.32 1.00
[ 2] -3.76 -6.80 -1.38 9.69 0.59 0.00
[1999] -8.00 -5.77 -2.03 1.99 4.67 0.00
Creating Logistic Regression classifier
Setting numSwarms = 4
Setting numParticles = 3
Setting maxEpochs = 100
Starting training
Epoch = 10 Best error = 0.0220
Epoch = 20 Best error = 0.0096
Epoch = 30 Best error = 0.0040
Epoch = 40 Best error = 0.0026
Epoch = 50 Best error = 0.0017
Epoch = 60 Best error = 0.0016
Epoch = 70 Best error = 0.0015
Epoch = 80 Best error = 0.0015
Epoch = 90 Best error = 0.0015
Training complete
Best weights found:
4.0900 10.0000 8.4312 1.8636 -9.2719 1.8413
Accuracy on training data = 0.9998
Accuracy on test data = 0.9985
End Logistic Classification with Multi-Swarm demo

```

Figure 1 Logistic Regression with Multi-Swarm Optimization in Action

of birds. MSO maintains several swarms that interact with each other, as opposed to PSO, which uses just one swarm.

A good way to see where this article is headed, and to get an idea of what LR with MSO is, is to take a look at the demo program in **Figure 1**. The goal of the demo program is to use MSO to create an LR model that predicts Y for a set of synthetic (programmatically generated) data.

The demo program begins by creating 10,000 random data items where there are five predictor variables (often called features in ML terminology). Each feature value is between -10.0 and +10.0, and the Y-value, 0 or 1, is in the last column of the data set. The feature values don't correspond to a real problem.

Code download available at msdn.microsoft.com/magazine/msdnmag0115.

The 10,000-item data set is randomly split into an 8,000-item training set used to create the LR model, and a 2,000-item hold-out test set used to evaluate the accuracy of the model after training. The demo program creates an LR classifier and then uses four swarms, each with three particles, to train the classifier. MSO is an iterative process and the maximum number of iterations, maxEpochs, is set to 100.

A swarm is a collection of particles that move in a way inspired by group behavior such as the flocking of birds.

The demo displays the best (smallest) error found by any particle, every 10 epochs. After training is completed, the best weights found are (4.09, 10.00, 8.43, 1.86, -9.27, 1.84). Using these weights, the accuracy of the LR model is computed for the training data (99.98 percent correct, which is 7,998 out of 8,000) and for the test data (99.85 percent correct, which is 1,997 out of 2,000). The accuracy of the model with the test data gives you a rough approximation of how well the model would do when presented with new data that has unknown output values.

This article assumes you have at least intermediate programming skills, but doesn't assume you know anything about logistic regression classification or multi-swarm optimization. The demo

program is coded using C#, but you shouldn't have too much difficulty refactoring the code to another language such as Visual Basic .NET or Python.

The demo code is too long to present here in its entirety, but the complete source code is available in the code download that accompanies this article. The demo code has all normal error checking removed to keep the main ideas as clear as possible and the size of the code small.

Overall Program Structure

The overall program structure, with a few minor edits to save space, is presented in **Figure 2**. To create the demo, I launched Visual Studio and created a new C# console application named LogisticWithMulti. The demo has no significant Microsoft .NET Framework dependencies, so any recent version of Visual Studio will work.

After the template code loaded into the Visual Studio editor, in the Solution Explorer window I renamed file Program.cs to the more descriptive LogisticMultiProgram.cs and Visual Studio automatically renamed class Program for me. At the top of the source code, I deleted all using statements that pointed to unneeded namespaces, leaving just the reference to the top-level System namespace.

The program class has helper methods MakeAllData, MakeTrainTest, ShowData and ShowVector. All of the logistic regression logic is contained in a single LogisticClassifier class. The LogisticClassifier class contains nested helper classes Particle, Swarm, and MultiSwarm to encapsulate the MSO data and logic used during training. These helper classes could've been defined outside of the LogisticClassifier class.

Figure 2 Overall Program Structure

```
using System;
namespace LogisticWithMulti
{
    class LogisticMultiProgram
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Begin demo");

            int numFeatures = 5;
            int numRows = 10000;
            int seed = 0;

            Console.WriteLine("Generating " + numRows +
                " artificial data items with " + numFeatures + " features");
            double[][] allData = MakeAllData(numFeatures, numRows, seed);
            Console.WriteLine("Done");

            Console.WriteLine("Creating train and test matrices");
            double[][] trainData;
            double[][] testData;
            MakeTrainTest(allData, 0.80, seed, out trainData, out testData);
            Console.WriteLine("Done");

            Console.WriteLine("Training data: ");
            ShowData(trainData, 4, 2, true);

            Console.WriteLine("Test data: ");
            ShowData(testData, 3, 2, true);

            Console.WriteLine("Creating Logistic Regression classifier");
            LogisticClassifier lc = new LogisticClassifier(numFeatures);

            int numSwarms = 4;
            int numParticles = 3;
            int maxEpochs = 100;
            Console.WriteLine("Setting numSwarms = " + numSwarms);

            Console.WriteLine("Setting numParticles = " + numParticles);
            Console.WriteLine("Setting maxEpochs = " + maxEpochs);
            Console.WriteLine("\nStarting training");
            double[] bestWeights = lc.Train(trainData, maxEpochs,
                numSwarms, numParticles);
            Console.WriteLine("Training complete");
            Console.WriteLine("Best weights found:");
            ShowVector(bestWeights, 4, true);

            double trainAcc = lc.Accuracy(trainData, bestWeights);
            Console.WriteLine("Accuracy on training data = " +
                trainAcc.ToString("F4"));

            double testAcc = lc.Accuracy(testData, bestWeights);
            Console.WriteLine("Accuracy on test data = " +
                testAcc.ToString("F4"));

            Console.WriteLine("End demo");
            Console.ReadLine();
        } // Main

        static double[][] MakeAllData(int numFeatures,
            int numRows, int seed) { . . . }

        static void MakeTrainTest(double[][] allData, double trainPct, int seed,
            out double[][] trainData, out double[][] testData) { . . . }

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

        static void ShowVector(double[] vector, int decimals,
            bool newLine) { . . . }
    } // Program

    public class LogisticClassifier { . . . }
} // ns
```

The Main method has a lot of WriteLine noise. The key calling statements are quite simple. The synthetic data is generated, like so:

```
int numFeatures = 5;
int numRows = 10000;
int seed = 0; // Gives representative demo
double[][] allData = MakeAllData(numFeatures, numRows, seed);
```

Method MakeAllData creates random b-weight values between -10.0 and +10.0, then for each data item, random x-values between -10.0 and +10.0 are generated and combined with the b-weight values, which are then used to generate Y-values. The synthetic data corresponds to a data set where the x-values have been normalized, and where there are roughly equal counts of Y = 0 and Y = 1 values.

The data is split into training and test sets with these statements:

```
double[][] trainData;
double[][] testData;
MakeTrainTest(allData, 0.80, seed, out trainData, out testData);
```

The LR model is created and trained with these statements:

```
LogisticClassifier lc = new LogisticClassifier(numFeatures);
int numSwarms = 4;
int numParticles = 3;
int maxEpochs = 100;
double[] bestWeights = lc.Train(trainData, maxEpochs, numSwarms, numParticles);
```

And the accuracy of the model is evaluated with these two statements:

```
double trainAcc = lc.Accuracy(trainData, bestWeights);
double testAcc = lc.Accuracy(testData, bestWeights);
```

Understanding the MSO Algorithm

Here's the MSO optimization in high-level pseudo-code:

```
for-each swarm
  initialize each particle to a random position
end-for

for-each swarm
  for-each particle in swarm
    compute new velocity
    use new velocity to compute new position
    check if position is a new best
    does particle die?
    does particle move to different swarm?
  end-for
end-for

return best position found
```

The key part of the MSO algorithm is computing a particle's velocity, which is just a set of values that control to where a particle will move. For example, for a problem with just two x-dimensions, if a particle is at (6.0, 8.0) and the velocity is (-2.0, 1.0), the particle's new position will be at (4.0, 9.0).

Velocity is calculated so that a particle tends to move in its current direction; tends to move toward its best position found to date; tends to move toward the best position found by any of its fellow swarm members; and tends to move toward the best position found by any particle in any swarm.

In math terms, if $\mathbf{x}(t)$ is a particle's position at time t , then a new velocity, $\mathbf{v}(t+1)$ is calculated as:

$$\mathbf{v}(t+1) = w * \mathbf{v}(t) + (c1 * r1) * (\mathbf{p}(t) - \mathbf{x}(t)) + (c2 * r2) * (\mathbf{s}(t) - \mathbf{x}(t)) + (c3 * r3) * (\mathbf{g}(t) - \mathbf{x}(t))$$

Term $\mathbf{p}(t)$ is the particle's best-known position. Term $\mathbf{s}(t)$ is the best position of any particle in the particle's swarm. Term $\mathbf{g}(t)$ is the global best position of any

particle in any swarm. Term w is a constant called the inertia factor. Terms $c1$, $c2$ and $c3$ are constants that establish a maximum change for each component of the new velocity. Terms $r1$, $r2$ and $r3$ and random values between 0 and 1 that provide a randomization effect to each velocity update.

When a particle dies, it's replaced with a new particle at a random position.

Suppose a particle is currently at (20.0, 30.0) and its current velocity is (-1.0, -3.0). Also, the best-known position of the particle is (10.0, 12.0), the best-known position of any particle in the swarm is (8.0, 9.0), and the best-known position of any particle in any swarm is (5.0, 6.0). And suppose that constant w has value 0.7, constants $c1$ and $c2$ are both 1.4, and constant $c3$ is 0.4. Finally, suppose random values $r1$, $r2$ and $r3$ are all 0.2.

The new velocity of the particle (with rounding to one decimal) is:

$$\mathbf{v}(t+1) = 0.7 * (-1.0, -3.0) + (1.4 * 0.2) * ((10.0, 12.0) - (20.0, 30.0)) + (1.4 * 0.2) * ((8.0, 9.0) - (20.0, 30.0)) + (0.4 * 0.2) * ((5.0, 6.0) - (20.0, 30.0))$$

$$= 0.7 * (-1.0, -3.0) + 0.3 * (-10.0, -18.0) + 0.3 * (-12.0, -21.0) + 0.1 * (-15.0, -24.0)$$

$$= (-8.8, -16.2)$$

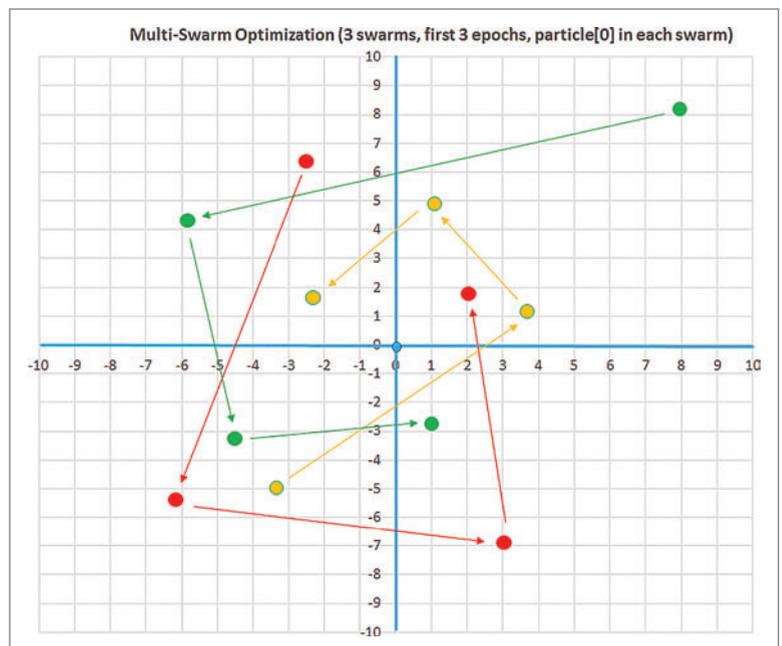


Figure 3 The Multi-Swarm Optimization Algorithm Illustrated

And so the particle's new position is:

$$\begin{aligned} \mathbf{x}(t+1) &= (20.0, 30.0) + (-8.8, -16.2) \\ &= (11.2, 13.8) \end{aligned}$$

The graph in **Figure 3** illustrates the MSO process for a problem with two x-values, three swarms with five particles each, and where the optimal position is at (0, 0). The graph shows how the first particle in each swarm closes in on the optimal position. The spiral motion is characteristic of MSO.

In the MSO pseudo-code, a particle can die with some low probability. When a particle dies, it's replaced with a new particle at a random position. A particle can immigrate with some low probability. When immigration occurs, a particle is swapped with another particle from a different swarm. The death and immigration mechanisms add an element of randomness to MSO and help prevent the algorithm from getting stuck in a non-optimal solution.

Implementing Logistic Regression Using MSO

The definition of method Train begins as:

```
public double[] Train(double[][] trainData, int maxEpochs,
    int numSwarms, int numParticles)
{
    int dim = numFeatures + 1;
    double minX = -10.0;
    double maxX = 10.0;

    MultiSwarm ms = new MultiSwarm(numSwarms, numParticles, dim);
    ...
}
```

The dimension of the problem is the number of predictor features, plus one, to account for the b0 constant. Variables minX and maxX hold the smallest and largest values for any single value in a Particle object's position array. The LogisticClassifier class contains a nested MultiSwarm class. The MultiSwarm class constructor creates an array-of-arrays of Particle objects, each of which initially has a random position, and a random velocity. Because the logistic regression Error method isn't directly visible to the nested Particle definition, the MultiSwarm constructor doesn't supply error value for each particle, so method Train adds error information. First, each particle gets an error value:

```
for (int i = 0; i < numSwarms; ++i)
{
    for (int j = 0; j < numParticles; ++j)
    {
        Particle p = ms.swarms[i].particles[j];
        p.error = Error(trainData, p.position);
        p.bestError = p.error;
        Array.Copy(p.position, p.bestPosition, dim);
    }
    ...
}
```

Next, the current swarm's best error and the overall global best error are computed:

Figure 4 New Velocity for Current Particle Is Calculated

```
for (int k = 0; k < dim; ++k)
{
    double r1 = rnd.NextDouble();
    double r2 = rnd.NextDouble();
    double r3 = rnd.NextDouble();
    p.velocity[k] = (w * p.velocity[k]) +
        (c1 * r1 * (p.bestPosition[k] - p.position[k])) +
        (c2 * r2 * (ms.swarms[i].bestPosition[k] - p.position[k])) +
        (c3 * r3 * (ms.bestPosition[k] - p.position[k]));
    if (p.velocity[k] < minX)
        p.velocity[k] = minX;
    else if (p.velocity[k] > maxX)
        p.velocity[k] = maxX;
} // k
```

```
...
if (p.error < ms.swarms[i].bestError) // Swarm best?
{
    ms.swarms[i].bestError = p.error;
    Array.Copy(p.position, ms.swarms[i].bestPosition, dim);
}
if (p.error < ms.bestError) // Global best?
{
    ms.bestError = p.error;
    Array.Copy(p.position, ms.bestPosition, dim);
}
} // j
} // i
```

The main training loop is prepared, like so:

```
int epoch = 0;
double w = 0.729; // inertia
double c1 = 1.49445; // particle
double c2 = 1.49445; // swarm
double c3 = 0.3645; // multiswarm
double pDeath = 1.0 / maxEpochs;
double pImmigrate = 1.0 / maxEpochs;

int[] sequence = new int[numParticles];
for (int i = 0; i < sequence.Length; ++i)
    sequence[i] = i;
```

The values for constants w, c1 and c2 are the result of some PSO research. Interestingly, compared to many numerical optimization algorithms, PSO and MSO are relatively insensitive to the values used for internal magic constants (called free parameters or hyper parameters). There's little available research for the c3 constant, which influences the tendency of a particle to move toward the best-known position found to date by any particle in any swarm. The value I use, 0.3645, is half the value of the inertia constant, and has worked well for me in practice.

The sequence array holds indices of the training data. This array will be scrambled so that in each swarm, particles will be processed in a different order in each iteration of the main training loop.

The main training loop begins:

```
while (epoch < maxEpochs)
{
    ++epoch;
    // Optionally print best error here
    for (int i = 0; i < numSwarms; ++i) // Each swarm
    {
        Shuffle(sequence);
        for (int pj = 0; pj < numParticles; ++pj) // Each particle
        {
            int j = sequence[pj];
            Particle p = ms.swarms[i].particles[j];
            ...
        }
    }
}
```

Knowing when to stop training is one of the most difficult challenges in machine learning. Here, a fixed number, maxEpochs, of iterations is used. This is a simple approach, but there's a risk you might not train enough. Or you might train too much, which would result in a model that fits the training data extremely well, but works poorly on new data. This is called over-fitting. There are dozens of techniques that can be used to combat over-fitting.

Next, the new velocity for the current particle is calculated, as explained earlier (see **Figure 4**).

After the velocity is calculated, each of its component values is checked to see whether the magnitude is large, and if so, the value is reined back in. This prevents a particle from moving a very great distance in any one iteration. For some ML training tasks, eliminating the velocity constraint appears to speed up the training, but there are no solid research results to give any specific advice here.

Next, the new velocity is used to calculate the current particle's new position and associated error:

```
for (int k = 0; k < dim; ++k)
{
    p.position[k] += p.velocity[k];
    if (p.position[k] < minX)
        p.position[k] = minX;
    else if (p.position[k] > maxX)
        p.position[k] = maxX;
}
```

After the current particle has moved, its new error is calculated:

```
p.error = Error(trainData, p.position); // Expensive
if (p.error < p.bestError) // New best position for particle?
{
    p.bestError = p.error;
    Array.Copy(p.position, p.bestPosition, dim);
}
```

The current particle's new error might be a new swarm best or global best:

```
if (p.error < ms.swarms[i].bestError) // New best for swarm?
{
    ms.swarms[i].bestError = p.error;
    Array.Copy(p.position, ms.swarms[i].bestPosition, dim);
}
if (p.error < ms.bestError) // New global best?
{
    ms.bestError = p.error;
    Array.Copy(p.position, ms.bestPosition, dim);
}
```

After the current particle has moved, an option is to possibly kill that particle and generate a new one. If a random value is below some small threshold, a new particle is generated:

```
double p1 = rnd.NextDouble();
if (p1 < pDeath)
{
    Particle q = new Particle(dim); // A replacement
    q.error = Error(trainData, q.position);
    Array.Copy(q.position, q.bestPosition, dim);
    q.bestError = q.error;
}
```

Instead of using a fixed probability of death, an alternative is to gradually increase the probability of death, for example:

```
double pDeath = (maxProbDeath / maxEpochs) * epoch;
```

After the replacement particle has been created, that particle may be, by pure luck, a new swarm best or global best:

```
if (q.error < ms.swarms[i].bestError) // Best swarm error by pure luck?
{
    ms.swarms[i].bestError = q.error;
    Array.Copy(q.position, ms.swarms[i].bestPosition, dim);
    if (q.error < ms.bestError) // Best global error?
    {
        ms.bestError = q.error;
        Array.Copy(q.position, ms.bestPosition, dim);
    }
}
```

The death mechanism concludes by exchanging the current particle for the replacement, effectively killing the current particle:

```
...
ms.swarms[i].particles[j] = q;
} // Die
```

Next, the (optional) immigration mechanism starts with:

```
double p2 = rnd.NextDouble();
if (p2 < pImmigrate)
{
    int ii = rnd.Next(0, numSwarms); // rnd swarm
    int jj = rnd.Next(0, numParticles); // rnd particle
    Particle q = ms.swarms[ii].particles[jj]; // q points to other
    ms.swarms[i].particles[jj] = q;
    ms.swarms[ii].particles[j] = p; // the exchange
}
...
```

The immigration mechanism is quite crude. The code selects a random swarm and then a random particle from that swarm,

and then swaps the random particle with the current particle. As coded, there's no guarantee the randomly selected particle will be in a swarm different from the current particle.

The immigration mechanism finishes by checking to see if swapping particles in two swarms resulted in one or two new swarm-best positions:

```
...
if (q.error < ms.swarms[i].bestError) // Curr has new position
{
    ms.swarms[i].bestError = q.error;
    Array.Copy(q.position, ms.swarms[i].bestPosition, dim);
}
if (p.error < ms.swarms[ii].bestError) // Other has new position
{
    ms.swarms[ii].bestError = p.error;
    Array.Copy(p.position, ms.swarms[ii].bestPosition, dim);
} // Immigrate
```

Knowing when to stop training is one of the most difficult challenges in machine learning.

Method Train concludes by returning the best position found by any particle:

```
...
} // j - each particle
} // i - each swarm
} // while
return ms.bestPosition;
} // Train
```

The return value is a reference to an array representing a set of weights of logistic regression. A minor alternative to reduce the chances of an unwanted side effect is to copy the values to a local array and return a reference to the local array.

Wrapping Up

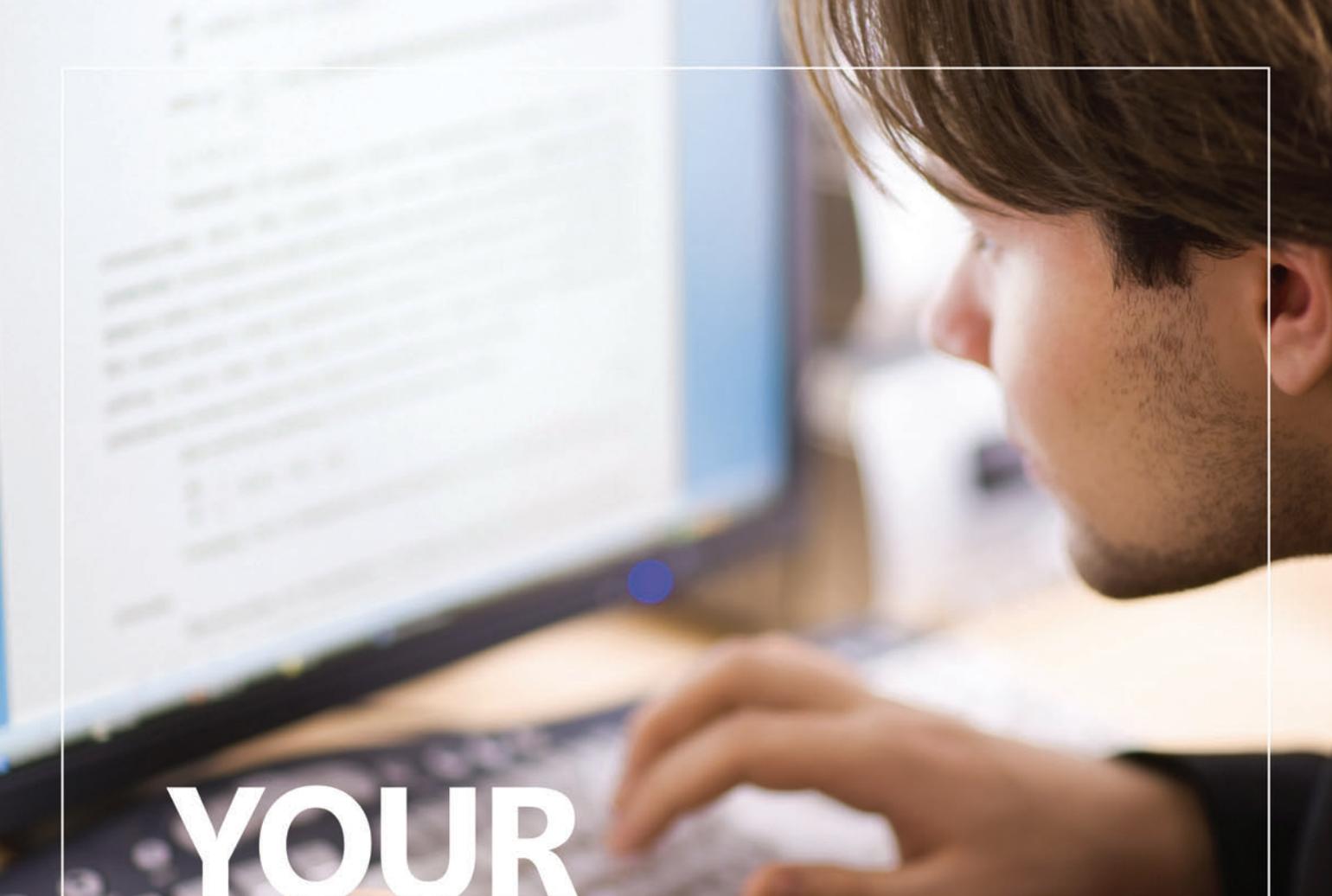
This article and accompanying code should get you up and running if you want to explore logistic regression classification using multi-swarm optimization for training. MSO is really more of a meta-heuristic than an algorithm. By that I mean MSO is a set of design principles that can be implemented in many different ways. There are dozens of modifications to the basic MSO design you can investigate.

Logistic regression is one of the simplest forms of machine learning classification. For many classification problems, LR just doesn't work well. However, many ML practitioners, including me, usually begin investigating a classification problem using LR, and then use more sophisticated techniques, if necessary.

Compared to older, calculus-based numerical optimization techniques such as gradient descent and L-BFGS, based on my experience, training using MSO tends to produce better ML models, but MSO is almost always an order of magnitude slower. ■

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

THANKS to the following Microsoft technical experts for reviewing this article: *Todd Bello and Alisson Sol*



YOUR .NET Resources

msdn
magazine

Visual Studio
Magazine

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

ONLINE | NEWSLETTERS | PRINT | CONFERENCES



An Upstart Again

It's interesting to watch Microsoft as an underdog.

Not that long ago that underdog bestrode the computing world like a colossus. You probably remember the joke, "How many Microsoft programmers does it take to change a light bulb?" The answer, of course, was: "None. Bill Gates just declares Microsoft Darkness to be the new standard."

But while Microsoft concentrated on PCs (see msdn.microsoft.com/magazine/hh224515), new business sectors exploded across the IT industry, namely mobile devices and the cloud. The most recent figures show Microsoft doing well in the latter, which can be seen as the logical progression of the server side of its PC business.

Microsoft is having much more difficulty in the former.

As I wrote in my November 2013 column (msdn.microsoft.com/magazine/dn463796), Microsoft's key to greater market share is making more and better apps available in its ecosystem. This in turn means

The integration of Xamarin with Visual Studio works exactly the way existing Visual Studio users would expect, and their skills transfer about as seamlessly as they could.

harnessing Microsoft's "army of developers ... [who] would love to use their current skills to develop apps for this brave new device-and-tablet world." These developers need to support iOS and Android. They'd support Windows 8, too, if it didn't take much extra effort. "Microsoft should develop a toolkit that covers all three from the same code base," I wrote at the time. "Either do it yourself or buy Xamarin as a jump-start. Do it quickly, though, before the army gets hungry and switches sides."

The purchase of Xamarin hasn't happened, but the companies have moved closer. Last April's Build 2014 conference showcased Xamarin front and center. And the recent Xamarin Evolve 2014 summit in Atlanta sold out in advance.

The integration of Xamarin with Visual Studio works exactly the way existing Visual Studio users would expect, and their skills transfer about as seamlessly as they could. It hasn't yet reached the level of, "Click one button, and out come iOS, Android and Windows versions." The three major mobile platforms are different enough that you can't abstract them completely. But Xamarin helps you separate your mobile app's business logic, which doesn't vary from one platform to another, from the presentation, which does. And it lets you develop the whole package with your familiar languages, such as C#, and tools such as debuggers. Just at press time, the company announced a new program making Xamarin free for students. It's moving in the right direction, and pretty quickly.

Xamarin asked me a few years ago if I was interested in starting up its developer training effort. I needed more cash than the company was willing to part with at the time, so the discussion never progressed, but I took its online training class as a student last spring and found it quite good. And if the instructors didn't quite have my flair (and let's face it, who does?), there were certainly more of them available live on multiple days in different time zones than just one of me. Now I sort of wish I'd held the price down to where Xamarin could swing it.

As if to trumpet its arrival in the big leagues, Xamarin has acquired former *MSDN Magazine* columnist Charles Petzold as an in-house guru, describing him in the press release announcing the hiring as a "Turing-complete engine" that "converts APIs into books" (see bit.ly/1o9tf7). His forthcoming book, "Creating Mobile Apps with Xamarin.Forms," is due in the spring of 2015. I learned Windows from his famous book ("Programming Windows") back in 1990. I'm looking forward to this one.

In "Why Software Sucks" (see pp. 193-194), I wrote about the early 1990s, when Microsoft was seen as the brash upstart, upending the established order of stodgy mainframes. I miss those days, as I miss my now-teenage daughters' first steps and first school and first sleepover parties. I'm looking forward to seeing some of that again as a grandparent. Is that what I'm seeing in the device world today, the now-grown child Microsoft, and the grandchild Xamarin? ■

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.

All New
Releases!



.NET Controls for the Professional Developer

ComponentOne delivers Data Visualization, UI, Spreadsheet and Reporting Controls for Microsoft Visual Studio Development

Reporting

ar ActiveReports

Easily create a variety of form-based, analytical and transaction reports. Visualize data with graphical components, design reports in the IDE & extend the reach of your reports with versatile viewers.

Data Visualization

Studio Enterprise

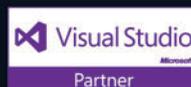
Hundreds of data and UI controls for all .NET platforms including grids, charts, reports and schedulers. Make building next-gen UIs in today's apps easier with an array of samples with source code.

Spreadsheet

sp Spread Studio

A Microsoft Excel-like spreadsheet control with the functionality of an advanced data grid, for adding power to any .NET application. Includes a formula engine and flexible tabular layout.

Download your free trials at
ComponentOne.com





**PREDICTIVE
ANALYTICS
MADE EASY FOR
DEVELOPERS**

Deploy predictive analytics solutions
within .NET applications.

- ★ Model using R[®], SAS[®], and SPSS[®] and deploy using .NET.
- ★ Deploy using rich client, web, services, and Hadoop.
- ★ No runtime dependency on the modeling environment.
- ★ Absolutely no deployment fees.

syncfusion.com/predictiveanalytics



Contact us to learn more:
1-888-9-DOTNET
sales@syncfusion.com

 **Syncfusion[®]**



FROM THE ORGANISERS OF SOFTWARE ARCHITECT

DEVWEEK 2015

Monday 23 – Friday 27 March | Central Hall, Westminster, London

18th SUCCESSFUL YEAR!

THE DEFINITIVE DEVELOPER CONFERENCE

World-renowned speakers, including

ALLEN HOLUB



KEVLIN HENNEY



DINO ESPOSITO



NEAL FORD



and many more

Keep your whole development team's skills up to date with the latest technologies, best practices and frameworks

- 23 FULL-DAY WORKSHOPS
- 105 90-MINUTE BREAKOUT SESSIONS

➤ BOOK BY FRIDAY 30 JANUARY 2015
SAVE UP TO £200

GOLD SPONSORS:



SPONSORS:



MEDIA PARTNERS:



Agile | Architecture | BI | Big Data | Cloud | Database | DevOps | IoT | Leadership | Mobile | MQ | MS Tech
Patterns | Programming Languages / Techniques | Security | Software Design | Testing | UI/UX | Web



DON'T MISS THE UK'S LEADING DEVELOPER CONFERENCE

DevWeek 2015 is the UK's number one destination for software developers, architects and analysts. With insights on the latest technologies, best practice and frameworks from industry-leading experts, plus hands-on workshop sessions, DevWeek is your chance to sharpen your skills – and ensure every member of your team is up to date.

23 full-day workshops
105 90-minute breakout sessions
49 expert speakers
Plus keynote presentations from Allen Holub and Kevlin Henney

VENUE

IN THE HEART OF LONDON

For five full days, from Monday 23 March to Friday 27 March, DevWeek takes over Westminster's iconic Central Hall.

Built in 1912, Central Hall is one of the UK's oldest purpose-built conference centres. Offering easy access to public transport and the restaurants and theatres of the West End, the venue fuses architectural grandeur with state-of-the-art facilities – making it the ideal setting for the UK's premier developer conference.

For more information about Central Hall, visit www.c-h-w.com



NEAREST TUBE STATIONS:

Westminster
(District, Circle & Jubilee)
St James's Park
(District & Circle)



ENJOY YOUR CONFERENCE, YOUR WAY

With shareable tickets and online catch-up, DevWeek gives you all the information you need, when you want it – ensuring you and your team make the most of every session.

Share your ticket

Only have time to attend one day? Get great value out of DevWeek by sharing your ticket with others in your team. That way, you can make the most of five packed days of sessions and workshops. Our online registration page lets you add colleagues' details so we know who will be joining us for each day of DevWeek.

Never miss a session

So much content is packed into DevWeek's five days, there are bound to be times when you wish you could be in two places at once. But you needn't miss out: all our sessions are filmed (subject to speaker approval) – and as a registered delegate, you'll have exclusive access to the whole event online to watch when you want.

Topics

- Agile
- Architecture
- BI
- Big Data
- Cloud
- Database
- DevOps
- IoT
- Leadership
- Mobile
- MQ
- MS Tech
- Patterns
- Programming Languages / Techniques
- Security
- Software Design
- Testing
- UI/UX
- Web



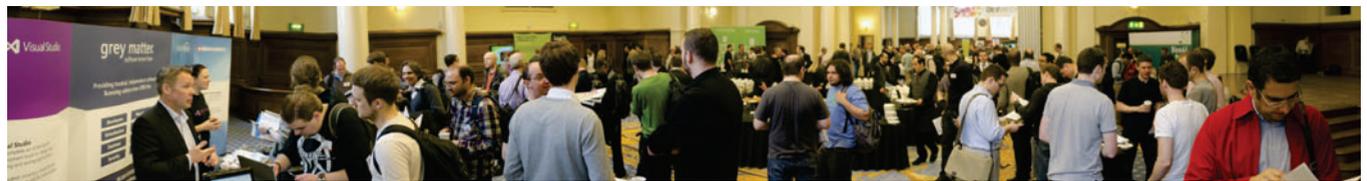
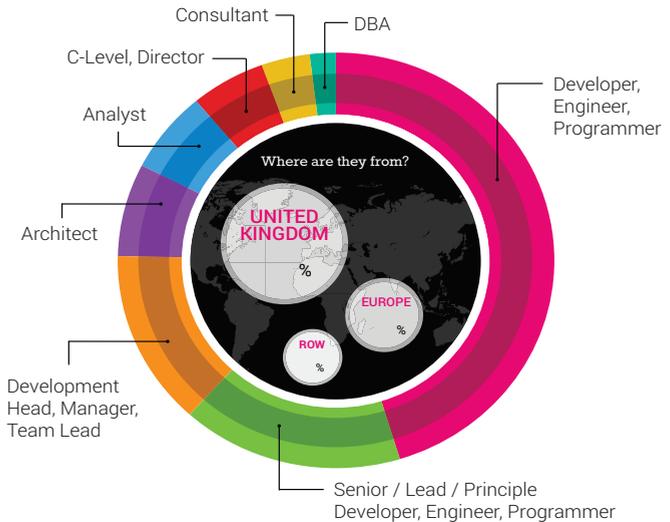
SPONSORSHIP & EXHIBITION

Do you provide services or technologies to the developer community? Let DevWeek work for you

DevWeek is the UK's leading event for software developers, DBAs and IT architects. As an exhibitor or sponsor, we can help you reach an engaged audience of professionals with a package that's custom-designed to suit your needs.

To discuss the wide range of sponsorship and exhibition opportunities available, contact **Chris Handsley**
+44 (0)207 830 3634
chris.handsley@publicis-blueprint.co.uk

WHO WILL YOU MEET AT DEVWEEK?



GOLD SPONSORS:



SPONSORS:



MEDIA PARTNERS:



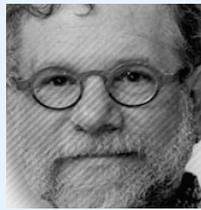


PRE-CONFERENCE WORKSHOPS

DevWeek's programme of pre-conference workshops covers a wide range of subjects, from Agile systems to app development.

All workshops run for a full day, from 09.30 to 17.30, with short breaks in the morning and afternoon, and a lunch break at 13.00.

09.30 - 17.30



ALLEN HOLUB

AGILE/DESIGN FROM START TO FINISH

Agile systems are, by necessity, tightly coupled to the user's notion of what the system is doing. Without that connection, the software can't stand up to the stress of constant change that all Agile processes mandate. Moreover, the process you use influences the architecture of your system. That's why hybrid processes that mix Agile and traditional practices often fail. The hybrid architectures that come out of those processes are unworkable.

In this workshop, Allen will talk about both process and architecture. We'll look at how Agile processes work, and see how both architecture and low-level design naturally fall out of those processes. Specifically, we'll examine the role of stories: how they're created and developed, and how they flow through the process, with a focus on developing an optimal architecture that closely mirrors both the stories themselves and the assumptions that underlie the stories.

We'll also work through a real-world example of the process from requirements gathering and problem-statement definition, to story development (use-case analysis) and the simultaneous construction of lightweight dynamic and static models that underlie the code.

WORKSHOP REF: DW01



PEARL CHEN

INTERNETTING YOUR THINGS

From big consumer success stories, such as the Nest Thermostat, to more offbeat monitoring systems, such as Botanicalls (which lets your plant call you when it is thirsty), your *things* are finding their own voice through small but powerful embedded microcontrollers. The market for the Internet of Things (IoT) and wearables is exploding. But what are your options for getting physical things when you're more used to writing software? In this workshop, Pearl explores a few hardware options that are great for hobbyist and rapid prototyping.

You'll get hands-on time to choose from some of the available hardware platforms, from a standard Arduino to an Intel Edison, MaKey MaKey, or LightBlue Bean. You'll be given a self-paced guide on how to set up the development environment on your computer and get a "Hello World" program running (typically an onboard blinking light). Once you have a feel for the technology, we'll go through a project brainstorming exercise. Finally, you'll be let loose to build your own hardware-based prototype in groups of 2-4. The workshop will be capped at 30 participants so expect a very team-oriented day.

WORKSHOP REF: DW02



SAHIL MALIK

JAVASCRIPT: BEYOND THE BASICS

In this workshop, Sahil will teach you advanced concepts in JavaScript, allowing you to structure your code well and learn JavaScript beyond the basics we have been tinkering with.

This workshop teaches you not to just *know* JavaScript, but to be *good at* JavaScript. There are plenty of hands-on labs, which will walk you through objects, prototypes, scopes, this variable, debugging, performance and best practices.

We will begin with the very basics of JavaScript: variables, language syntax, referencing files, loops, conditions, built-in functions and custom functions, arrays and so on. We'll move on to talk about functions as expressions, closures to structure your code, the concept of nested scopes, and the confusion around "this". We'll learn all about objects and prototypes, and how you can mimic things in JavaScript that are usually reserved for higher-level languages.

We'll solidify that knowledge with some best practices and debugging tricks. We'll also explain how to avoid common pitfalls, and how to organise your code in modules to keep it maintainable and understandable. Lastly, there's performance – let's not forget performance!

WORKSHOP REF: DW03



JAMES MONTEMAGNO

REUSING YOUR .NET AND C# SKILLS TO DEVELOP NATIVE APPS FOR WINDOWS, IOS AND ANDROID

With .NET, Xamarin and portable class libraries, you can now create native apps that target iOS, Android and Windows without compromising on performance, user experience or developer productivity. And you can do it all within Visual Studio IDE or Xamarin Studio IDE for Mac or PC.

James will introduce the Xamarin platform, including building native iOS and Android apps with C#. We'll look at the fundamentals of how each platform works, and deep-dive into platform-specific functionality. Designing iOS and Android apps could not be easier with Xamarin's integrated iOS and Android designers for both Visual Studio and Xamarin Studio. You'll find out how to use both of these designers to craft unique user interfaces for each platform. We'll also take a look at sharing code with Universal Windows apps, enabling you reach all platforms from a single shared C# code base.

You'll gain a full understanding of the Xamarin platform and how to build iOS and Android apps using C#, as well as a solid introduction to code reuse techniques, plus lots of sample code to take home!

WORKSHOP REF: DW04

“
Excellent mix of subjects – would recommend unreservedly
”

SOFTWARE DEVELOPER
2014 DELEGATE

**ROBERT SMALLSHIRE
& AUSTIN BINGHAM****THE POWER OF
REVIEW**

Review by peers, colleagues, experts and stakeholders is perhaps the most effective tool we have for improving the quality of software. But if review is so wonderful, why is it used so infrequently?

In this workshop, Robert and Austin will show you how to conduct effective code, design and requirements reviews using a variety of techniques from the relatively informal sort of reviews you're perhaps doing already, through to the most formal inspections. We'll work together to understand what makes a good review, and help you to identify behaviours that lead to poor outcomes, in the form of either defective software or unhappy colleagues.

Throughout this workshop, you'll receive plenty of advice on how you can introduce effective technical reviews into your engineering culture.

**WORKSHOP
REF: DW05****JULES
MAY****F#: THE LITTLE
LANGUAGE WITH
A LOT OF BITE**

Since it was released in 2008, F# has inspired more and more excitement. It's a functional language, like Haskell or ML, and yet it's built on top of .NET, so it leverages all the mature object-oriented power that the platform provides. The result is neither purely functional, nor obviously object-oriented, but is something entirely new: a unique and powerful mixture of the two.

Those programmers who have already discovered F# find they can write code that is shorter, faster, vastly more reliable, and delightfully reusable. This is no research language – it's being used on real, large-scale projects, and it has the backing of the F# Software Foundation and Microsoft.

If you want to understand what the fuss is about, this is the day for you. In this workshop, Jules will touch on what makes F# programming so special, and show where it gets its power from. We'll write some functional code, we'll write some of F#'s unique take on object-oriented code, and we'll write some code that only F# can do.

**WORKSHOP
REF: DW06****IDO
FLATOW****SECURING ASP.
NET APPLICATIONS
AND SERVICES:
FROM A-Z**

When you think of ASP.NET security, the first things that come to mind are Windows authentication and forms authentication using ASP.NET Membership. For years, those were the common authentication techniques for ASP.NET applications and services. But with the new releases to the ASP.NET Identity system, those days are long gone.

For the enterprise, ASP.NET broadened its support from the on-premises Active Directory to include Microsoft Azure Active Directory. By supporting external identity providers, such as Facebook, Microsoft Account and Twitter, the new ASP.NET Identity system makes the process of securing an application less scary than ever.

In this workshop, Ido will start from the basics of getting to know concepts such as SSL, OAuth, OpenID and claim-based authorisation. From there we will continue to explore the various scenarios of using self-managed identities, Active Directory and ADFS, external identity providers (Facebook, Google, Microsoft) and Microsoft Azure Active Directory.

**WORKSHOP
REF: DW07****KLAUS
ASCHENBRENNER****SQL SERVER QUERY
TUNING FOR
DEVELOPERS**

Are you a developer writing T-SQL queries for SQL Server databases? Maybe you're already mastered the basics of T-SQL, but want to reach a higher level in T-SQL to write better performing queries? In this workshop, Klaus will take a full day to talk about how to improve your T-SQL skills to solve complex problems, and how to further speed up your queries by applying a good indexing strategy to your T-SQL queries.

In the workshop, we'll cover four modules: query processing basics (set theory, predicate logic, relational models and logical query processing), physical query processing (execution plans, data access paths, physical join operators, aggregation operators and spool operators), temporary data and aggregations (temp tables, table variables, common table expressions, and aggregations and pivoting), and working with windowing functions (window aggregate functions, ranking functions, distribution functions, offset functions, query tuning guidelines and parallelism optimisations).

**WORKSHOP
REF: DW08****DINO
ESPOSITO****ONE DAY OF
TWITTER
BOOTSTRAP**

As a responsive web framework, Twitter Bootstrap is leading the world of web development today, setting new standards and capturing followers. In this workshop, Dino will provide a day-long tour of the library and delve deep into its HTML chunks, CSS styles and JavaScript components.

We'll focus on facilities available for building static and responsive layouts, rich input forms and advanced features such as auto-completion, modals, tabs, carousels and more. We'll also consider the downsides of the framework, missing pieces (ie. image handling) and its overall role in the broader context of responsive and device-friendly sites. This workshop is ideal for clearing up a few things you may already have heard about Bootstrap, or just for gaining an additional perspective about it. In any case, after this workshop you should be ready to get into it at any level of further complexity.

**WORKSHOP
REF: DW09****SEB
ROSE****BDD BY EXAMPLE**

In this workshop, Seb will provide a practical introduction to using examples to specify software. You'll learn to break down complex business requirements with your stakeholders, using examples in their own language, giving you the tools you need to explore their ideas before you even write any software.

This workshop is for everybody involved in the process of developing software, so please bring product owners, testers and architects along. As well as describing what BDD is (and isn't), we'll spend a lot of time practising collaborative analysis to make sure that our stories are appropriately sized, easy to read and unambiguous. We'll develop a "ubiquitous language", explore the workings of the Three-Amigos meeting, and really get to grips with the slippery interaction between features, stories, acceptance criteria and examples.

We'll use pens, cards and other bits of paper, so you won't need to know any tools in advance, or even remember your laptop!

To build further on these ideas, don't miss Seb's post-conference workshop: *"Applied BDD with Cucumber, Cucumber-JVM and SpecFlow"*.

**WORKSHOP
REF: DW10**

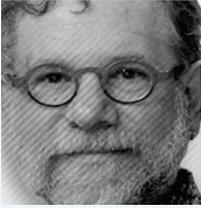


DevWeek’s first day of sessions kicks off with our keynote presentations, giving all attendees the chance to hear industry experts Kevlin Henney and Allen Holub tackle two of the biggest issues in software development.

DAY 1 AGENDA

09.30	 <p>KEVLIN HENNEY</p> <p>KEYNOTE PRESENTATION</p>	<p>A SYSTEM IS NOT A TREE</p> <p>Trees. Both beautiful and useful. But we’re not talking about the green, oxygen-providing ones. As abstract structures we see trees all over the place – file systems, class hierarchies, ordered data structures, and so on. They are neat and tidy, nested and hierarchical – a simple way of organising things; a simple way of breaking large things down into small things.</p> <p>The problem is, though, that there are many things – from modest fragments of code up to enterprise-wide IT systems – that do not comfortably fit into this way of looking at the world and organising it. Software architecture, design patterns, class decomposition, performance, unit tests... all of these cut across the strict hierarchy of trees. In this keynote, Kevlin will look at what this means for how we think and design systems, whether large or small.</p>
--------------	--	---

11.30	<p>ROBERT SMALLSHIRE</p> <p>UNDERSTANDING TRANSDUCERS Transducers – a portmanteau of “transform reducers” – are a new functional programming concept introduced into the Clojure programming language. Although transducers are actually pretty straightforward, wrapping your brain around them, especially if you’re not already a competent Clojureist, can be challenging. In this session, Robert will introduce transducers by implementing them from scratch in everybody’s favourite executable pseudocode: Python. We’ll start with the familiar staples of functional programming and derive transducers from first principles.</p>	<p>SASHA GOLDSHTEIN</p> <p>AUTOMATING PROBLEM ANALYSIS AND TRIAGE What do you do when your application crashes or hangs in production? Nothing can compete with a debugger or a full process dump captured on a production system. But you can’t always afford the time to analyse hundreds of crash dumps. In this session, Sasha will show you how to perform automatic dump analysis and triage using Microsoft’s CLRMD, a .NET library that can explore threads, call stacks and exceptions; visualise threads and locks to form wait chains and detect deadlocks; and walk heap memory to inspect important objects for your application.</p>	<p>JOHN K. PAUL</p> <p>ECMASCRIPT 6 FOR ALL OF US Coming from a Java background, there was a time when JavaScript was nothing but annoyances. Now, even after we’ve grown to love the language, there are dozens of times when we feel the pain of missing features that Java has built in. ECMAScript 6 is changing all of that. The next version of JavaScript brings with it an amazing standard library that rivals that of Java, Python and their ilk. In this session, John will explain some of the great new additions to the language and demonstrate use cases that take advantage of ES6’s elegance for client-side development.</p>	<p>DROR HELPER</p> <p>NAVIGATING THE TDD ALPHABET SOUP TDD, BDD, ATDD are all methodologies that enable incremental design that is suitable for Agile environments. It seems that every day a new xDD methodology is born with the promise of being better than what came before. Should you use behaviour-driven tests or plain old unit tests? Which methodology is better? And how exactly would it benefit the development life cycle? In this session, Dror will help to sort out the various methodologies – explaining where they came from, the tools they use, and discussing how and when to use each one.</p>	<p>ALLEN HOLUB</p> <p>SECURITY 101: AN INTRODUCTION TO SOFTWARE SECURITY As more and more of our applications move on to the web, security becomes even more critical. Good security, however, has to be built in, not tacked on as an afterthought. In this session, Allen will give you an overview of what it means to make an application secure. He’ll cover topics such as security architectures, code and design review, penetration testing, risk analysis and risk-based testing, security-related requirements, static analysis, abuse cases, security operations and crypto.</p>
--------------	---	--	---	--	--



ALLEN
HOLUB

KEYNOTE
PRESENTATION

#NOESTIMATES

Estimates are always guesses – and they’re always wrong. Consequently, estimate-based planning is foolhardy at best, and time spend creating them is a waste. In spite of this fact, estimates are a central part of most software-development processes, even some Agile processes. Getting rid of estimates doesn’t mean that you can’t plan, but you do have to go about planning in a more effective way. In this keynote, Allen will discuss both the problems surrounding an estimation culture and how to solve those problems by using actual measurements and priority-based planning.



<p>KLAUS ASCHENBRENNER</p> <p>SQL SERVER 2014 IN-MEMORY OLTP DEEP DIVE (HEKATON) Hekaton is the Greek word for 100 – and the goal of In-Memory OLTP in SQL Server 2014 is to improve query performance up to 100 times. In this session, Klaus will look inside the case of Hekaton and the multiversion concurrency control (MVCC) principles on which Hekaton is built. He'll start by looking at the challenges that can be solved by Hekaton, especially locking, blocking and latching within SQL Server. Based on this foundation, he'll move into the principles of MVCC, and how a storage engine and transaction manager can be built on that concept.</p>	<p>DEJAN SARKA</p> <p>INTRODUCING R AND AZURE ML R is a free software programming language and software environment for statistical computing, data mining and graphics. Azure Machine Learning (Azure ML) is the new Microsoft cloud service and environment for advanced data analysis, which utilises the R algorithms intensively. In this session, Dejan will introduce both R, with RStudio IDE, and Azure ML.</p>	<p>ALLAN KELLY</p> <p>DIALOGUE SHEETS FOR RETROSPECTIVES AND DISCUSSION Retrospectives are a key tool in the Agile toolkit, but they aren't easy. In fact, it's not just retrospectives. Teams need to learn to talk, discuss and reflect together over many things. Good conversation makes for good software. Retrospective dialogue sheets can help overcome these problems. In this hands-on session, in which everyone will get the chance to work with a dialogue sheet, discovering what one is and how to use it, Allan will discuss some of the ways teams are using them and look to the future.</p>	<p>MARK SMITH</p> <p>DESIGNING ADAPTIVE APPLICATIONS FOR THE IOS PLATFORM With the introduction of the iPhone 6 and 6+, we now have several form factors to consider when designing our iOS applications. In this session, Mark will focus on the designer and layout features Apple has included in iOS to help you create a single, unified storyboard that is capable of working with all variations of iOS. This will include layout constraints (auto layout), size classes, unified storyboards and the updated UISplitViewController.</p>	<p>SEB ROSE</p> <p>LESS IS MORE – AN INTRODUCTION TO LOW-FIDELITY APPROACHES In this session, Seb will demonstrate some key techniques that help decompose large problems. Decomposing problems is a skill all software developers need, but we're often not very good at. Whether it's stories that take longer than an iteration, or features that can't be delivered in the expected release, we've all seen the problems that tackling an over-large problem can cause. We'll work through two detailed examples to demonstrate the value of delivering small, low-fidelity pieces of work early rather than prematurely focusing on fully-polished final version.</p>	<p>MICHAEL KENNEDY</p> <p>GETTING STARTED WITH SWIFT (APPLE'S NEWEST LANGUAGE) Swift is Apple's newest language for building native, high performance applications for both iOS and OS X. This session will introduce you to this exciting language. Developers with a background in either C# or Python will see many similarities. Almost everyone will find Swift a much more comfortable and inviting language when compared to Objective C or C. Come and learn why it's now fun to develop (natively) for iOS and OS X.</p>
---	--	--	--	--	---



DAY 1 AGENDA CONTINUED

14.00	ADAM TORNHILL	TOBIAS KOMISCHKE	SHAY FRIEDMAN	RALPH DE WARGNY	DINO ESPOSITO
	<p>TREAT YOUR CODE AS A CRIME SCENE We'll never be able to understand large-scale systems from a single snapshot of the code. Instead, we need to understand how the code evolved and how the people who work on it are organised. We also need strategies that let us find design issues and uncover hidden dependencies between both code and people. Where do you find such strategies if not within the field of criminal psychology? In this session, Adam will use this approach to predict bugs, detect architectural decay and find the code that is most expensive to maintain.</p>	<p>THE PROMISED LAND (OF UX) In today's competitive landscape, a stellar user experience is a strong product differentiator and enabler of market success. Companies hope to get to this "Promised Land" where their own staff deploys mature UX design practices, their customers are happy, and their market share and profits increase. No one ever said that this journey to the Promised Land is easy – a lot of companies have tried and failed. In this inspirational overview session, Tobias will show the path and identify what strategic elements are critical to successful design.</p>	<p>ANGULARJS - THE ONE FRAMEWORK TO RULE THEM ALL In the last couple of years, we've seen the rise of client-side JavaScript frameworks. From almost nothing, now we have at least a dozen to choose from. One of the new kids in the block, AngularJS, comes straight from the Google offices and tries to stand out from the crowd with a complete set of tools and utilities. Some are very powerful and try to ease your way towards your SPA web site. In this session, Shay will go through the different features of AngularJS and see what makes it one of the most popular JavaScript frameworks out there.</p>	<p>CODING THE PARALLEL FUTURE IN C++ Many-core processors and computing platforms will be ubiquitous in the future: from multi-core and many-core CPUs to integrated GPUs to compute clusters in the cloud, it's the new parallel universe for software developers. In this session, you will learn the tools, techniques and best practices available to C/C++ developers to make sure your code is ready today to run with maximum performance and reliability in this new parallel universe.</p>	<p>ASP.NET IDENTITY, CLAIMS AND SOCIAL AUTHENTICATION ASP.NET Identity is the new and comprehensive membership system for the whole ASP.NET platform, including Web API and SignalR. Similar in many ways to the popular simple membership provider, ASP.NET Identity goes well beyond in a number of aspects: replaceable storage, flexible representation of user profiles, external logins, claims-based authentication and role providers. Dino will use the Identity API to set up social authentication via Facebook and Twitter, and to collect any user information made available by social networks.</p>

16.00	ED COURTENAY	PEARL CHEN	PHIL LEGGETTER	OREN RUBIN	SASHA GOLDSHEIN
	<p>INVERSION OF CONTROL 101 The dependency injection/inversion of control design pattern is an important technique that helps to write testable and maintainable code. In this session, Ed will debunk the myth that it's hard to understand or only for "enterprise development", and demonstrate how to use it in everyday code. This demo-led session will discuss the rationale behind dependency injection, demonstrate injection with and without a dependency container, as well as writing a simple container from scratch.</p>	<p>INTERNETTING YOUR THINGS From big consumer success stories, such as the Nest Thermostat, to more offbeat monitoring systems, such as Botanicalls (which lets your plant call you when it is thirsty), your <i>things</i> are finding their own voice through small but powerful embedded microcontrollers. The market for the Internet of Things (IoT) and wearables is exploding. But what are your options for getting started with making physical things when you're more used to writing software? In this session, Pearl will go through a few hardware options that are great for hobbyists and rapid prototyping.</p>	<p>PATTERNS AND PRACTICES FOR BUILDING ENTERPRISE SCALE HTML5 APPS Developing large apps is difficult. Ensuring that code is consistent, maintainable, testable and has an architecture that enables change is essential. When it comes to large server-focused apps, solutions to some of these challenges have been tried and tested. But, how do you achieve this when building HTML5 single-page apps? In this session, Phil will highlight signs to watch out for as your HTML5 SPA grows, and patterns and practices to help you avoid problems. He will also explain the architecture that their HTML5 apps have that is core to ensuring they scale.</p>	<p>TEST AUTOMATION DONE RIGHT: THE HOLY GRAIL OF CONTINUOUS DEPLOYMENT In this session, Oren will explain all there is to know about end-to-end test automation. Starting with the basics and a comparison to unit testing, he will then drill down into what is considered uncharted territory for many developers. You will learn the best practises and design patterns, common pitfalls, and – most importantly – the full ecosystem and how it connects to your existing toolchain. You will learn about different approaches to UI verifications, and see real industry use cases and bugs.</p>	<p>WHAT'S NEW IN VISUAL STUDIO 2015? In this session, Sasha will lead us through a sample of the new Visual Studio 2015 features, ranging from developer productivity to low-level C++ code optimisations. In a series of quick-paced demos, we will show how Visual Studio 2015 makes key diagnostics experiences easier, improves IntelliTrace analysis, helps developer collaboration and productivity, and produces higher-quality and faster code for both managed (with .NET Native) and C++ applications. If you're using Visual Studio, you can't afford to miss this talk!</p>



JOIN US FOR DRINKS AT THE END OF DAY 1 – NETWORKING DRINKS SPONSORED BY



<p>MICHAEL KENNEDY</p> <p>SCALING THE NOSQL WAY WITH MONGODB The great promise of the NoSQL databases has been their ability to scale out rather than scaling up. In this session, Michael will look at a concrete example of scaling one of the most generally useful and most widely deployed NoSQL database: MongoDB. He'll explore why you might need to scale out, and you'll see the full spectrum of choices for scaling (replication, sharding, geo-replication, etc). NoSQL document databases typically outperform RDBMSes on single servers but with the ability to scale out they can truly achieve an entirely new level of performance.</p>	<p>IQBAL KHAN</p> <p>LEARN HOW TO SCALE .NET APPS IN MICROSOFT AZURE WITH DISTRIBUTED CACHING Discover the scalability bottlenecks for your .NET applications in Microsoft Azure, and how you can improve their scalability with distributed caching. This session provides a quick overview of scalability bottlenecks, and answers some key questions: What is distributed caching and why is it the answer in Microsoft Azure? Where in your application can you use distributed caching? What are some important features in a distributed cache? You'll also see hands-on examples of using a distributed cache.</p>	<p>NEAL FORD</p> <p>BUILDING MICROSERVICE ARCHITECTURES Inspired by the success of companies such as Amazon and Netflix, many organisations are moving rapidly towards microservice architectures. This style of architecture is important because it's the first architecture to fully embrace the Continuous Delivery and DevOps revolutions. In this session, Neal will cover the motivations for building a microservice architecture, some considerations you must make before starting (such as transactions versus eventual consistency), how to determine service partition boundaries, and ten tips for success.</p>	<p>NUNO GODINHO</p> <p>EVENT HUBS, AZURE STREAMING AND AZURE ISS Recently, there has been a lot of talk around IoT, M2M, big data and similar topics. It's important to understand how we can take advantage of these concepts, and how they can help us achieve our goals. Fortunately, Microsoft has some solutions for us. These are Azure Event Hubs, Azure Stream Analytics and Azure Intelligence Systems Service (ISS). In this session, Nuno will explore these three topics, demonstrate their interconnectivity, and show how they provide the perfect answer for our next-generation solutions and interactions.</p>	<p>JULES MAY</p> <p>IF CONSIDERED HARMFUL: HOW TO ERADICATE 95% OF ALL YOUR BUGS IN ONE SIMPLE STEP In 1968, CACM published a letter from Edgar Dijkstra, called "Go To statement considered harmful". In it, he explained why most bugs in programs were caused by Gotos, and appealed for Goto to be expunged from programming languages. But Goto has a twin brother, which is responsible for nearly every bug that appears in our programs today. That twin is If. In this session, Jules revisits Dijkstra's original explanation to show why If and Goto have the same pathology, and how you can avoid it.</p>	<p>SAHIL MALIK</p> <p>TOP 10 JAVASCRIPT TIPS JavaScript, the <i>lingua franca</i> of the web, is incredibly freeform and therefore hard to get right. We've all hacked JavaScript, but what do you need to know when you are doing big and complex JavaScript projects? This isn't your Dad's browser, y'know! So you have written JavaScript, but want to go beyond the basics? In this session, Sahil will show you the JavaScript concepts that every modern JavaScript developer needs to know.</p>
<p>MIKE WOOD</p> <p>MESSAGING PATTERNS There are many reasons why asynchronous messaging should be introduced in applications, as well as many approaches in incorporating messaging subsystems. In some cases, intensive workloads need to be pushed to back-end processing, or perhaps specialised (and often expensive) resources need to be utilised to perform certain operations. In this session, Mike will cover several scenarios where introducing messaging can help, discuss a few messaging patterns, and look at abstracting your messaging subsystem to guard against evolving technology and designs.</p>	<p>GARY SHORT</p> <p>DATA SCIENCE FOR FUN AND PROFIT Make no mistake: data science can be hard, but it can also be fun. In this session, Gary will introduce you to classic and Bayesian statistics and machine learning, all through the medium of predicting horse-racing results. He'll explore a number of techniques for making such predictions and finish by combining them into a powerful "mixed model" prediction engine that's sure to pick the next big winner. This session won't only improve your knowledge, it'll improve your bank balance too! (Note: Session may not improve bank balance.)</p>	<p>AUSTIN BINGHAM</p> <p>HIGH-QUALITY DECISION MAKING WITH OPEN DESIGN PROPOSALS Making complex decisions in software design involves balancing many factors, and maintaining that balance can be challenging. By opening up the decision process for evolution, we can harness the insight of fellow developers, communicate plans and designs more effectively, and produce a useful record of the work we do. In this session, Austin will look at a specific technique: Open Design Proposals. He'll examine implementations of this approach, see why it's effective, and show how development teams can use it to manage their own decision making.</p>	<p>JAMES MONTEMAGNO</p> <p>IBEACONS AND CONTEXTUAL LOCATION AWARENESS IN IOS AND ANDROID APPS iBeacons are taking the world by storm – from retail stores to major sporting events, you'll soon be finding iBeacons just about everywhere. This gives you the ability to enable any number of device proximity-based scenarios that were never before possible. In this session, James will explain what an iBeacon is, how they work, how you would want to use them, and how to get started making apps in both Android and iOS. All demonstrations will be coded in C#, but will be applicable to any iOS or Android developer in any language.</p>	<p>KEVLIN HENNEY</p> <p>FP 4 OOP FTW! Although not yet fully mainstream, functional programming has finally reached a critical mass of awareness among developers. The problem, however, is that many developers are up against an even greater critical mass of existing code. Much of this code purports to be object oriented, but beyond the use of the class keyword, falls somewhat short of putting the OO to good use. Many techniques identified as functional have broader applicability. In this session, Kevlin will explore how some FP habits can be incorporated into OOP practice.</p>	<p>SAHIL MALIK</p> <p>TOP 10 JAVASCRIPT DEBUGGING TRICKS We've all been writing lots of JavaScript code lately. But JavaScript is incredibly free form, and that sharp double-edged sword can also make debugging JavaScript errors a lot more difficult. The new operating system is the browser, and complex JavaScript pages cannot ignore performance, or their unpredictable behaviour under different bandwidths. In this session, Sahil will show you some really useful debugging techniques and demonstrate how to use each browser for its best capabilities. Ninety minutes spent here will save you hours in your day job.</p>



Day 2 of DevWeek continues with a packed programme of workshops and breakout sessions – and don't forget that if you can't make it to any of the talks you're interested in, as a registered delegate you'll be able to catch up later online at the DevWeek website.

DAY 2 AGENDA

<p>09.30</p>	<p>ADAM TORNHILL</p> <p>CODE THAT FITS YOUR BRAIN People think, remember and reason in a very different way from that in which code is presented. So how should code look to make it both easier to understand and maintain? To see what really works, we need to look across languages and paradigms. In this session, Adam will start with common problematic constructs such as null references, surprising corner-cases and repetitive code, and discuss the cognitive costs and consequences of each. He'll then apply ideas from object-orientation, functional programming and lesser-known array languages to explore better approaches.</p>	<p>TOBIAS KOMISCHKE</p> <p>VISUAL DESIGN FOR NON-DESIGNERS: IT'S NOT JUST ABOUT COLOUR! In an ideal world, front-end developers don't need to worry about visual design because they get specs and assets from professionally trained designers. The reality is that developers often need to make their own decisions about how to make the UI attractive. In this session, Tobias will provide a solid base knowledge about what constitutes attractiveness and what design principles can be applied to boost the visual appeal of UIs. By the time you leave, you'll be ready to step up to the mark when designers are nowhere to be found!</p>	<p>PHIL LEGGETTER</p> <p>WHY YOU SHOULD BE USING WEB COMPONENTS – AND HOW Web Components are touted as the future of web development. In this session, Phil will explain what Web Components are, the state of native support in web browsers, what your options are for building componentised web apps right now using AngularJS, Ember, Knockout or React, and why Web Components probably <i>are</i> the future of web development. He'll also cover the benefits of a component-based architecture and how it helps when building JavaScript apps, as well as how components can communicate in a loosely coupled way, and why.</p>	<p>ED COURTNEY</p> <p>AN INTRODUCTION TO TYPESCRIPT JavaScript is the scripting glue that holds the web together – largely because of its flexibility. This flexibility also means that it can be difficult to manage, especially in large-scale applications. In this session, using real-world examples, Ed will explore some of the problems with client-side JavaScript development as a motivating example and introduce TypeScript as a way of solving some of these issues. He'll also explore how existing JavaScript codebases can be targeted by TypeScript and how it can be integrated into build systems.</p>	<p>SASHA GOLDSHTEIN</p> <p>COOL LIBRARIES FOR MODERN C++ The C++ standard library dates back to the 1990s, but that doesn't mean there aren't new and exciting frameworks to use in your C++ application. In this session, Sasha will look at some brand-new and some existing C++ libraries that can speed up C++ cross-platform development rapidly. Some of the libraries we might cover include: Casablanca (C++ REST SDK), Cinder (creative coding), Boost (general-purpose), Google Test (unit testing), SOCI (modern database access) and many others. There's something for everyone!</p>
<p>11.30</p>	<p>PAVEL SKRIBTSOV</p> <p>COMPUTING LIKE THE BRAIN: AN INTRODUCTORY GUIDE TO AI Every now and again, every professional developer faces a program that he or she has trouble writing. Try to imagine an algorithm that has to differentiate a dog from a cat. They come in different shapes and sizes, and there is no single feature that could discriminate between the two. Any attempt to code that algorithm manually using deep-nested "if/else" branches is doomed. Human beings, on the other hand, have no trouble with this task. In this session, Pavel will introduce the basics of an artificial intelligence-based approach to solving these problems.</p>	<p>JOE NATOLI</p> <p>THINK FIRST: WHY GREAT UX STARTS BETWEEN YOUR EARS (AND NOT ON THE SCREEN) When developers are tasked with improving UX, their focus tends to be on the screen: elements, interactions, workflow, often accompanied by the worrying cry, "I'm not a UI designer!" Fortunately, Joe has good news: you can still design great user experiences without a shred of visual design talent. In this session, Joe will show you how changing the way you think about features, functions and implementations can make a massive, positive change in the experience people have with your UI and your product.</p>	<p>ALLEN HOLUB</p> <p>KNOCKOUT: AN INTRODUCTION The Knockout framework is a standalone implementation of the MVVM (Model-View-ViewModel) pattern, which is one of the best user-interface architectures for web applications. It provides an alternative to AngularJS – more limited in scope but smaller and, in some contexts, faster. In this session, Allen will look at Knockout's architecture and how to leverage that architecture to build highly interactive web-application user interfaces. The session will include several code (JavaScript) examples.</p>	<p>DROR HELPER</p> <p>BATTLE OF THE .NET MOCKING FRAMEWORKS Writing unit tests is hard, isn't it? You need an entire set of tools just to start. One of the crucial decisions when building this set is picking up a mocking framework. But beware – the mocking framework you choose has the ability to make or break you! In this session, Dror – at one time a mocking framework developer – will cover the capabilities and functionality of the leading frameworks, showing the good and the bad of the different options (both free and commercial), and making them battle to the death!</p>	<p>SHAI REZNIK</p> <p>BUILD PRODUCTION-READY JAVASCRIPT APPS WITH GRUNT In this session, Shai will deliver an overview of the steps required in order to build JavaScript apps and get them ready for deployment. He'll cover build theory, asking "Why build in JS?" He'll also talk about the build steps and then jump to Grunt, explaining what it is and providing a live demo. Finally, Shai will cover the scaffolding tool, Yeoman. This session is intended to be both funny and informative, so get ready to have a good time while picking up some essential tips to make your day job that much easier.</p>

<p>KLAUS ASCHENBRENNER</p> <p>THE DANGEROUS BEAUTY OF BOOKMARK LOOKUPS</p> <p>You know Bookmark Lookups in SQL Server? You like their flexibility to retrieve data? If you do, then you should be warned that you are dealing with the most dangerous concept in SQL Server! Bookmark Lookups can lead to massive performance losses that will devastate your CPU and I/O resources! In this session, Klaus will provide a basic understanding of Bookmark Lookups and how they are used by SQL Server. After laying out the foundations, he'll talk in more detail about the various performance problems they can introduce.</p>	<p>DEJAN SARKA</p> <p>DATA MINING ALGORITHMS WITH SQL SERVER AND R (PART 1)</p> <p>Data mining is gaining popularity as the most advanced data analysis technique. With modern data mining engines, products and packages, such as SQL Server Analysis Services (SSAS) and R, data mining has become a black box. It's possible to use data mining without knowing how it works, but this can lead to many problems, such as using the wrong algorithm for a task, misinterpretation of the results and more. In this session (and Part 2, at 11.30), Dejan will explain how the most popular data mining algorithms work and when to use each one.</p>	<p>NEAL FORD</p> <p>CONTINUOUS DELIVERY FOR ARCHITECTS</p> <p>Yesterday's best practice is tomorrow's anti-pattern. Architecture doesn't exist in a vacuum: a painful lesson developers who built logically sound but operationally cumbersome architectures learned. Continuous Delivery is a process for automating the production-readiness of your application every time a change occurs to code, infrastructure or configuration. In this session, Neal will take a deep dive into the intersection of the architect role and the engineering practices in Continuous Delivery.</p>	<p>JAMES MONTEMAGNO</p> <p>IOS AND ANDROID DEVELOPMENT FOR C# DEVELOPER WITH XAMARIN</p> <p>As the mobile landscape continues to expand and evolve, managing multiple codebases in different programming languages and development tools can quickly become a nightmare. Wouldn't you love to build native UIs for iOS, Android and Windows Phone from a single codebase? In this session, James will show how to leverage the awesome features of C# and combine them with Xamarin technology to create beautiful, native, cross-platform, mobile apps from a shared C# codebase, with the tools that you love.</p>	<p>SAHIL MALIK</p> <p>LEARN ANGULARJS: THE ROAD TO POWERFUL, MAINTAINABLE APPLICATIONS</p> <p>JavaScript, by its nature, makes it difficult to write maintainable code. HTML, by its nature, is loosely structured. AngularJS fixes both of those. It's a structural framework for dynamic web apps, allowing you to extend HTML's syntax, enabling you to write powerful, maintainable applications succinctly. In this workshop, Sahil will build on your existing knowledge of JavaScript and teach you the ins and outs of AngularJS. There are plenty of examples, which will walk you through a basic introduction, models, controllers and views in Angular; templates and databinding, services and dependency injection, directives, routing and single-page applications.</p> <p><i>For a full description of the workshop, please see Page 20</i></p>	<p>ANDY CLYMER & RICHARD BLEWETT</p> <p>SOLID ASYNC PROGRAMMING IN .NET</p> <p>In this special two-day workshop, Andrew and Richard will take you through the core skills required to successfully develop async and multithreaded code, both in the .NET and web worlds. Not only do we cover the core APIs, but also how they are used effectively, tested and debugged.</p> <p><i>For a full description of the workshop, please see Page 20</i></p>
<p>MICHAEL KENNEDY</p> <p>APPLIED NOSQL WITH MONGODB AND PYTHON</p> <p>NoSQL is a hot topic in the tech industry today. But what exactly is NoSQL and should I use it to build my next application? In this session, Michael will dig into why NoSQL databases are sweeping the industry and discuss the trade-offs between the various types (key-value stores vs document databases, for example). He will explore the most broadly applicable variant of NoSQL, document databases, through hands-on demos with the most popular and successful of the document databases, MongoDB.</p>	<p>DEJAN SARKA</p> <p>DATA MINING ALGORITHMS WITH SQL SERVER AND R (PART 2)</p> <p>Data mining is gaining popularity as the most advanced data analysis technique. With modern data mining engines, products and packages, such as SQL Server Analysis Services (SSAS) and R, data mining has become a black box. It is possible to use data mining without knowing how it works, but this can lead to many problems, such as using the wrong algorithm for a task, misinterpretation of the results and more. In this session (following on from Part 1, at 09:30), Dejan will explain how the most popular data mining algorithms work and when to use each one.</p>	<p>DINO ESPOSITO</p> <p>DDD MISCONCEPTIONS</p> <p>For too long, domain-driven design (DDD) has been sold as the ideal solution for very complex problems that only a few teams are actually facing. While technically correct, this statement sparked a number of misconceptions. In fact, DDD is only an approach to the design of software systems and is driven by the domain of the problem. In this session, Dino will clear the ground around DDD, emphasising the theoretical pillars of the approach: ubiquitous language and bounded context.</p>	<p>MIKE WOOD</p> <p>5 LIGHTWEIGHT MICROSOFT AZURE FEATURES FOR FAST-MOVING MOBILE DEVS</p> <p>Mobile development has exploded, and everyone has an idea they want to try out. But bootstrapping a mobile app doesn't always seem that easy. Consumers demand slick user experiences and the ability to share data across a plethora of devices and platforms, while we're trying to get a minimal viable product out the door to test our ideas as fast as possible. Thankfully, Azure has powerful features available to help. In this session, Mike will take a practical look at five features of Azure that are useful for mobile developers of any platform.</p>	<p><i>For a full description of the workshop, please see Page 21</i></p>	<p>ONE-DAY WORKSHOP</p> <p>WORKSHOP REF: MC01</p> <p>TWO-DAY WORKSHOP</p> <p>WORKSHOP REF: MC03</p>

DAY 2 AGENDA CONTINUED

14.00	ED COURTENAY	HOWARD DEINER	SASHA GOLDSHTEIN	MICHAEL HABERMAN	ANTHONY SNEED
	<p>BEHAVING LIKE A GIT, AND GETTING AWAY WITH IT Although Git has rapidly become almost a <i>de facto</i> standard in recent years, it can be intimidating or confusing for those transitioning from other systems or those new to using source control. In this session, Ed will explain how to use Git effectively, how to navigate your way around a repository, and how to work as part of a team. He'll attempt to cut through the mystique and demonstrate how easy it can actually be to use. Then he'll go on to show some of the more advanced ways of working with Git.</p>	<p>CONTINUOUS DELIVERY: THE WHYS, WHATS, AND HOWS DevOps is commonly believed to be accomplished by having the development staff collaborate more closely with the operations staff. That's definitely necessary, but woefully inadequate to achieve the goal of faster and better delivery in the "last mile" of an Agile shop. In this session, Howard will discuss the rationale behind Continuous Delivery, along with specific practices to get you started on making your sprints toward customer satisfaction less tiring and more enjoyable for everyone involved.</p>	<p>MAKING .NET APPLICATIONS FASTER Speed is king on mobile devices, embedded systems, and even run-of-the-mill desktop applications that need to start up quickly and deliver good performance on low-power machines. In this session, Sasha will review a collection of practical tips you can use today to make your .NET applications faster. He'll talk about choosing the right collection, improving start-up times, reducing memory pressure, and many other techniques for quickly improving your app's performance.</p>	<p>UNIT TESTING AND E2E TESTING USING JS-BASED FRAMEWORKS Unit testing and end-to-end (e2e) testing are the tools to enforce stability on applications. They create an environment that ensures our code does what it was designed to do. Recently, web application developers are looking to identify the best testing option, as their applications are getting increasingly large and more complex. In this session, Michael will review two methods for testing web applications in different JS-based frameworks: the unit-testing approach and end-to-end testing. He will also review the benefit of combining the two.</p>	<p>SECURING WEB APIs THE NEW WAY WITH OWIN AND KATANA Sometimes the technology landscape is changing so fast, it feels like you're standing on quicksand. That is certainly the case with ASP.NET Web API, the new OWIN hosting model and Microsoft's Katana implementation. In this session, Anthony will show how to correctly apply security at the transport level to ensure confidentiality, integrity and server authentication, as well as the nuts and bolts of configuring SSL for both web and self-hosted web APIs using the new OWIN hosting model.</p>

16.00	KEVLIN HENNEY	JULES MAY	IDO FLATOW	SHAY FRIEDMAN	SASHA GOLDSHTEIN
	<p>PROGRAMMING WITH GUTS These days, testing is considered a sexy topic for programmers. Who'd have thought it? But what makes for good unit tests (GUTs)? There's more to effective unit testing than just knowing the assertion syntax of a testing framework. Testing represents a form of communication and, as such, it offers multiple levels and forms of feedback, not just basic defect detection. Effective unit testing requires an understanding of what forms of feedback and communication are offered by tests. In this session, Kevlin will explore exactly what makes a good unit test.</p>	<p>TEAM BUILDING No programmer is an island. Modern programs are created by teams of developers. And everybody knows you need great teams to build great products – so you need to build your teams carefully. But what, exactly, makes a great programming team? Great programming skills? Great interpersonal skills? Working-all-night-because-the-boss-has-thrown-a-fit skills? Turns out, it's none of these. In this session, Jules will reveal that what makes a programming team great is exactly the same stuff that makes any other team great – and most programming teams don't have it.</p>	<p>ASP.NET VNEXT: REIMAGINING WEB APPLICATION DEVELOPMENT IN .NET ASP.NET vNext is being designed from the bottom up to be a lean and composable .NET stack for building web and cloud-based applications. Envision an ASP.NET stack where MVC, Web API, and web pages are all merged into the same framework, where you have a server-optimised version of ASP.NET with a smaller memory footprint. This is the new ASP.NET vNext. In this session, Ido will explore the ecosystem of ASP.NET vNext, its new project system and configuration system, and how to use it to build exciting web applications.</p>	<p>CHROME DEVELOPER TOOLS – A DEEP DIVE Every developer needs a set of tools, especially web developers that bend under the pressure of multiple languages, environments, IDEs and what not. One of the most comprehensive toolsets out there today is Chrome Developer Tools. It contains so many amazing features beyond the common ones, and it's just a shame most developers don't know about them! In this session, Shay will tell you all about the known and less-known features of Chrome Developer Tools, and you'll see how your everyday web development can become easier with just a few simple steps.</p>	<p>PRACTICAL C# 6 AND BEYOND Visual Studio 2015, .NET 2015 and C# 6 are just around the corner. The new language features have been out of the bag for a while now, but how do you apply them effectively? How do you refactor existing code to be shorter and sweeter? In this fast-paced session, Sasha will lead us through experiments with the new language features, including expression-bodied members, enhancements to automatic properties, null propagation, string interpolation and many others.</p>

ALSO EACH DAY...



8.30 COFFEE & REGISTRATION



11.00 COFFEE BREAK



13.00 LUNCH BREAK



15.30 COFFEE BREAK

<p>ALLEN HOLUB</p> <p>ZEROMQ AND RABBITMQ: MESSAGING FOR AGILITY AND SCALABILITY Messaging is an essential technology in high-volume, dynamically scalable server applications. It's the most effective way to pass non-time-critical information between servers, and to distribute work within a server farm. At the inter-server level, messaging is ideal for use with remote databases, monitoring, logging and so on, and a far better solution to intra-server data sharing than a shared database. Allen looks at messaging from an architectural perspective, with practical examples using RabbitMQ and ZeroMQ.</p>	<p>GARY SHORT</p> <p>HADOOP KICKSTARTER FOR MICROSOFT DEVS Big data is the new shiny thing right now, and if you read the blogosphere you'd be forgiven for thinking it was a tool just for Linux devs – or worse, only for those annoying hipsters with their shiny Macs. Nothing could be further from the truth. Windows makes an excellent platform for Hadoop and, in this session, Gary will show you everything you need to know to get started. From downloading and installing, to writing your first map-reduce job, using both the streaming API and the SDK. This session will cover it all, so come along and join the big data wave!</p>	<p>SANDER HOOGENDOORN</p> <p>INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND FOOLS The first statement in the Agile Manifesto favours individuals, teams, interaction and collaboration over processes and tools. But there are two sides to every story. When it comes to tools, the Agile Manifesto is often misinterpreted, in the sense that it's wrong to use tooling in Agile projects. Despite this, more and more vendors are trying to jump on the Agile bandwagon and sell their tools as being the most Agile toolset available. In this session, Sander shines a critical light on the sense and nonsense of tools in the Agile field.</p>	<p>MARK SMITH</p> <p>GETTING YOUR MOBILE APPS READY FOR THE WORLD Localising your applications can open up a whole new audience of users for your software. In this session, Mark will take a look at how to get your application ready for localisation and how to then utilise the built-in services of iOS, Android and Windows Phone to display proper information for different cultures and regions.</p>	<p>SAHIL MALIK</p> <p>LEARN ANGULARJS: THE ROAD TO POWERFUL, MAINTAINABLE APPLICATIONS One-day workshop continues from the morning session. <i>For a full description of the workshop, please see Page 21</i></p>	<p>ANDY CLYMER & RICHARD BLEWETT</p> <p>SOLID ASYNC PROGRAMMING IN .NET Two-day workshop continues from the morning session. <i>For a full description of the workshop, please see Page 20</i></p>
<p>KLAUS ASCHENBRENNER</p> <p>UNIQUEIDENTIFIERS AS PRIMARY KEYS IN SQL SERVER Is it good practice to use uniqueidentifiers as primary keys in SQL Server? They have a lot of pros for devs, but DBAs just cry when they see them enforced by default as unique clustered indexes. In this session, Klaus will cover the basics of uniqueidentifiers: why they are sometimes bad and sometimes good; and how to discover if they affect the performance of your performance-critical database. If they are having a negative impact, you will also learn some best practices you can use to resolve those limitations without changing your underlying application.</p>	<p>DAN CLARK</p> <p>AUTOMATING SSIS PACKAGE CREATION WITH BIML (BIML) is a powerful XML-based markup language that allows you to generate SSIS packages programmatically. Using BIML along with C#, you can create metadata-driven packages, greatly reducing development time and increasing consistency across the team. In this session, Dan will show you how to automate your SSIS package creation using the power of BIML and C#. You'll see how to create a template for loading dimension tables that will greatly increase your productivity.</p>	<p>SEB ROSE</p> <p>CUCUMBER AND SPECFLOW AS PART OF YOUR DEVELOPMENT PROCESS Behaviour-driven development (BDD) and specification by example (SBE) are quite recent additions to the software development toolbox. Sometimes it feels like we're using a hammer to drive in a screw. So, in this session, Seb will explore what they're good for and when to use them. He'll also look at what problems they don't help with and when not to use them. By the end of this session, you'll know enough to decide whether your problems are more like a screw or a nail – and whether Cucumber/SpecFlow is the right hammer.</p>	<p>NUNO GODINHO</p> <p>IOT & M2M: HOW ARE THEY CHANGING THE WORLD WE LIVE IN? Internet of Things (IoT) is here, and every day a new sensor or device starts to generate more data. With that, more and more machine-to-machine (M2M) communications start to happen, which make our solutions behave differently and face new issues. In this session, Nuno will look at how both of these new "buzz words" are changing the world we live in, from fitbit to Google Glass and smart watches. How can we prepare for this? How can we anticipate and get some business opportunities from it? Join us and find out.</p>	<p>ONE-DAY WORKSHOP WORKSHOP REF: MC01</p>	<p>TWO-DAY WORKSHOP WORKSHOP REF: MC03</p>

DevWeek Day 3 sees the concluding part of our two-day workshop on asynchronous programming in .NET, plus a new workshop on Entity framework, plus 36 more breakout sessions covering a wide range of topics.

DAY 3 AGENDA

<p>09.30</p>	<p>MICHAEL KENNEDY</p> <p>PYTHON: AN AMAZING SECOND LANGUAGE FOR .NET DEVELOPERS The modern software development landscape is a terrain of many platforms and technologies. Gone are the days where knowing one technology really well was enough to stay on the cutting edge. Even as we know we should learn more and branch out, that choice is increasingly difficult as the technology options explode. In this session, Michael offers one very solid choice: Python. It may seem like a very different language and ecosystem from .NET but beneath the surface, there are many more similarities than differences.</p>	<p>DINO ESPOSITO</p> <p>THE (DRAMATIC?) IMPACT OF UX ON SOFTWARE ARCHITECTURE AND DESIGN Always neglected in favour of domain analysis and modelling, the presentation layer of applications receives little attention. But whether your application is web, mobile or desktop, the presentation layer is the face it shows to users. Dino will discuss a design approach that starts from requirements and builds the system from top to bottom, focusing on use-cases, screens and overall user experience measured by a new professional figure – the UX architect – and backed by new, but partially green, tools such as UXPin and Balsamiq.</p>	<p>CHRISTOS MATSKAS</p> <p>MEET THE NEW KID ON THE BLOCK: MICROSOFT ASP.NET 5 Imagine if you could write an ASP.NET application using your favourite text editor, compile it and run it on Mac OS X. Imagine if you could mix and match Web Forms, MVC, Web API and SignalR within a single project. How would it feel to create a faster, leaner and more memory-efficient ASP.NET application that has been freed from the shackles of Windows, and all you need are your coding skills, a couple of NuGet packages and your imagination? In this session, Christos provides an intro to Microsoft's ASP.NET 5 – the “new kid on the block”.</p>	<p>DROR HELPER</p> <p>UNIT TESTING PATTERNS FOR CONCURRENT CODE Getting started with unit testing is not hard, the only problem is that most programs are more than a simple calculator with two parameters and a return value that's easy to verify. Writing unit tests for multi-threaded code is harder still. In this session, Dror will demonstrate useful patterns that he has discovered over the years, and that have helped him to test multi-threaded and asynchronous code and enabled the creation of deterministic, simple and robust unit tests. He'll also point out the pitfalls to avoid.</p>	<p>EON WOODS</p> <p>SYSTEM SECURITY BEYOND THE LIBRARIES Security is now important to all of us, not just people who work at Facebook. But it's a complicated domain, with a lot of concepts to understand. In any technical ecosystem, there is a blizzard of security technology, as well as generic concepts such as keys, roles, certificates, trust, signing and so on. Yet none of this is useful unless we know what problem we're really trying to solve. In this session, Eoin will dive into the fundamentals of system security to introduce the topics we need to understand in order to decide how to secure our systems.</p>
<p>11.30</p>	<p>PAVEL SKRIBTSOV</p> <p>DEEP LEARNING: THE HANDCRAFTED CODE KILLER In this session, Pavel will reveal how a new direction in artificial intelligence, called “deep learning”, is gradually reducing demand for hand-crafted code for intellectual data analysis, primarily in the area of feature extraction. He will explain why the internet giants (Google, Microsoft etc) are interested in deep learning, and the connection with big data projects. He will also cover practical examples of applying existing deep-learning software frameworks to an image-recognition problem.</p>	<p>AMY CHENG</p> <p>DRAWING WITH JAVASCRIPT: ANIMATIONS AND INFOGRAPHICS A number of libraries and frameworks allow developers to use JavaScript to create engaging visuals without switching programming languages. So let's explore the visual (and fun) side of JavaScript! In this session, Amy will provide a whirlwind tour of a few libraries and frameworks that let you create animations, simple drawings and infographics with JavaScript. First, she'll examine the advantages (and disadvantages) of using JavaScript for graphics and animations. Then she'll go through “Hello World” examples of a few JavaScript-based libraries.</p>	<p>SANDER HOOGENDOORN</p> <p>INTRODUCING AND EXTENDING BOOTSTRAP Bootstrap is by far the most popular web framework of all, with many ready-to-use styles and components in CSS and JavaScript. In this session, Sander will show you how to build a basic web site, leveraging the many components of the Bootstrap framework. He will then go on to show the use of additional frameworks and libraries to add drop-down support, icons and date pickers to your web pages, and how to build additional reusable components using Razor syntax, in JSF, and applying Angular directives. Of course, Sander's talk will be illustrated with many coding demos.</p>	<p>ALLEN HOLUB</p> <p>DBC (DESIGN BY CODING) Design by Coding (DbC) is a way to develop an architecture incrementally as you code. It builds on test- and behaviour-driven-development techniques, but adds a focus on the “story” that's central to all Agile processes. The process answers the question of how you can build a coherent Agile system incrementally, without a formal up-front design process. In this session, Allen will explain how DbC eliminates the need for a separate design phase in the development process, since your code is effectively your design artefact.</p>	<p>TONI PETRINA</p> <p>AWESOME THINGS YOU CAN DO WITH ROSLYN Roslyn, the revamped compiler for C# and Visual Basic.NET, goes beyond a mere black-box compiler and gives us limitless possibilities. Besides enabling a new era for C# as a language, it gives everyone a chance to utilise compiler powers for building custom tools. It acts as a CaaS, or Compiler as a Service, which allows you to plug in at any point in the compilation process. But what can you do with it? In this session, Toni will show you, demonstrating how you can build Visual Studio extensions, create your own editors or host C# compiler to form a scripting environment.</p>



<p>KLAUS ASCHENBRENNER</p> <p>HEADACHE GUARANTEED: DEADLOCKING IN SQL SERVER! SQL Server needs its locking mechanism to provide the isolation aspect of transactions. As a side-effect, your workload can run into deadlock situations – a guaranteed headache for any DBA! In this session, Klaus will look into the basics of locking and blocking in SQL Server. Based on that knowledge, you will learn about the various kinds of deadlocks that can occur in SQL Server, how to troubleshooting them, and how you can resolve them by changing your queries, your indexing strategy and your database settings.</p>	<p>DEJAN SARKA</p> <p>DATA EXTRACTION AND TRANSFORMATION WITH POWER QUERY AND M Power Query, a free add-in for Excel 2010 and 2013 and part of the Power BI suite in Excel 365, is a powerful tool. Dejan will show how you can use Power Query to gather all kinds of data, from databases to web sites and social media, inside Excel data models. In this way, you can make Excel an analysing engine for structured and unstructured data. In addition to the queries you can create through the UI, there is a fully functional language, called M, behind the scenes. This session introduces both Power Query and M.</p>	<p>ALLEN HOLUB</p> <p>MICRO SERVICES: A CASE STUDY In this session, Allen will take a deep dive into a micro-service implementation. He'll look at both the architecture and the implementation of authentication and comment-management micro-services suitable for use in a blog or similar application. The core system is written in Java, so you'll need to know Java, C++, C#, or equivalent to follow along easily. Auxiliary technologies include Mongo, JavaScript, AngularJS and Bootstrap, so this session provides a real-world example of how those technologies work. You don't need to be familiar with any of them, though.</p>	<p>SASHA GOLDSHTEIN</p> <p>MODERNISING C++ CODE The C++ standard library dates back to the 1990s, but that doesn't mean there aren't new and exciting frameworks to use in your C++ application. In this session, Sasha will look at some brand-new and some existing C++ libraries that can speed up C++ cross-platform development rapidly. Some of the libraries we might cover include: Casablanca (C++ REST SDK), Cinder (creative coding), Boost (general-purpose), Google Test (unit testing), SOCI (modern database access) and many others. There's something for everyone!</p>	<p>ANTHONY SNEED</p> <p>SOUP TO NUTS: DEVELOPING REAL-WORLD BUSINESS APPS WITH ENTITY FRAMEWORK AND ASP.NET WEB API Performance. Scalability. Maintainability. Testability. Security. Today's application developers need to build systems that are designed to achieve these goals from the outset. In this in-depth workshop, Anthony will take you beyond the basics, to learn how to build RESTful services that are robust, scalable and loosely coupled, using dependency injection with repository and unit of work design patterns. But there's more to building loosely coupled systems than applying a set of design patterns. Anthony will show you how to harness the power of code generation by customising T4 templates for reverse engineering Code First classes from an existing database, in order to produce entities with persistence concerns that are completely stripped away. You'll also learn ninja techniques for handling cyclical references with code-based configuration and using efficient binary formatters, all without polluting your entities with mapping, serialisation or validation attributes. This workshop will focus on developing real-world business apps using the Entity framework and ASP.NET Web API.</p> <p>ONE-DAY WORKSHOP WORKSHOP REF: MC02</p>	<p>ANDY CLYMER & RICHARD BLEWETT</p> <p>SOLID ASYNC PROGRAMMING IN .NET Two-day workshop continues from the previous day's session.</p> <p><i>For a full description of the workshop, please see Page 20</i></p> <p>TWO-DAY WORKSHOP WORKSHOP REF: MC03</p>
<p>SASHA GOLDSHTEIN</p> <p>RAVENDB: THE .NET NOSQL DATABASE The big data hype is all about NoSQL databases that can support huge amounts of data, replication, scaling and super-fast queries. RavenDB is the best NoSQL database for .NET developers, because it has a first-class .NET client with a LINQ API. In this session, Sasha will show you how to model data as documents for storage in RavenDB; how to query the data efficiently; and how to construct indexes that will help you get the data you need in just a few milliseconds. We'll also review full-text search and query suggestions support, which make it very easy to add search capabilities.</p>	<p>DAN CLARK</p> <p>CREATING SOLID POWER PIVOT DATA MODELS Self-service business intelligence is gaining popularity among business analysts today. It greatly relieves the problems created by traditional data warehouse implementations. Using tools such as Microsoft's Power Pivot, Power Query and Power View alters the process significantly. In this session, Dan will take you through the process of creating a solid data model in Power Pivot. You will learn how to import data from various sources, combine these in a scalable data model, use DAX to create measures, and incorporate time-based analysis.</p>	<p>HOWARD DEINER</p> <p>GETTING PAST THE 50 + 70 = 120 CALCULATOR IN CUCUMBER: 12 THINGS TO WORK ON With the best intentions, people have flocked to behaviour-driven development by way of Cucumber over the past few years, and that's a great thing! But often, BDD can fall by the wayside due to the pressure to deliver more and more functionality, sprint after sprint. In this session, Howard will explore 12 of the most important issues, such as imperative versus declarative style, and how to keep Gherkin-driven Selenium WebDriver tests working dependably through the use of advanced Expected-Condition techniques.</p>	<p>NUNO GODHINO</p> <p>ARCHITECTURE BEST PRACTICES ON WINDOWS AZURE When new technologies and paradigms appear, it's essential to learn them quickly and well. But this can be difficult, since some things are only learned with experience. That's why best practices are so important. In this session, Nuno will look at some architecture best practices that will help us make our solutions better across several levels, including performance, cost, integration, security and so on. By doing this, you'll gain the knowledge needed to quickly start using the technology and paradigms that can help improve your business.</p>		





DAY 3 AGENDA CONTINUED

14.00	PETER O'HANLON	ANDREY ADAMOVIICH	SHAI REZNIK	IDO FLATOW	MICHAEL HABERMAN
	<p>BUILDING NATURAL USER INTERFACES WITH REALSENSE DEVICES</p> <p>The Intel RealSense SDK is the powerhouse behind the perceptual computing cameras powering the next generation of Ultrabook devices. In this session, Peter will take a look at how RealSense devices can provide unique ways to interact with applications, explaining the advantages, and offering tips and tricks to building compelling applications using RealSense devices. You will learn how to easily create applications that use gesture and facial recognition, emotion detection, as well as speech recognition and speech synthesis.</p>	<p>GROOVY DEVOPS IN THE CLOUD</p> <p>In this session, Andrey will focus on a set of tools to automate the provisioning of (cloud) servers using Groovy libraries and Gradle plug-ins. He will explore how to leverage those to create an infrastructure for building, configuring and testing the provisioning of boxes in the cloud – elegant and groovy. This session will help those Java/Groovy developers interested in reusing their existing skills for infrastructure provisioning and learning more about problems encountered during system operations.</p>	<p>18 TIPS FOR THE ANGULAR ARCHITECT</p> <p>So your company is planning to build a large-scale web application, and has chosen to do it in Angular.js. That raises a lot of questions: Where do I start from? What tools should I use? And, basically, how do I avoid making mistakes and do the job efficiently? Shai has worked with more than 20 companies, helping them with their struggles migrating to Angular and avoiding crucial mistakes in the process. In this talk, Shai will present useful time-saving architecture tips that will help you prepare for scalability, write cleaner code, and even make your life happier.</p>	<p>DEBUGGING THE WEB WITH FIDDLER</p> <p>Every web developer needs to see what goes on “in the wire”, whether it is a jQuery call from JavaScript, a WCF service call from a client app, or a simple GET request for a web page. With Fiddler, the most famous HTTP sniffer, this is simple enough to do. But Fiddler is more than just a sniffer. With Fiddler you can intercept, alter and record messages, and even write your own message visualiser. In this session, we will learn how to use Fiddler from bottom to top to debug, test and improve web applications.</p>	<p>FROM XAML/C# TO HTML/JS</p> <p>In recent years, the mobile evolution caused many developers to find themselves migrating from desktop applications to web applications. In this session, Michael will explore how to make the transition from XAML and C# to HTML5 and JavaScript. He will review how to port the MVVM design pattern to the web environment, and go on to tackle important architecture concepts, such as dependency injection and modularity.</p>

16.00	KEVLIN HENNEY	JOE NATOLI	DINO ESPOSITO	GIL FINK	NEAL FORD
	<p>GIVING CODE A GOOD NAME</p> <p>Code is basically made up of three things: names, spacing and punctuation. With these three tools, a programmer needs to communicate intent, and not simply instruct. But if we look at most approaches to naming, they are based on the idea that names are merely labels, so that discussion of identifier naming becomes little more than a discussion of good labelling. A good name is more than a label; a good name should change the way the reader thinks. Good naming is part of good design. In this session, Kevlin will look at why and what it takes to get a good name.</p>	<p>THE BIG LIE: WHY FORM DOESN'T (AND SHOULDN'T) FOLLOW FUNCTION</p> <p>The prescriptive interpretation of this axiom has guided the work of engineers, programmers, developers – and even designers – for a very long time. The result of this has been sites, software and systems that exhibit poor usability, frustrating user experiences and a marked failure to deliver expected business results. In this session, Joe will show you why pure function is rarely the single or most important component of success. He will explain how every force at play in any project is what really evolves form (and dictates function).</p>	<p>MAKING AN EXISTING WEB SITE A MOBILE-FRIENDLY WEB SITE</p> <p>Do you feel frustrated every time you run across a web site that doesn't adjust to the viewport of your current phone browser? In some cases, for a better experience, you have to know that a mobile version of the site exists somewhere with a different URL. There's no reason to further delay plans to make your primary web site display nicely on small-screen devices, including smartphones. In this session, Dino will lead the discussion and explore pros, cons and technologies that could make each option viable.</p>	<p>THE CHARACTERISTICS OF A SUCCESSFUL SPA</p> <p>Single-page applications (SPAs) are web applications that are built using a single page, which acts as a shell to all the other web pages, with a rich JavaScript front-end. As opposed to traditional web applications, most of the SPA development is done on the front-end. The server, which once acted as a rendering engine, provides only a service layer to the SPA. In this session, Gil will explain the characteristics and building blocks that form the foundation of any successful SPA.</p>	<p>BUILD YOUR OWN TECHNOLOGY RADAR</p> <p>ThoughtWorks' Technical Advisory Board creates a “technology radar” twice a year: a working document that helps the company make decisions about what technologies are interesting. This is a useful exercise both for you and your company. In this session, Neal will describe the radar visualisation, how to create litmus tests for technologies, and the process of building a radar. Attendees will leave with tools that enhance your filtering mechanisms for new technology and help you (and your organisation) develop a cogent strategy to make good choices.</p>



<p>SASHA GOLDSHTEIN</p> <p>DAWN OF A NEW ERA: AN OPEN-SOURCE .NET</p> <p>The .NET framework is now open source, with the CLR to follow along and a cross-platform reference implementation for Linux and OS X to show up during the year. In this session, Sasha will talk about the future of .NET in this new era, what it means for the core stack on the server and desktop, and how it's going to affect our .NET applications. We will also see how to build, test and run our own version of the .NET framework and CLR, and how to make changes to components previously treated as a black box.</p>	<p>DEJAN SARKA</p> <p>VISUALISING GRAPHICAL AND TEMPORAL DATA WITH POWER MAP</p> <p>Power Map is a new 3D visualisation add-in for Excel, used for mapping, exploring and interacting with geographical and temporal data. Power Map exists as free preview add-in for Excel 2013 and in the Power BI suite in Office 365. In this session, Dejan will explain how to use Power Map to plot geographic and temporal data visually, analyse that data in 3D, and create cinematic tours to share with others.</p>	<p>JULES MAY</p> <p>PROBLEM SPACE ANALYSIS</p> <p>How do you design a large system? The architecture of any system is crucial to its success – get this wrong, and the project may never recover. And yet, we are expected to deliver designs that can last five, 10, sometimes 30 years into an unknowable future. Problem space analysis is a technique that informs and documents system designs by anticipating and defining the variabilities of a long-lived, evolving system. In this session, Jules will explain the principles of the method, give an outline of the benefits, and demonstrate its power with some illustrative examples.</p>	<p>TUSHAR SHARMER</p> <p>DOES YOUR DESIGN SMELL?</p> <p>In this session, Tushar will propose a unique approach to developing high-quality software design. Borrowing a phrase from the healthcare domain, “a good doctor is one who knows the medicines but a great doctor is one who knows the disease”, the proposed approach is grounded on the philosophy that “a good designer is one who knows about the design principles but a great designer is one who understands the problems (or smells) with the design, their cause, and how they can be addressed by applying proven and sound design principles”.</p>	<p>ANTHONY SNEED</p> <p>SOUP TO NUTS: DEVELOPING REAL-WORLD BUSINESS APPS WITH ENTITY FRAMEWORK AND ASP.NET WEB API</p> <p>One-day workshop continues from the morning session.</p> <p><i>For a full description of the workshop, please see Page 21</i></p>	<p>ANDY CLYMER & RICHARD BLEWETT</p> <p>SOLID ASYNC PROGRAMMING IN .NET</p> <p>Two-day workshop continues from the morning session.</p> <p><i>For a full description of the workshop, please see Page 20</i></p>
<p>KLAUS ASCHENBRENNER</p> <p>JOINS IN SQL SERVER - AS EASY AS ABC?</p> <p>Have you ever looked at an execution plan that performs a join between two tables? And have you ever wondered what a “Left Anti Semi Join” actually is? Joining two tables using SQL Server is far from easy! In this session, Klaus will take a deep dive into how join processing happens in SQL Server. Initially, he will lay out the foundation of logical join processing, then dig deeper into physical join processing in the execution plan. After attending this session, you will be well prepared to understand the various join techniques used by SQL Server.</p>	<p>GARY SHORT</p> <p>TROLL HUNTING ON THE INTERNET</p> <p>With so many people on social media these days, almost inevitably not a day goes by without some tragedy befalling someone. As if that wasn't horrible enough, these poor souls and their families can then become victims of the perverse behaviour of the “trolls. In this session, Gary will examine this problem from a data scientist's point of view, showing how to use computational linguistics to ensure that such posts never reach people's streams, and network theory to trace and expose the trolls so that they no longer have the shield of anonymity to hide behind.</p>	<p>SEB ROSE</p> <p>MONAD AT THE HYPERBOLE (AND OTHER AWESOME STORIES)</p> <p>In this session, Seb will help you to be a better software developer. As software developers, we have to deliver something useful to our customers. We have to produce it in a manner that acknowledges their requirements and context. And usually, we need to be able to work as part of a team. Are my customers more likely to be satisfied if I'm awesome? In what circumstances would monads (or any other implementation level detail) be a critical part of a successful solution? Seb will analyse real-world examples of projects that succeeded and failed.</p>	<p>IDO FLATOW</p> <p>MIGRATING APPLICATIONS TO MICROSOFT AZURE: LESSONS LEARNED FROM THE FIELD</p> <p>How much time will it take us to move to Azure? Can we just “Lift & Shift” our servers? Will my load-balancer work in Azure? Should I use SQL Databases or an SQL Server VM? These are just some of the questions customers ask when they consider migrating their applications to Azure. If you're evaluating Azure, come to this session, where Ido will explain what to do, what not to do, what to avoid and what to embrace when moving your apps to Azure. These are not general best practices; these are lessons learned from the field.</p>	<p>ONE-DAY WORKSHOP</p> <p>WORKSHOP REF: MC02</p>	<p>TWO-DAY WORKSHOP</p> <p>WORKSHOP REF: MC03</p>



DevWeek concludes with our final day of post-conference workshops.

As on previous days, all workshops run for a full day, from 09.30 to 17.30, with short breaks in the morning and afternoon, and a lunch break at 13.00.

POST-CONFERENCE WORKSHOPS

09.30



NEAL FORD

CONTINUOUS DELIVERY

Getting software released to users can be painful, risky and time-consuming. Here, Neal sets out the principles and practices that enable rapid, incremental delivery of high-quality, valuable new functionality to users. By automating the build, deployment and testing process, and improving collaboration between developers, testers and operations, delivery teams release changes in a matter of hours or even minutes.

Neal will look at the differences between related topics such as continuous integration, continuous deployment and continuous delivery, and explore the new technical artefact that continuous delivery introduces: the deployment pipeline. He'll discuss the various stages, how triggering works, and how to pragmatically determine what "production ready" means, before covering the role of testing and the testing quadrant, including the audience and engineering practices around different types of tests. This is followed by version control usage and offering alternatives to feature branching, such as toggle and branch by abstraction. Neal will then go on to cover operation, DevOps and programmatic control of infrastructure, using tools such as Puppet and Chef.

WORKSHOP REF: DW11



JOE NATOLI

GENERATING MEANINGFUL REQUIREMENTS

Ask any group of people what they want or need and you'll find no shortage of opinions or answers. Clients and stakeholders will always have a voluminous laundry list of features and functions, all of which they will insist are equally important. Your clients, employers, project stakeholders and users all share something very important in common: they're all human beings. And we human beings all have a fundamental flaw: we often make very confident – but equally false – predictions about our future behaviour.

So the requirements that will actually be most useful and most valuable – the ones that will increase user adoption or sales; the ones that will make or save money – are almost never surfaced in traditional requirements sessions.

In this workshop, Joe will show you how to change that, along with how to tell the difference between what people say they need and what they *actually* need. Finally, he'll show you how to uncover the things they don't know they need (but absolutely do).

WORKSHOP REF: DW12



JULES MAY

PROBLEM SPACE ANALYSIS

How do you design a large system? We know Waterfall doesn't work very well; yet we also know that Agile scales poorly. Various proposals have been made (BDUF, domain-driven design, prototyping) but none really solves the problem.

The key to managing a large system is managing *change*. No specification ever survives its own implementation: as a system takes shape, everyone – developers, architects, stakeholders – change their minds. In any non-trivial project, goalposts are constantly in motion. A robust architecture is one that anticipates those changes, and a good design is one that accommodates them cheaply and efficiently.

Problem space analysis is a technique that simply and clearly anticipates, documents and defines the changes that can affect a project. It informs the architectural design so that it can accommodate those changes, and it delivers a change-tolerant ubiquitous language to unify and coordinate the development effort.

Jules introduces the principles of problem space analysis and how they translate into architectures and working systems, even while the goalposts are moving. The technique will be actualised using a real-life design problem.

WORKSHOP REF: DW13



TUSHA SHARMA

ACHIEVING DESIGN AGILITY BY REFACTORING DESIGN SMELLS

Software design plays an important role in adapting changing requirements for software development. Software products that follow Agile methodologies are no exception. Agile methodologies welcome rapid requirement changes and yet promote timely delivery. Often, software teams follow these practices (to some extent) but ignore the software design and its quality due to negligence or ignorance. "Design agility" suggests that the software design has to be Agile if the team intends to follow Agile practices in a true sense. In the absence of agility of design, it's difficult to achieve the benefits of being Agile.

Tushar emphasises the importance of design agility by exposing design smells in software systems. He explains design smells and connects them to their impact on software design, design agility, bug-proneness and delivery schedule of the software.

This workshop offers an understanding of the importance of software design quality and design agility. You'll also learn the vocabulary of smells, using a comprehensive classification, and practical refactoring strategies to repay technical debt.

WORKSHOP REF: DW14

“
Great lectures,
very informative.
Fantastic venue

JUNIOR WEB DEVELOPER
2014 DELEGATE

**IDO
FLATOW****THE ESSENTIAL
TOOLBOX FOR
TROUBLESHOOTING
ASP.NET WEB
APPLICATIONS**

Is your web application working slower than anticipated? Have you rewritten your application code, but still wonder if there is some ASP.NET or IIS trick you can use to boost things up?

Testing and profiling a web application is trickier than with desktop or mobile apps. It requires testing both front-end and back-end, and the network in-between the two. And since web applications are a mix of HTML, CSS, JavaScript and .NET code, you often need to use several tools to accomplish this task.

In this workshop, Ido will go over the process of testing and profiling web applications, and demonstrate how to use various tools of the trade – both for front and back ends. Ido will then go into detail on how to speed up your web applications, by providing important tips and tricks you can apply to the different parts of the application: ASP.NET (Web Forms, MVC and Web API), general .NET best practices, IIS server, networking tips, HTML/CSS and JavaScript.

**WORKSHOP
REF: DW15****SASHA
GOLDSHTEIN****MAKING THE MOST
OF C++11/14**

In this workshop, Sasha will look at the most important C++ language features that improve system performance and developer productivity, and see how to apply them to existing code.

The C++11 standard is already behind us, and C++14 is just around the corner. With a huge variety of language features such as lambdas, rvalue references, auto and decltype, and variadic templates, it's easy to get lost in C++. In fact, it often seems like a completely new and foreign language. We will make the most of Visual C++ 2013 and give a special focus to converting and refactoring code to use modern C++ idioms. Specifically, we will look at how to best use STL algorithms with lambda functions, when to use each kind of smart pointer class, how to convert macros and non-generic code to templates, and a variety of best practices concerning concurrency in C++ applications.

This will be a particularly relevant workshop for C++ developers who watched the C++ 11/14 train passing by and weren't able to apply all the best practices of modern C++ to their applications just yet.

**WORKSHOP
REF: DW16****DEJAN
SARKA****BI WITH
MICROSOFT TOOLS:
FROM ENTERPRISE
TO A PERSONAL
LEVEL**

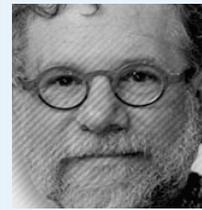
Microsoft has really been investing a lot in business intelligence (BI) over the last 10 years. The result is a huge number of analytical tools and services. When building a BI solution, many companies make basic mistakes and choose inappropriate tools for the problem they are trying to solve.

In this workshop, Dejan will help you put the building blocks into the right context. You will learn about data warehousing with SQL Server, reporting with SQL Server Reporting Services (SSRS), Power View and Power Map, on-line analytical processing (OLAP) with SQL Server Analysis Services (SSAS), Multidimensional, Tabular, Power Pivot for SharePoint Server, data mining with SSAS, Excel, R and Azure Machine Learning (ML), and about the extract-transform-load process with SQL Server Integration Services (SSIS) and Power Query.

**WORKSHOP
REF: DW17****GIL
FINK****BUILDING
SCALABLE
JAVASCRIPT APPS**

Building and maintaining large and scalable JavaScript web apps isn't easy. So how do you build your front-end-oriented applications without being driven to madness? Using and combining proven JavaScript patterns will do the trick.

In this workshop, Gil will discuss the patterns behind some of the largest JavaScript apps, such as Gmail and Twitter, and we'll explore how to apply them in your own apps. We'll start from object patterns and discuss how to write more object-oriented-like code in JavaScript. Then, we will focus on module patterns and asynchronous module definitions (AMD). We will also discuss patterns such as promises, timers and mediator. At the end of the day, we will combine the patterns and see how to use them to build your next scalable JavaScript web app.

**WORKSHOP
REF: DW18****ALLEN
HOLUB****THE SWIFT
PROGRAMMING
LANGUAGE**

Swift is Apple's new programming language. It represents a significant improvement over both Objective C and C++, incorporating many contemporary language features (such as duck typing and closures) without abandoning object orientation. It promises to become the dominant language of the Apple platform.

Swift is, unfortunately, a mixed bag. It supports many important language features, but it also omits features that are essential for long-term maintainability (privacy, for example).

In this workshop, geared to programmers who already know an OO language (C++, Java, C#, etc.), Allen will present all the interesting parts of the Swift language. We'll gloss over the basic stuff (declarations, flow control, etc.) and focus on those parts of the language that will be new to you (lambdas, subscripts, the inheritance model, extensions and chaining, etc.), with considerable emphasis on places where the language can get you into trouble.

**WORKSHOP
REF: DW19****SEB
ROSE****APPLIED BDD WITH
CUCUMBER/JAVA
AND SPECFLOW/C#**

It's all very well reading books, but nothing beats actually getting practical experience. In this workshop, working in your choice of Java, C# or Ruby, Seb will drive out the implementation of a simple utility by specifying its behaviour in Cucumber. The tyrannical Product Owner will regularly change his mind, so we'll need to keep our code well factored and easy to modify.

This session is designed for developers and testers. By the end of the day, you'll be comfortable working with Cucumber in your chosen development environment. You'll have seen, first hand, how to use Cucumber to drive out valuable features for your customers and how that can help keep your stakeholders engaged in the software development process. It will also be clear how BDD interacts with TDD.

Bring a laptop with your chosen development environment installed, and try to pre-install your chosen Cucumber variant before you come (instructions available). But don't worry, we can install on the day if necessary.

If this subject interests you, but you'd prefer a gentler intro, don't miss Seb's earlier workshop on Monday: "BDD by example".

**WORKSHOP
REF: DW20**



DevWeek's main conference days now offer more choice than ever before. Alongside the breakout sessions, you can choose from a special two-day workshop that runs across both Wednesday and Thursday, and one-day workshops to choose from on each day.

TWO-DAY WORKSHOP:

WED & THURS



ANDREW
CLYMER



RICHARD
BLEWETT

SOLID ASYNC PROGRAMMING IN .NET

In this special two-day workshop, Andrew and Richard will take you through the core skills required to be successful developing async and multithreaded code, both in the .NET and web worlds. Not only do we cover the core APIs but also how they are used effectively, tested and debugged.

Tasks

When the Parallel Framework Extensions (PFX) were first announced, it looked as though they were going to target a narrow set of requirements around parallelising processor intensive code. Over time, the scope of the library has grown significantly, such that it will become the main model for building asynchronous code. The pivotal type enabling this transition is the Task class. This is a functionally very rich type, allowing the creation of both short- and long-lived asynchronous work, Tasks can have dependencies on one another and support cancellation. In this, the first of the PFX modules we look specifically at how this class gives us a unified framework for building multithreaded code.

Thread safety

Asynchronous programming requires careful attention to detail, since most objects are not designed with multithreaded access in mind. This module introduces the importance of Interlocked and Monitor-based synchronisation.

Concurrent data structures

Ever since its inception, .NET has had support for a number of synchronisation primitives (such as Interlocked, Monitor and Mutex). However, on their own, these primitives do not provide support for more complex synchronisation situations, and so people have had to use them as building blocks to build things such as efficient semaphores. PFX finally brings to the library a set of richer primitives, such as lazy initialisation, a lightweight semaphore and a countdown event. But more than this, it also introduces a set of high-performance concurrent data structures that allow you to use them without you having to provide your own synchronisation logic around them. We also look at the new Immutable

Collections package, now available on NuGet, which provides another way to model data for async applications.

Parallel

The initial goal of PFX was to simplify the parallelisation of processor intensive tasks – and this remains a key feature. This part of its functionality is focused on the Parallel class and its For and ForEach members. In this module, we look at the simplified model but also highlight that parallelising algorithms is never as simple as it might first seem – we show you some of the pitfalls that you should be aware of when trying to parallelise functionality using the Parallel class.

async/await

C# 5 builds on the Task API, introducing async and await keywords, which bring asynchronous execution as a first-class concept in the C# language. These new keywords create a very elegant model for all sorts of async work and this module explains not only how to use them but also how they work under the covers.

Server-side async

The nature of server-side applications often means they are asynchronous by their very design, servicing many clients at the same time on different threads. But as you dig deeper, you often find these threads performing long-running blocking operations – particularly in terms of IO resulting in consuming more threads (an expensive resource) than necessary. This module focuses on a range of server-side technologies and demonstrates how to perform maximum concurrency for least number of threads.

TPL Dataflow

TPL Dataflow is a downloadable addition to the TPL (Task Parallel Library) that ships with the .NET framework. TPL Dataflow provides an alternative approach to define concurrency. Instead of just simply throwing threads at a synchronously structured program and having to deal with all the thread safe and race conditions that introduces; we have the concept of many autonomous objects, each with its own thread of execution. These autonomous objects co-operate with other such objects through asynchronous message passing. In this module, we will see how TPL Dataflow can greatly reduce the complexity normally associated with asynchronous programming.

Rx

Reactive Framework is a new library that uses the .NET 4.0 IObservable interface and LINQ to create a compelling new programming model that allows you to build “event”-based code with declarative LINQ statements. This module introduces the Reactive Framework and shows how it can greatly simplify your code.



A wonderful way
to rethink what
it means to be a
great developer

SOFTWARE DEVELOPER
2014 DELEGATE

ONE-DAY WORKSHOPS:

WEDNESDAY

SAHIL
MALIK

LEARN ANGULARJS: THE ROAD TO POWERFUL, MAINTAINABLE APPLICATIONS

JavaScript, by its nature, makes it difficult to write maintainable code. HTML, by its nature, is loosely structured. AngularJS fixes both of those. It is a structural framework for dynamic web apps, allowing you to extend HTML's

syntax, enabling you to write powerful, maintainable applications succinctly.

In this workshop, Sahil will build on your existing knowledge of JavaScript and teach you the ins and outs of AngularJS. There are plenty of examples, which will walk you through a basic introduction, models, controllers and views in Angular, templates and databinding, services and dependency injection, directives, routing and single-page applications.

An introduction to AngularJS

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. This module gets you started with AngularJS showing the basic syntax and some quick starts to get us running.

Models, controllers and views

Controllers in AngularJS are a fundamental building block of Angular. AngularJS encourages better code architecture by encouraging you to use MVC-based patterns. There is rich support for controllers and views, and this module will familiarise you with those.

Templates and databinding

When working with complex forms, or even reactive user interfaces, it really helps to leverage concepts such as databinding. Databinding in JavaScript is especially valuable but can be difficult to implement, unless you use something like Angular.

Services in Angular, dependency injection

Services in Angular are substitutable objects that are wired together using dependency injection. Fancy! What on Earth does that even mean? Well for one, services are a classic way of bundling together reusable code across controllers in your application, and they can be lazily instantiated or singletons. There are services that Angular provides, such as \$http, or ones you can create. And then there is something called interceptors. This module gets in in the deep of AngularJS services.

Directives

Directives in AngularJS are markers on DOM elements that tell the AngularJS HTML compiler to attach some specified behaviour to that particular DOM element or its children. There are many directives, such as ngBind, ngModel, ngClass, and this module shows you the most important directives.

Routing and single-page applications (SPAs)

AngularJS routes enable you to create different URLs for different content in your application. Having different URLs for different content enables the user to bookmark URLs to specific content. Single-page applications allow you to write HTML and JavaScript code that more or less performs and behaves like a thick client application. It's pretty neat – trust me, you'll be using this.



Very informative, covering a broad range of technology and a good opportunity to engage with industry experts

SOFTWARE ARCHITECT, 2014 DELEGATE

THURSDAY

ANTHONY
SNEED

SOUP TO NUTS: DEVELOPING REAL-WORLD BUSINESS APPS WITH ENTITY FRAMEWORK AND ASP.NET WEB API

Performance. Scalability. Maintainability. Testability. Security. Today's application developers need to build systems that are designed to achieve these goals from the outset. In this in-depth workshop, Anthony will take

you beyond the basics, to learn what it takes to build RESTful services that are robust, scalable and loosely coupled, using dependency injection with repository and unit of work design patterns.

But there's more to building loosely coupled systems than applying a set of design patterns. Anthony will show how you can harness the power of code generation by customising T4 templates for reverse engineering Code First classes from an existing database, in order to produce entities with persistence concerns that are completely stripped away. You'll also learn ninja techniques for handling cyclical references with code-based configuration and using efficient binary formatters, all without polluting your entities with mapping, serialisation or validation attributes.

This workshop will focus primarily on developing real-world business apps using the Entity framework and ASP.NET Web API.

DevWeek's speakers are acknowledged experts in their field. Recognised internationally, the 2015 speaker faculty comprises professional consultants, trainers, industry veterans, thought-leaders and published authors. Find out more about them here.

SPEAKER PROFILES

<p>AMY CHENG</p> <p>Amy is a web developer at the Brooklyn Museum of Art in New York City, working to increase dialogue between the museum and its visitors through technology, and has been a mentor for the non-profit Girls Who Code. She is interested in using code to create art.</p> 	<p>DAN CLARK</p> <p>Dan is a senior business intelligence (BI)/programming consultant specialising in Microsoft technologies. A former physics teacher, he has written several books and numerous articles on .NET programming and BI development, and is a regular conference speaker.</p> 	<p>ANDREY ADAMOVICH</p> <p>Andrey is a software craftsman with years of experience. His true love is the JVM ecosystem, and applying it to his company's DevOps initiatives. He is one of the authors of the <i>Groovy 2 Cookbook</i>, and a frequent speaker at conferences.</p> 	<p>KLAUS ASCHENBRENNER</p> <p>Klaus provides independent SQL Server Consulting Services across Europe and the US. He has worked with SQL Server 2005/2008/2012/2014 from its very beginning, and has also written the book <i>Pro SQL Server 2008</i>.</p> 	<p>AUSTIN BINGHAM</p> <p>Austin is a founder of Sixty North, a Norway-based software consulting, training and application development company. Developer of industry-leading oil reservoir modelling software in C++ and Python, he is an experienced presenter and teacher.</p> 	<p>RICHARD BLEWETT</p> <p>Richard has worked on distributed systems, including as middle-tier architect on the UK national police systems. He focuses on technologies that enable developers to build large-scale systems on the Microsoft platform, such as WCF, BizTalk, Workflow and Azure.</p> 	<p>PEARL CHEN</p> <p>Pearl's cross-disciplinary approach ranges from Android to Arduino, HTML to LEDs. Her work has taken her from Facebook campaigns for Google Chrome to projects that turn payphones into gumball machines or dynamically create origami from SMS messages.</p> 
<p>AMY CHENG</p> <p>Amy is a web developer at the Brooklyn Museum of Art in New York City, working to increase dialogue between the museum and its visitors through technology, and has been a mentor for the non-profit Girls Who Code. She is interested in using code to create art.</p> 	<p>DAN CLARK</p> <p>Dan is a senior business intelligence (BI)/programming consultant specialising in Microsoft technologies. A former physics teacher, he has written several books and numerous articles on .NET programming and BI development, and is a regular conference speaker.</p> 	<p>ANDREW CLYMER</p> <p>Andy is a co-founder of Rock Solid Knowledge, creating Kiosk-based solutions on Windows Embedded with .NET. He cut his teeth programming on a host of platforms at various start-ups, and now consults and teaches for a diverse range of clients.</p> 	<p>ED COURTENAY</p> <p>Ed is an experienced software developer and technical evangelist who has been programming professionally for more than 25 years. He currently works for a major manufacturer and retailer in the UK, leading the team responsible for its ecommerce web site.</p> 	<p>HOWARD DEINER</p> <p>Howard is a software consultant and educator who specialises in Agile process and practices. With a career spanning more than 30 years, he's been a developer, analyst, team lead, architect and project manager, and is a long-standing member of the ACM and IEEE.</p> 	<p>DINO ESPOSITO</p> <p>Dino is a trainer, speaker, consultant and author. CTO of Crionet, a company providing software and mobile services to professional sports, Dino is also technical evangelist for software developer JetBrains, focusing on Android and Kotlin.</p> 	<p>GIL FINK</p> <p>Gil is a web development expert, ASP.NET/IIS Microsoft MVP and the founder of sparXys. He consults for various enterprises and companies, where he helps to develop web and RIA-based solutions, and conducts lectures and workshops.</p> 
<p>IDO FLATOW</p> <p>Ido is a senior architect and trainer at SELA Group, a Microsoft ASP.NET/IIS MVP, and an expert on Microsoft Azure and web technologies such as WCF, ASP.NET and IIS. He has co-authored a number of books and official Microsoft courses.</p> 	<p>NEAL FORD</p> <p>Neal is director, software architect and meme wrangler at ThoughtWorks, a global IT consultancy focusing on end-to-end software development and delivery. He's the author of applications, articles, and books on a variety of subjects and technologies.</p> 	<p>SHAY FRIEDMAN</p> <p>Shay is a Visual C#/IronRuby MVP and the author of <i>IronRuby Unleashed</i>. With more than 10 years' experience in the software industry, he is the co-founder of CodeValue, a company that creates products for developers, consults and conducts courses.</p> 	<p>NUNO GODINHO</p> <p>Nuno is Director of Cloud Services, Europe at Aditi Technologies, and has more than 16 years' experience in IT. His specialities include enterprise architecture and solution architecture, cloud computing, development and training.</p> 	<p>SASHA GOLDSHTEIN</p> <p>Sasha is the CTO of Sela Group, a Microsoft C# MVP and Azure MRS, a Pluralsight author, and an international consultant and trainer. The author of two books, Sasha is a prolific blogger and author of numerous training courses.</p> 	<p>MICHAEL HABERMAN</p> <p>Michael (MCT, MCPD) is a senior consultant and lecturer specialising in rich client technologies such as WPF, Windows Phone, XNA and HTML/JS. He has helped to develop complex infrastructures using Prism, MVVM and Angular.</p> 	<p>DROR HELPER</p> <p>Dror is a senior consultant at software company CodeValue, with a decade of experience ranging from Intel and SAP to small start-ups. He evangelises Agile methodologies and test-driven design in his work, at conferences and as a consultant.</p> 
<p>SANDER HOOGENDOORN</p> <p>Sander is the author of the best-selling book <i>This Is Agile</i>. An independent mentor, trainer, programmer, architect, speaker and writer, Sander is a catalyst in the innovation of software development at many international clients.</p> 	<p>ALLAN KELLY</p> <p>Allan has held just about every job in software, before joining Software Strategy to help teams adopt and deepen Agile practices. He has written books including <i>Xanpan - Team-centric Agile Software Development</i>, and is a regular speaker and journal contributor.</p> 	<p>MICHAEL KENNEDY</p> <p>Michael is an author, instructor and technical curriculum director at DevelopMentor, and lead developer for its online training platform, LearningLine. He has been building commercial applications with .NET since its initial public beta in 2001.</p> 	<p>TOBIAS KOMISCHKE</p> <p>Tobias is Senior Director of User Experience at Infragistics, Inc., and has worked in user experience for more than 10 years. He specialises in Human Factors Engineering, which is rooted in his academic background in cognitive psychology.</p> 	<p>IQBAL KHAN</p> <p>Iqbal is the President and Technology Evangelist of software developer Alachisoft, maker of NCache, the industry's leading open-source distributed cache for .NET. NCache is also available for Microsoft Azure.</p> 	<p>PHIL LEGGETTER</p> <p>Phil is a Developer Evangelist at Caplin Systems, working on the BladeRunnerJS open source project. He writes frequently and specialises in JavaScript development and real-time web technologies.</p> 	<p>SAHIL MALIK</p> <p>Sahil, the founder and principal of Winsmarts.com, has been a Microsoft MVP and INETA speaker for 11 years. Author of books and articles about Microsoft technologies, iOS and JavaScript, Sahil helps make the most difficult topics fun.</p> 

KEYNOTE SPEAKERS



KEVLIN HENNEY

Kevlin is an independent consultant and trainer based in the UK. His development interests are in patterns, programming, practice and process. He is co-author of two volumes in the Pattern-Oriented Software Architecture series, editor of the book *97 Things Every Programmer Should Know*, and a columnist for various magazines and web sites.



ALLEN HOLUB

Allen is an internationally recognised consultant, trainer, speaker and author. He specialises in lean/Agile processes and culture, Agile-focused architecture and cloud-based web-application development. He has written a dozen books, hundreds of magazine articles, and currently blogs on Agile for Dr Dobb's Journal.

CHRISTOS MATSKAS	JULES MAY	JAMES MONTEMAGNO	JOE NATOLI	PETER O'HANLON	JOHN K. PAUL	TONI PETRINA
<p>Christos is a software engineer with more than 10 years' experience mainly focusing on the .NET stack. He has worked with big names including MarkIT, Strathclyde University, Amor/Lockheed Martin, Ignis Asset Management and Barclays.</p> 	<p>Jules is a software architect with a particular interest in languages (both for programming and discourse), presently active in web and mobile convergence. He has been writing, teaching and speaking for 25 years, and is the originator of "Problem Space Analysis".</p> 	<p>James is a Developer Evangelist at Xamarin. He has been a .NET developer for more than a decade, working in industries including games development, printer software and web services, with several published apps on iOS, Android and Windows.</p> 	<p>Joe has been preaching and practising the gospel of user and customer experience to Fortune 100, 500 and government organisations for more than 25 years. As founder of Give Good UX, he offers coaching, training and product audit programmes.</p> 	<p>Peter's fascination with new technologies has seen him writing articles for CodeProject, blogging and contributing to open-source projects. He was recently made an Intel Software Innovator for his work with RealSense technology.</p> 	<p>John is the VP of engineering at Penton Media and former lead technical architect of Condé Nast's platform engineering team. He also organises the NYC HTML5 meetup group, and contributes to a number of open-source projects.</p> 	<p>Toni is a Microsoft MVP for C#, developer, speaker, blogger and technology enthusiast. With years of professional experience working on range of technologies, his recent focus has been on Windows Phone and Windows 8 as a platform.</p> 
SHAI SKRNIK	SEB ROSE	OREN RUBIN	DEJAN SARKA	TUSHAR SHARMA	PAVEL SKRIBTSOV	GARY SHORT
<p>Shai is an AngularJS consultant working with enterprise companies, helping with migration and building large-scale projects. He recently founded HiRezio, an online training web site teaching front-end architecture with humour.</p> 	<p>Seb focuses on helping teams adopt and refine their Agile practices. The founding trainer with Kickstart Academy, he has more than 30 years' industry experience (including IBM Rational and Amazon), and is a popular speaker at international meetings.</p> 	<p>Oren has more than 16 years' experience with IBM, Cadence, Wix and others, and is the founder of Testim.io. He regularly speaks about new technologies in web development and test automation, and teaches at the Technion, Israel institute of Technology.</p> 	<p>Dejan, MCT and SQL Server MVP, is an independent consultant, trainer and developer focusing on database and business intelligence applications. He specialises in topics like data modeling, data mining and data quality, and has written 13 books.</p> 	<p>Tushar is a technical expert at the Siemens Research and Technology Center in Bangalore, India. His research into software design, design smells and refactoring has resulted in several patents, research papers and tools.</p> 	<p>Pavel is a graduate of the Moscow Institute of Physics and Technologies (MIPT), with a PhD in Neurocomputer application for the representation of static and dynamic 3D data. He is also the founder, CEO and ideological leader of Pawlin Technologies Ltd.</p> 	<p>Gary is a freelance data science practitioner and trainer. He has a deep understanding of the full Hadoop and HDInsight environment, as well as an interest in Social Network Analysis, (UCINet and Pajek) and computational linguistics (NLTK).</p> 
ROBERT SMALLSHIRE	MARK SMITH	ANTHONY SNEED	ADAM TORNHILL	RALPH DE WARGNY	MIKE WOOD	EOIN WOODS
<p>Robert is a founding director of Sixty North, a software product and consulting business in Norway. He has worked in senior architecture and technical management roles, providing tools for dealing with the masses of information flowing from today's energy sector.</p> 	<p>Mark runs the curriculum team at Xamarin University, building and managing the growing course catalogue used to train Xamarin developers all over the world. He is a Microsoft MVP, Wintellect author and Xamarin Consulting partner.</p> 	<p>Tony is a course author, instructor and consultant for Wintellect, specialising in robust, scalable and maintainable applications using Entity framework, WCF, Windows Identity Foundation and ASP.NET Web API. He is the author of two popular open-source frameworks.</p> 	<p>Adam combines degrees in engineering and psychology for a different perspective on software. An architect and programmer, he writes open-source software in a variety of languages, and is the author of <i>Your Code as a Crime Scene</i>.</p> 	<p>Ralph has spent 10 years focusing on the software development market at Intel, helping companies and institutions to maximise application performance and move code from serial to parallel. He has spoken at TechEd, TechDays, OOP and other conferences across Europe.</p> 	<p>Mike is a Technical Evangelist for Red Gate Software, on the Cerebrata Team. He describes himself as a "problem-solving, outdoorsy, user group founding, dog-loving, blog writing, solution-creating, event planning, married, technology-speaking, father-of-one".</p> 	<p>Eoin is a lead architect in the Operations Technology group at UBS. Prior to UBS, he spent 20 years in software engineering at Bull, Sybase, InterTrust and BGI. His main interests are software architecture, distributed systems, computer security and data management.</p> 

DEVWEEK 2015 PRICING STRUCTURE

BOOK NOW

All prices include refreshments, buffet lunch and session notes, but exclude travel and accommodation

BOOK BY FRIDAY
19 DECEMBER 2014
SAVE UP TO £300

BOOK BY FRIDAY
30 JANUARY 2015
SAVE UP TO £200

BOOK BY FRIDAY
6 MARCH 2015
SAVE UP TO £100

BOOK AFTER
FRIDAY
6 MARCH 2015

	BOOK BY FRIDAY 19 DECEMBER 2014 SAVE UP TO £300	BOOK BY FRIDAY 30 JANUARY 2015 SAVE UP TO £200	BOOK BY FRIDAY 6 MARCH 2015 SAVE UP TO £100	BOOK AFTER FRIDAY 6 MARCH 2015
UNIVERSAL PASS ALL 5 DAYS	£1,495 +VAT	£1,595 +VAT	£1,695 +VAT	£1,795 +VAT
4-DAY PASS MAIN CONFERENCE + PRE/POST WORKSHOP	£1,195 +VAT	£1,295 +VAT	£1,395 +VAT	£1,495 +VAT
3-DAY PASS MAIN CONFERENCE ONLY	£895 +VAT	£995 +VAT	£1,095 +VAT	£1,195 +VAT
2-DAY WORKSHOP PASS MAIN CONFERENCE 2-DAY WORKSHOP	£645 +VAT	745 +VAT	£845 +VAT	£945 +VAT
1-DAY WORKSHOP PASS PRE/POST MAIN CONFERENCE WORKSHOP	£345 +VAT	£395 +VAT	£445 +VAT	£495 +VAT

Accommodation is not included in the price of attending DevWeek 2015. If you would like to book a hotel room, please check the list of recommended hotels on the DevWeek site: www.devweek.com/about

TO REGISTER,
VISIT DEVWEEK.COM
For further information:
+44 (0)20 7407 9964
devweek@bsi.co.uk



TERMS AND CONDITIONS: Please note that the online submission of a completed registration form constitutes a firm booking, subject to the following terms and conditions. Any cancellations received after Friday 16 January 2015 will incur a 30% administration fee. Cancellations must be made in writing at least 60 days before the conference, or the full fee will be charged. We are happy to accept substitutions if they are submitted in writing before the conference begins. The organisers reserve the right to make changes to the programme and speakers without notice, if this is unavoidable. If delegates are unable to attend for any reason that is beyond the control of the organisers, such as transport problems, personal illness, bereavement, inclement weather, terrorism or Act of God, it will not be possible to make any refunds of conference or workshop fees.