TechJournal

New York Oracle Users Group

Second Quarter 2014

Second Quarter General Meeting

Tuesday, June 3, 2014
Borough of Manhattan Community College (BMCC)
199 Chambers Street

Sponsored by:
Dell Software

Free for Paid 2014 Members
Don't Miss It!

In This Issue –

Presentation Papers from the June 2014 General Meeting
Designing and Building Applications for the iPad, by Dr. Paul Dorsey
Going Mobile: New ADF UI Components for the Modern Era, by Peter Koletzke

DATABASE RECOVERY HAS NEVER BEEN SO

COMPLETE!

Axxana's award winning **Phoenix System** offering unprecedented Data Protection, Cross-application consistency, Storage and Replication agnostic.

ORACLE®
PARTNER NETWORK

AXXANA
BUILT TO LAST

info@axxana.com ▪ www.axxana.com

NYOUG Officers / Chairpersons

ELECTED OFFICERS - 2013

President
Michael Olin
president@nyoug.org

Vice President
Mike La Magna
vicepresident@nyoug.org

Executive Director
Caryl Lee Fisher
execdir@nyoug.org

Treasurer
Robert Edwards
treasurer@nyoug.org

Secretary
Cathy Wang-Wender
secretary@nyoug.org

CHAIRPERSONS

Chairperson / WebMaster
Thomas Petite
info@nyoug.org

Chairperson / Technical Journal Editor
Melanie Caffrey
editor@nyoug.org

Chairperson / Member Services
Robert Edwards
membership@nyoug.org

Chairperson / Speaker Coordinator
Caryl Lee Fisher
speakers@nyoug.org

Chairperson / Vendor Relations
Caryl Lee Fisher
vendorcoordinator@nyoug.org

Chairperson / DBA SIG
Simay Alpoge
dbasig@nyoug.org

Chairperson / Data Warehousing SIG
Vikas Sawhney
dwsig@nyoug.org

Chairperson / Web SIG
Coleman Leviter
websig@nyoug.org

Chairperson / Long Island SIG
Simay Alpoge
lisig@nyoug.org

Director / Strategic Planning
Carl Esposito
planning@nyoug.org

CHAIRPERSON / VENUE COORDINATOR

Michael Medved
venuecoordinator@nyoug.org

EDITORS – TECH JOURNAL

Associate Editor
Jonathan F. Miller
jonathanfmiller@earthlink.net

Contributing Editor
Arup Nanda - DBA Corner

Contributing Editor
Jeff Bernknopf - Developers Corner

ORACLE LIAISON Emeritus

Kim Marie Ruquet

PRESIDENTS EMERITUS OF NYOUG

Founder / President Emeritus
Moshe Tamir

President Emeritus
Tony Ziemba

Chairman / President Emeritus
Carl Esposito
cesposi@bers.nyc.gov

President Emeritus
Dr. Paul Dorsey

Table of Contents

Summer General Meeting – June 3, 2014 Agenda 5
Message from the President's Desk 10
Editor's Corner 13
Designing and Building Applications for the iPAD 14
Going Mobile: New ADF UI Components for the Modern Era 22

Legal Notice
Copyright© 2014 New York Oracle Users Group, Inc. unless otherwise indicated. All rights reserved. No part of this publication may be reprinted or reproduced without permission.

The information is provided on an "as is" basis. The authors, contributors, editors, publishers, NYOUG, Oracle Corporation shall have neither the liability nor responsibility to any person or entity with respect to any loss or damages arising from information contained in this publication or from use of programs or program segments that are included. This magazine is not a publication of Oracle Corporation nor was it produced in conjunction with Oracle Corporation.

New York Oracle Users Group, Inc.
#0208
67 Wall Street, 22nd floor
New York, NY 10005-3198
(212) 978-8890

Summer General Meeting – June 3, 2014 Agenda

Sponsored by Dell Software

AGENDA

Time	Activity	Track/Room	Presenter
8:30-9:00	**REGISTRATION AND BREAKFAST**		
9:00-9:30	**Opening Remarks** **General Information**	(single session) Richard Harris Terrace	Michael Olin NYOUG President
SESSION 1 9:30-10:30	**Oracle Key Vault Sneak Peek**	(single session) Richard Harris Terrace	Todd Bottger Oracle Corp.
10:30-10:45	**BREAK**		
SESSION 2 10:45 -11:45	**Understanding Oracle Locking Internals**	DBA Richard Harris Terrace	Arup Nanda Starwood Hotels
	Universal Theme and Template Options in Oracle APEX 5.0	Developer Room S341	Marc Sewtz Oracle Corp.
	The Right Tool for the Task: Using Mongo DB in Combination with a Traditional RDBMS	Open Source Hudson Room	Evan Bauer Dealertrack
SESSION 3 11:45 -12:30	**Ask the Experts Panel**	(single session) Richard Harris Terrace	Michael Olin Moderator
12:30 -1:30	**LUNCH - Richard Harris Terrace – Dell Presentation**		
SESSION 4 1:30-2:30	**Configure Oracle Database 12c to Tune Itself: New Features for Stability and Optimization**	DBA Richard Harris Terrace	John Watson Dave Anderson Skillbuilders
	Designing and Building Applications for the iPad	Developer Room S341	Dr. Paul Dorsey Dulcian, Inc.
	Introduction to Hadoop	Open Source Hudson Room	Vic Sawhney Amvean
2:30-2:45	**BREAK**		
SESSION 5 2:45-3:45	**Oracle Licensing – Mitigate Risk, Understand Compliance**	DBA Richard Harris Terrace	Stu Kaplan DBAK
	Calling SQL from PL/SQL the Right Way: Thinking in Sets	Developer Room S341	Grigoriy Novikov Dulcian Inc.
	Optimizing Oracle and MySQL Network Transfer	Open Source Hudson Room	Jonah H. Harris Nextgres, LLC

ABSTRACTS

9:30-10:30 AM KEYNOTE: Oracle Key Vault Sneak Peek

With many global regulations calling for data encryption, centralized and secure key management has become a need for most organizations. This session introduces Oracle Key Vault for centrally managing encryption keys, wallets, and passwords for databases and other enterprise servers. Oracle Key Vault enables large-scale deployments of Oracle Advanced Security's Transparent Data Encryption feature and secure sharing of keys between Oracle Real Application Clusters (Oracle RAC), Oracle Active Data Guard, and Oracle GoldenGate deployments. With support for industry standards such as OASIS KMIP and PKCS #11, Oracle Key Vault can centrally manage keys and passwords for other endpoints in your organization and provide greater reliability, availability, and security.

Todd Bottger is a Senior Principal Product Manager in the Oracle Database Security development team. He supports popular database features that enable customers to improve their security and comply with regulations. His main areas of emphasis are data-at-rest encryption, on-the-fly data redaction and persistent data masking of sensitive information stored in the database. Todd is a frequent speaker at Oracle OpenWorld on the topics of Oracle Advanced Security Transparent Data Encryption (TDE) and encryption key management. He has been closely involved in defining Oracle's key management strategy. Prior to Oracle, Todd held product management positions at Siebel, Sun and Protegrity. He has a B.S. in Computer Science from Brigham Young University, an MBA from the University of Michigan, and current CISSP certification from ISC2.

10:45-11:45 AM DBA TRACK: Understanding Oracle Locking Internals

Have you ever wondered how the Oracle Database keeps track of who has what type of lock and who is next in the line? In this session you will learn how locking works, the mysterious Interested Transaction Lists, how it is managed, how to get the information on ITL wait, and how to tune for resolution of locking related issues, with live examples. This presentation will help you understand in clear detail how Oracle transactions and locking work, how you can avoid locking issues and transaction contentions and design efficient applications with the knowledge gathered.

Arup Nanda (arup@proligence.com) has been working as an Oracle DBA for last 20 years touching all aspects of the database from modeling to performance tuning to security and disaster recovery. He has presented 300 sessions, written 500 articles, co-authored 5 books and delivered training in 22 countries. He is an Oracle ACE Director, a member of Oak Table Network, a member of the Board of Directors of the Exadata SIG and an editor for the SELECT journal. He was the recipient of two prestigious awards from Oracle: DBA of the Year in 2003 and Architect of the Year in 2012.

10:45-11:45 AM DEVELOPER TRACK: APEX 5.0 Universal Theme and Template Options

This presentation introduces the new Universal Theme, which is based on modern HTML5 and CSS3 standards and very flexible, thanks to its use of responsive web design principles. Marc will show how theme styles and template options can be used to choose from a variety of available styles, and to easily and declaratively customize the UI to match specific requirements. This will empower developers even without prior HTML knowledge to build applications with modern user interfaces.

Marc Sewtz is a Senior Software Development Manager at Oracle Corporation in New York. With over sixteen years of industry experience, Marc held roles in Consulting, Sales and Product Development and now manages a global team of Software Developers and Product Managers in the Oracle Application Express (APEX) development team, part of the

Oracle Database Tools group. Marc and his team are responsible for features such as the development of Mobile Web Applications with APEX, Reporting and Charting, Tabular Forms, Oracle Forms to APEX conversion and integration with Oracle Business Intelligence Publisher. Marc has an MS degree in computer science from the University of Applied Sciences in Wedel, Germany.

10:45-11:45 AM OPEN SOURCE TRACK: The Right Tool for the Task: Using MongoDB in Combination with a Traditional RDBMS

(Abstract not received)

Evan Bauer has over 30 years of experience as an executive, consultant, analyst, and software engineer in the architecture, design, implementation, and maintenance of complex information systems and software products for mission-critical applications, primarily in the financial services industry. After 8 years as an independent consultant often incubating new firms and software communities, Evan is currently Vice President and Enterprise Architect for DealerTrack, Inc., a leading provider of on-demand software and services to the automotive industry.

1:30-2:30 PM DBA TRACK: Configure Oracle Database 12c to Tune Itself: New Features for Stability and Optimization

Release 11g introduced several powerful capabilities: SQL Plan Management, Adaptive Cursor Sharing, Cardinality Feedback, and Extended Statistics. Release 12c takes these mechanisms to the next level. The result is that DBAs can use declarative techniques to ensure that your users' SQL will run even better, and better without the users changing anything!

Dave Anderson is SkillBuilders President and Founder. Dave is an Oracle DBA with 20 years of experience.

John Watson is an Oracle Certified Master consultant, lecturer, and author with twenty years' experience with the Oracle technology stack, including 11g Grid Infrastructure and RAC. John has authored 5 Oracle Press Exam Guides. He is SkillBuilders' Director of Oracle Database Services.

1:30-2:30 PM DEVELOPER TRACK: Designing and Building Applications for the iPad

Building applications for the IPAD is a completely different challenge from building for a desktop computer. Applications that work fine in a browser on a PC are frequently nearly unusable when ported to the IPAD. The user experience on a touch screen requires many modifications. It is not just about making everything bigger for fingers to be able to press buttons on screen, but also involves a total rethinking of the user experience. Technically, you have to decide whether you are going to build for the browser, make a native application or some kind of hybrid. Depending upon the technology you choose, communication with the database can become another challenge. This presentation will discuss how to take a very complicated end-user centric application that was built for a browser-based PC platform and adapt it for an IPAD platform. Our decisions, UX changes and technologies chosen will be discussed.

Dr. Paul Dorsey is the founder and president of Dulcian, Inc. an Oracle consulting firm specializing in business rules and web-based application development and chief architect of Dulcian's BRIM® tool. Paul co-authored 7 Oracle Press books on JDeveloper, UML Modeling, and Oracle database tools as well as *PL/SQL For Dummies*. He is an Oracle Ace, past IOUG and ODTUG volunteer of the year, the first inductee into the IOUG SELECT Journal Hall of Fame, and an Oracle 9i Certified Master. Dr. Dorsey's submission of a Survey Generator built to collect data for The Preeclampsia Foundation was the winner of the 2007 Oracle Fusion Middleware Developer Challenge and Oracle selected him as the 2007 PL/SQL Developer of the Year. Paul can be contacted at paul_dorsey@dulcian.com .

1:30-2:30 PM OPEN SOURCE TRACK: Introduction to Hadoop

You have probably been hearing the buzz about Big Data and how you can now reduce complex analysis tasks from days to hours or minutes. If you want to learn what the buzz is all about, this presentation will take you on an initial dive into Hadoop. It will cover the origins of Hadoop, its architecture, and include a step-by-step walkthrough of a real-time setup of a Hadoop cluster. The presentation will also cover execution of a sample analysis job including an introduction to Map Reduce, PIG, and Hive.

Vikas Sawhney is a certified Oracle DBA with 15 years of experience specializing in design of data warehousing and OLTP systems. Vikas is a managing partner at Amvean Consulting Group (www.amvean.com) headquartered in New York. Amvean specializes in design, integration, and deployment of large-scale database systems.

2:45-3:45 PM DBA TRACK: Oracle Licensing – Mitigate Risk, Understand Compliance

This session is designed to help DBAs, Developers, System Administrators, CIOs, CFOs, IT Directors, Procurement, and others responsible for Oracle licensing within their organization to understand what is at risk when it comes to Oracle Licensing compliance. The presentation provides an overview of current hardware, software, database, and support licensing. The session focuses on what you need to know about software compliance, how to gain visibility into your licensing environment, and how to gain an understanding of licensing costs and the impact on your company. The session will examine licensing in the context of technical, support, user, and business issues. It includes a look at new information about licensing for the virtual environment, new Oracle Licensing agreements (OMA/TOMA) and discusses ways to mitigate risk to your organization.

Stu Kaplan is Solutions Business Development Director for DBAK and has been configuring technology solutions for more than 30 years including software and database implementation, database upgrades, migrations, conversions, infrastructure transformation, business requirements analysis, hosting, and ongoing support requirements. His experience extends across multiple vendor (DEC, Compaq, HP and Oracle) disciplines and technologies, and he has counseled large and small companies on mapping servers to licensing, virtualization, methods of licensing, alignment of licensing, and rules and regulations for specific infrastructure for Hyperion, Oracle EBS and the Oracle technology stack. Stu focuses on straightforward communication and synergistic outcomes that bring value to the compliance environment. He has deep knowledge of broader technology stacks and technology trends that help companies keep their licensing concerns in context.

2:45-3:45 PM DEVELOPER TRACK: Calling SQL from PL/SQL the Right Way: Thinking in Sets

Everybody knows that SQL is a set language. The most effective manner in which to communicate with a set language is by using set-based operations. Otherwise, the system incurs the extra cost of transforming sets into rows and reversing rows back into sets. Oracle databases currently support multiple ways of working in sets. This presentation reviews the results of two studies focused on comparing data access approaches. The first one looks for the most efficient ways of moving significant volumes of the data between remote locations, while the second tests the impact of MULTISET operations on the resource utilization.

Grigoriy Novikov is a Senior Developer at Dulcian, Inc., where he is responsible for new feature research, SQL and PL/SQL tuning, complex functionality, developing code generators and repository-based systems development and maintenance. He specializes in the development of various industry and government systems in the medical, finance, customs, real estate and military (both US and international) arenas. He is also responsible for database modeling, researching new database features and tuning. Grigoriy holds a BS in Economics (Institute for MIS) from the State University of Management in Moscow.

2:45-3:45 PM OPEN SOURCE TRACK: Optimizing Oracle and MySQL Network Transfer

Bugs. Errors. Slowdowns. Complete meltdowns. We have all run into performance issues. Most of the time, we focus on improving the obvious: indexes, query plans, and database hardware. But what about those things which hide beneath the surface? What about the network? Sure, you have had your share of disconnections caused by idle timeouts. Do you know how to optimize the network for your database and application usage? This presentation will discuss optimizing your database networking stack by going through the MySQL and Oracle database protocols and explaining how they work, how to tune them, and how to configure their underlying network protocols, such as TCP/IP.

Jonah Harris is currently the Vice President of Software and Systems Architecture at MeetMe.com, one of the top ten social networks in the United States and one of the twenty most trafficked sites in the US. He has spoken at various conferences, consulted in different areas of technology and business, and is a known industry expert in database network protocols. In addition to his software architecture and engineering background, he has also spent many years studying statistics, psychology, and data science.

Message from the President's Desk

Michael Olin

Summer, 2014

Summertime…

Summer has just officially arrived, and with it another urgent reminder that my President's Message is overdue and its absence is holding up publication and distribution of the summer issue of the NYOUG Technical Journal. To be honest, I haven't given much thought to what I could write about, and I'm planning to watch some World Cup soccer this evening. However, I did spend some time a little over a month ago trying to put some profound thoughts to paper. In mid-May, I delivered the commencement address to the College of Computing and Information at the University at Albany, almost thirty years after I earned my degrees there. Several retired faculty with whom I studied and worked still live locally, and the College was kind enough to assemble them along with the current faculty so that we could share a meal prior to the commencement ceremony. At lunch, I asked the Dean of the College about another faculty member who was not present. I was told that he was one of the last of their cohorts to retire, and was finally convinced that it was time to call it quits after one of his students remarked that his father has taken the same introductory course with that professor when he was a student. The commencement followed lunch. I delivered my remarks, walked around the campus with my wife (also an alum), who was last on campus more than a decade ago, and we drove home. A day or two later, I received an email from the professor who had invited me to speak. He arrived at the University at Albany a year or two before I graduated and was now planning to retire as well. He thanked me for coming to speak and told me that my remarks were well received. He then told me that at the university-wide commencement ceremony the day after the College commencement, he met a former student who was there for his child's commencement. They spoke about the prior day's ceremony and the professor mentioned that I had spoken to the graduates. A look of recognition crossed the former student's face and he told the professor that he remembered me. I had been the teaching assistant in his introductory computer science class decades ago. If only I had decided to become an academic, I now know that it would be time to retire!

Having provided a convoluted introduction to this message, it is time for the columnar cop-out. Instead of original thoughts for the summer, here are some recycled ones, namely my commencement address to the graduates of the College of Computing and Information at the University at Albany.

College of Computing and Information - University at Albany
Commencement Address
Delivered by Michael Olin '85, '86
Albany, NY - May 17, 2014

Dean Faerman, faculty, family, friends, and most importantly, graduates of the College of Computing and Information: I doubt that I could have ever envisioned myself uttering these words thirty years ago, but "It's great to be back in Albany!" Perhaps the sentiment should not be that much of a surprise to me. I've stayed involved with UAlbany since I was sitting where you are today, although back then we referred to this institution as "SUNYA." Quite a bit has changed since I was a student. The Downtown Campus has been completely renovated; the Uptown Campus has been expanded with new academic buildings, dormitories, a basketball arena, the football stadium and Bob Ford Field; the University added the East Campus and across Fuller Road, billions of dollars have been invested in NanoTech. The world records for the largest games of musical chairs and Twister that I helped set with a few thousand of my closest friends on campus no longer stand and Mayfest is a distant memory, along with the electrifying performance in 1983 of a relatively unknown group of 20-something Irish rockers who called themselves "U2".

The field of computing has changed just as dramatically, and not just in a Moore's Law smaller, faster, cheaper way. By the time I arrived at UAlbany in the fall of 1982, many businesses had automated operational tasks such as payroll, inventory and order entry using computers. Large businesses used mainframe computers or the minicomputers that had become so popular in the 1970's. Mid-sized businesses often did their computing using timeshare services, renting time on a computer that they could not afford to purchase outright. While the need for employees with programming and related skills was increasing, most large businesses were able to fill the positions with a combination of in-house training and the hiring of new computer science graduates. Although there had been quite a few personal desktop computers that came before it, the introduction of the IBM PC in 1981 changed everything. All of a sudden, computers were not just tools for big businesses or toys for hobbyists. IBM marketed the PC as a computer for everyone. By the time I graduated, almost six million PCs had been sold worldwide. Over the next four years, those sales exploded to over 60 million, and to date over 3.5 billion devices have been sold that qualify as "PC compatible." When I graduated in the mid-1980's, there was an acute shortage of programmers. I don't recall that any of my fellow computer science graduates didn't have the luxury of choosing from multiple job offers. I had been accepted to law school and was seriously considering pursuing a career in "Computer Law," which was a relatively new and limited discipline at the time, focused mainly on contracts and intellectual property. However, I chose to accept a job in a corporate IT department because the opportunities at the time seemed better. For a while, that was true. From the time I graduated until the turn of the century, you could jump from job to job every year or two, increasing your compensation by double-digit percentages with each move. I specialized in designing and developing relational databases (thank you Professors Ravi and Willard), and easily moved from assignment to assignment as an independent consultant. Everyone, it seemed was using computers to automate absolutely everything. This explosion in the use of information technology sent the demand for skilled professionals through the roof. As computing moved towards ubiquity, these inexpensive machines became necessary tools, and an integral part of just about every business's daily operations. The online revolution soon followed and the "tech bubble" created countless software millionaires. Many businesses thought that information technology could be used to gain a competitive advantage, and they invested heavily in both products and people. It was a great time to be working in IT.

A little over a decade ago, in 2003, Nicholas Carr, a writer who, at the time, was editor of the Harvard Business Review, wrote an article entitled "IT Doesn't Matter," which he followed up the next year with a book "Does IT Matter?" In the article, he compares the corporate adoption of information technology to the adoption of earlier technologies like railroads and electric power. At the time, corporate executives had fully embraced the idea that information technology provided a strategic advantage and that the way their companies made use of IT would differentiate them in the marketplace. Carr suggested that they were completely wrong. He argued:

> *Behind the change in thinking lies a simple assumption: that as IT's potency and ubiquity have increased, so too has its strategic value. It's a reasonable assumption, even an intuitive one. But it's mistaken. What makes a resource truly strategic – what gives it the capacity to be the basis for a sustained competitive advantage – is not ubiquity, but scarcity. You only gain an edge over rivals by having or doing something that they can't have or do.*

Industry reaction to the article came quickly and over the next several months rebuttals to his thesis came from the top executives at Microsoft, HP, Intel, Cisco and many others. The debate was covered in The New York Times, magazines such as Fortune and Wired, as well as industry publications such as Computerworld. A more nuanced view of Carr's article would have acknowledged that in some respects, he was absolutely correct. In cases where computing had replaced pen-and-paper business functions such as payroll, order entry and invoicing, there was no competitive advantage to be gained. On the other hand, there was a strategic advantage to be gained by other uses of information technology. Financial services firms were quite confident that their proprietary quantitative analysis and algorithmic trading provided them with an advantage in the marketplace.

Carr's prediction of the commoditization of IT was cemented, however, by the next trend that swept through corporate America. The widespread adoption of outsourcing placed IT firmly on the cost center side of the ledger. Successful CIOs were rewarded for how little they managed to spend, often without much thought to the value they provided to the business. In just a few years, the value of the average IT professional dropped significantly. Tens of thousands of IT workers were laid off and the work that they had been doing was contracted out, often overseas. It is a bit of an

oversimplification, but the difference between contracting for office supplies and contracting for information technology services has been getting smaller and smaller. We constantly hear from politicians, businesspeople, and educators that there is an enormous shortage of workers with STEM skills. Yet every year there are tens, if not hundreds of thousands of people all over the world entering the workforce with enough proficiency in these areas to meet the demand for the commodity that corporate IT has become. And here we are.

Once again, however, Moore's law has reshaped the industry, this time, with big data. The first IBM PC was able to address 32 megabytes of disk storage. A one-gigabyte disk drive for a mainframe computer was housed in a six-foot tall cabinet, about the same size as a typical data center rack today. My son got a new cell phone earlier this month, and was able to augment its 16 gigabytes of internal storage with a 128-gigabyte microSD card that is less than half a square inch in size. Large databases have long since blown past sizes measured in mega- or giga- or even terabytes. We've moved on to the next orders of magnitude, exa-, peta-, zetta- and yottabytes. A decade after Carr's article asserting that "IT Doesn't Matter," the Harvard Business Review published an article entitled "Data Scientist – The Sexiest Job of the 21st Century." Of course, this article generated just as much hand wringing as Carr's. The premise has been contested in articles and blog posts, including Information Week's "Data Scientist: The Sexist Job No One Has." Whether or not anyone actually hires "Data Scientists" is not that important. What is crucial, however, is that just as the ubiquity of computing generally led to the commoditization of IT, leveraging big data provides a roadmap to how businesses can once again use information technology to derive a sustained competitive advantage.

The Data Scientist, if such a person actually exists, mines the vast store of data that has been accumulated to discover trends and correlations that are hidden among the petabytes. These insights are then used to guide the development of new products and services, and improve the marketing of existing ones. In other words, the data scientist's work leads to increases on the revenue side of the ledger, where commoditized IT was focused on decreasing the expense side of the ledger. The vast middle ground between a profit-center and a cost-center was lost somewhere along the way to commoditization. People in corporate IT have always had the ability to contribute to increasing revenues, but this knowledge was shunted aside during the past decade's focus on cost reductions.

I've been fortunate to be able to spend the bulk of my career on assignments where my focus was on projects that helped the businesses I worked for leverage technology to get better at whatever it is that they do. What I discovered is that the key to being able to add value rather than just manage costs is to make sure that you have a holistic view of the organization where you work. Don't just focus on the specific requirements of the project in front of you. Understand what the business does, what generates revenue, where its profits come from. While it can be exciting to be heads-down coding, racing towards the next release, spending all of your time banging away at a keyboard is a sure way to be pigeonholed as a commodity resource. Cultivate relationships with people outside of the IT department. Talk to the people that IT tends to refer to collectively as "the business". Understand how they do their jobs and think about how technology can make them better at what they do. Don't spend all of your days locked inside the IT bubble. What you've learned here at UAlbany allows you to look at things through a different lens than how they are seen by the business. Combining your insights with theirs can steer information technology back towards providing a competitive advantage. Even if you're not heading from here to the sexiest job of the 21st century, you can help ensure once again, and going forward, that IT really does matter.

Michael Olin
President, NYOUG

Editor's Corner

Melanie Caffrey

The Last Issue

This is the very last issue of the NYOUG Technical Journal as you know it. That does not mean that the NYOUG will no longer publish articles from its many speakers and members, merely that we will now be doing so in a new format. We invite all those who have articles to share to submit them to execdir@nyoug.org (as you have always done), for publication in the soon-to-be-published NYOUG blog, which will be linked from http://www.nyoug.org. Our hope is that those who wish to submit articles and other types of technical expertise will feel free to use our blog to share their findings and knowledge with fellow members, speakers and experts. We sincerely hope you keep your ideas and articles coming, and that you find our new blog format easier and more flexible for your use and ability to continue to contribute to the NYOUG membership.

I have enjoyed being your editor for the NYOUG technical journal, and hope to read (or meet and swap knowledge personally) with you, in the future. Thanks very much to all of the wonderful authors who have shared their knowledge selflessly and tirelessly. The NYOUG technical journal would never have existed without your generosity. We invite each and every one of you to continue that spirit of authorship with the new NYOUG blog. We look forward to reading you soon.

Kind regards,
Melanie Caffrey

SIGS, SIGS and more SIGS!

The following Special Interest Groups (SIG) hold meetings throughout the year for the benefit of NYOUG members:

DBA SIG – Database Administration
Data Warehouse SIG – Business Intelligence
Web SIG – Web / XML / Java / Weblogic / APEX / Fusion
Long Island SIG – Nassau/Suffolk area - All topics

Designing and Building Applications for the iPAD

Dr. Paul Dorsey, Dulcian, Inc.

Abstract

Building applications for the iPAD is a different challenge from building for desktop computers. Applications that work fine in a browser on a PC are frequently nearly unusable when ported to the iPAD. The user experience on a touch screen requires many modifications. It is not just about making everything bigger for fingers to be able to press buttons on screen, but also involves a total rethinking of the user experience.

Technically, you have to decide whether you are going to build for the browser, make a native application, or some kind of hybrid. Depending upon the technology you choose, communication with the database may be an additional challenge.

This paper discusses how to take a very complicated end-user centric application that was built for a browser-based PC platform and adapt it for the iPAD platform. Our decisions, UX changes and technologies chosen will be discussed.

Target Audience

Anyone developing applications for the iPAD or attempting to port an application designed for a desktop environment will find our experience and tips useful.

Executive Summary

With the increasing popularity of smaller portable computers, tablet computers, and iPADs, there is a great demand for systems to work on these devices. This paper describes some of the challenges and issues encountered when developing a system that previously ran on desktop computers to run quickly and efficiently on an iPAD, as well as the solutions discovered to address them.

Case Study

Our company was tasked with building a patient check-in application for doctors' offices to support patients entering their complete medical histories, verifying personal information, correcting errors, and paying any fees or co-payments with an attached credit card swiper.

From an application user interface perspective, this was a very challenging assignment since the users would vary widely in their levels of computer literacy. Many would be elderly patients with no previous exposure to computers at all. The application needed to be very user-friendly, never requiring the patient to ask questions of the doctors' office staff. Any time required of the office staff to assist the patients with the application would be viewed as a failure of the software. The application also needed to be engaging and keep the patients' attention through completion of entering the information that would otherwise be done by the office staff members.

It was decided to build this system using an application generator. There were several parts to the task:

1. Specify the repository: How should we capture the metadata that would define the application? An application had to be created to populate the repository.
2. Define the screens and the information to be captured on each screen: This information was then entered into the repository as metadata.
3. Define the user experience: Decide how the screen navigation would work and how each component would be displayed.
4. Write a generator to take the metadata and generate the application. The generator would generate the database and the screens, bind the screens to the database, and generate the page flow logic code.

The reason for using a generator approach was to leverage the following benefits:

- The requirement was to support hundreds of different clients. We did not want to have hundreds of hand-coded applications to maintain.
- The goal was to generate to desktops, iPADs, and mobile phones without rewriting the application each time.
- The application should include the ability to make changes to the user experience without having to make hundreds of changes to the application. In order to change how a particular component would be displayed, it could simply be implemented by a change to a CSS class.

Before starting the project, we had already built a generator to support a desktop application. The generator had been cross-tested to ensure that the generated code would work for all major browsers and versions.
We thought that converting the application to work on the iPAD in Safari (not as a native application) would be relatively simple since everything was contained in the generator. This turned out to be far from the truth.

Basic User Interface

What seemed like the easiest part of the process of porting a PC-desktop application to an iPAD actually ended up being the most difficult. When we first opened the desktop application version, everything “mostly” worked. The iPAD’s 1024 x 768 resolution is lower than that of most desktop monitors but was sufficient for the application. The iPAD also has an onscreen keyboard and web widgets, which worked in Safari. However, there were a number of challenges associated with how users would interact with the application.

1) Support for the Computer Illiterate

There were many challenges involved with supporting users who had never used a computer before. Even for those who had previously used a computer might never have seen a touch screen, so basic concepts such as scrolling might be foreign to them. We tried creating scroll-up and scroll-down buttons to support the scrolling challenged, but these buttons caused more confusion than they solved.

Fortunately, our experience was that totally computer illiterate users were rarer than expected. For example, one patient over 60 who described himself as "not a computer kind of guy" had both a smart phone and an iPAD at home, so he had no difficulty using the application. It was ultimately decided that if a person was totally computer illiterate that there was little we could do to make our application completely suitable.

2) Tablet versus Desktop

Web-based applications have a different feel on tablets than applications designed for a desktop computer. Screens tend to be simpler, with fewer components. The main design idea is to make it friendly looking rather than efficient. One idea we embraced early on was to limit ourselves to a single component per row in the application. For example, "Last Name" would be on the first row and "First Name" would be on the second row of the screen. There is actually research about this topic indicating that users tend to prefer this layout. It also provides space for help text to the right of the field and sufficient room for a label to the left of the field. Since a generator was being used, the simpler the structure, the easier it is to make the generator work properly.

There were a few places where multiple objects were placed on the same screen. Multiple check boxes ("Select all the medical issues you have") looked better with two or three check boxes on each line. Otherwise, there was a significant scrolling issue.

3) Life without a Mouse

Designing an application that works without a computer mouse means that buttons must be large enough to accommodate large fingers. More space must be provided between items to prevent selection errors. As a result, checkboxes, radio groups, and text boxes must all be redesigned. LOVs do not work at all unless huge fonts are used for the list items.

Since everything must be bigger, there are two alternatives: use many screens or require scrolling on all screens. The ability to include help text is also limited. In short, the entire user interface must be rethought from the ground up when designing for an iPAD or other mobile devices such as smartphones. This also affects the entire user experience.

4) Be Prepared for the Absolutely Bizarre

We ran into things that just made us shake our heads in amazement. We had a part of our application that would not allow users to type a lower case "r". Every other letter worked just fine (upper and lower case). The problem existed in Chrome, but not in Internet Explorer. It turns out that we were using F3 to do something in the application and "r" was being interpreted as F3 in Chrome (You just have to love cross-browser compatibility!)

Performance on the iPAD

One need only Google "iPAD JavaScript performance" to find many well-justified complaints about iPAD performance. System architectures that are composed largely of JavaScript rarely encounter the same types of performance problems on other platforms. With the advent of the iPAD, sub-second performance suddenly required 10 seconds for many common actions. The issue is the underpowered engine of the iPAD. Anything that requires the execution of JavaScript takes time. A long script was timed on various machines with the following results:

1. An Acer Chrome book --- 10 seconds
2. A desktop running Chrome --- 15 seconds
3. iPAD Air -- 20 seconds
4. iPAD 4 -- 40 seconds
5. iPAD 2 -- never came back

The biggest surprise was that an inexpensive Acer Chromebook was much faster than a desktop. It seems that having a system specifically tuned to run a browser has its advantages.

Transitioning a desktop application to the iPAD resulted in many unnecessary objects placed in the Document Object Model (DOM). Apparently, iPADs are particularly sensitive to large numbers of objects in the DOM, which negatively affected performance. Tweaking the application generator to get rid of these objects markedly improved the application performance.

A great deal of tweaking was required in the generator. The goal was not to create any unnecessary objects at all. We eliminated hundreds of unnecessary frames slowly whittled down the performance issues.

There were a number of other problematic elements. Even native radio groups performed sluggishly on the iPAD despite responding instantaneously on a desktop computer. It was discovered that when triggering a Tap event on the iPAD, the device waits 300ms to see if a second tap comes which acts as the equivalent of a double mouse-click, which is the built-in support for zoom. If a mouse-click event is translated into a touch event, this causes a lag on the iPAD interface. In order to fix this, a touch event was substituted and the problem was solved.

Scrolling was also very slow, jerky, and/or sluggish on the iPAD (as much as 20x slower than on a desktop) because of the use of scrollable DIVs. We ended up having to set -webkit-overflow-scrolling :touch. Internally, we refer to this as the "goFaster=YES" setting.

Without a PC browser tool, it was difficult to determine what actions worked quickly or slowly on the iPAD. Breaking down the time lags was a challenge. To accomplish this, it was necessary to connect the iPAD to an Apple computer and use Safari Developer mode to track performance issues. Of course, this meant having access to a Mac in order to do the testing.

The last major task was to cache our screens on the client and then only refresh the renderer at runtime. When the application started, all of the screens were cached resulting in a 10-second performance hit. Then the page-to-page load time was decreased by 50% and loading at approximately 1 second to go from page to page.

If the DOM is large enough, it exceeds the memory of the device and the browser abnormally terminates. We ran into this issue when caching all of the screens in a large application. In Apple's defense, performance is improving. The new iPAD Air is twice as fast as the iPAD 4, but is still 50% slower than a desktop computer. Chromebook was a bit better and only 30% slower than a desktop for web-based applications.

Locking It Down

After looking at different device management software, nCLOSE was selected. Typically, device management software uses profiles or roles. What was needed in this situation was a single profile that would not allow patients to bypass the application in any way. The device would open and run the desired application without a security code required to unlock the device. The nCLOSE software was reasonably inexpensive, easy to set up and configure, and met the client requirements.

Screen Orientation

Since a desktop computer does not rotate, there is never an issue of screen orientation. However, on an iPAD, users are accustomed to turning the device and having the screen rotate as needed. This was problematic for the application being designed. The alternatives were to turn off the screen rotation feature or make the screens adapt to both orientations. Turning off the screen rotation was done in this situation to eliminate the challenge of changing screen orientation on the fly for a multi-screen complex application.

Similarly, the Zoom control can be challenging for JavaScript. For this application, it was disabled in order to simplify the application and avoid other problems.

Adaptive Design

It is now common for web screens to perform differently if the screen resolution is different. Solving this problem varies from fairly trivial to very complex using a number of different strategies:

1. Code a different page for each resolution to be supported.
2. Make screens minimally adaptive. Usually this involves a border layout and docking some components East and West. That way, when the width of the page changes, the user does not see excessive blank space on the left and right sides of the form.
3. Make smart CSS classes that dynamically change font and component size based upon the screen size.

For this project, we did not make the screen particularly adaptive. A nice border was added on the side. Since this was to be used as an in-office application, we were able to guarantee both the resolution and aspect ratio.

Individual Component Design

There are many components to redesign when moving a desktop application to an iPAD platform. This section describes each component and how the modifications were accomplished.

Basic Fields

As shown in Figure 1, there is nothing particularly different except the size of the fields. There is enough space for the user to click the field (which then pops up the on-screen keyboard). The field itself is wider (top to bottom) than necessary and there is white space between the fields. The font sizes were increased from those used in the desktop version. Notice that there is only one field per row, and the required field "*" is clear and obvious.

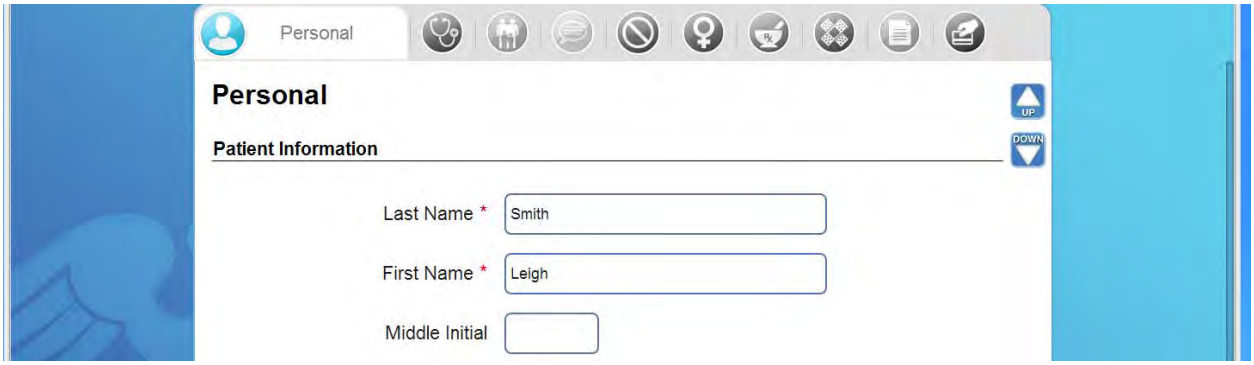


Figure 1: Sample iPAD Fields

LOVs

Native LOVs did not work properly. It was necessary to keep the text large and user-friendly with a long list of values from which users could easy select. This ended up being a completely new component as shown in Figure 2.

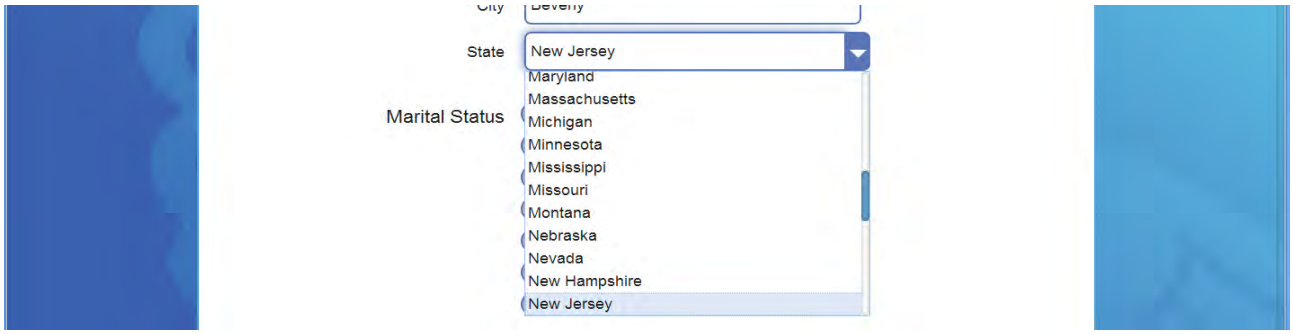


Figure 2: Sample LOV on iPAD

Radio Buttons

Native radio buttons were much too small so we created our own images for a selected and unselected radio button. As shown in Figure 3. In order to customize radio buttons and checkboxes, it was necessary to create a label for the checkbox/radio button and then use CSS styling on the label while hiding the actual checkbox or radio button.
This also required changing the triggering event from Click to Touch in order to avoid the 300ms delay where the iPAD waits for a possible double click.

Figure 3: Customized Radio Buttons

Simple Yes No Questions

For Yes/No questions, the selection images were made even bigger as shown in Figure 4.

Figure 4: Yes/No Selection Images

Checkboxes

Checkboxes formatting was one of the few places where more than one component was placed on a single line as shown in Figure 5. Without this layout, the page scrolling was too long, making for a terrible user experience.

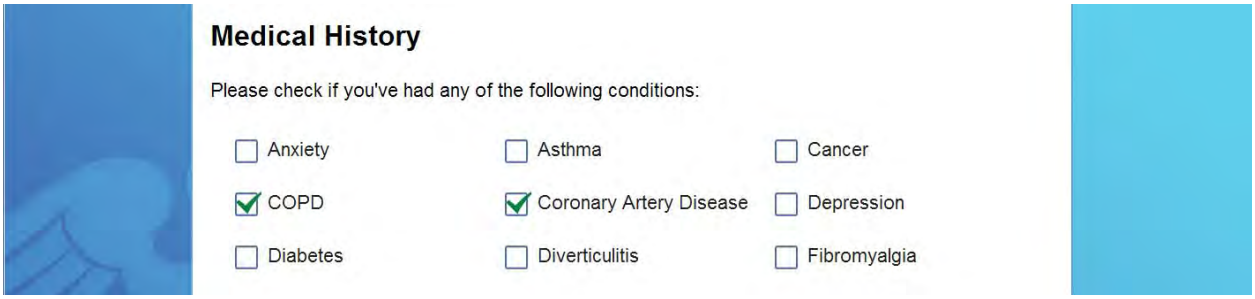


Figure 5: Checkbox Layout

Multi-Record Objects

Supporting multi-record groups was the biggest challenge. The goal was to be completely consistent so that every multi-record object would be the same. A "Have you ever…? Question was added in most places sp that prior to asking for objects, it can be determined whether or not there are any objects required. For example, rather asking about all of your children, the application first asks whether you have any children. Only if you answer "Yes" are additional questions about your children available. Using this approach meant that a number of possible combinations were needed:

1. Yes/No question is unanswered.
2. Yes/No question was previously answered "No" and you are returning to the page.
3. Yes/No question was previously answered "Yes" but no records were entered.
4. Last record is deleted.
5. A record is deleted but there are other records

   …and on and on for another 20 or so situations (states).

Each state has its own UI and various actions, which change the state of the group. We ended up filling up a whiteboard with a state diagram to map out the behavior of the object. The coding was relatively simple; figuring out what to code was difficult.

The result was to open an editing panel within the grid, which seemed to provide the best user experience as shown in Figure 6.

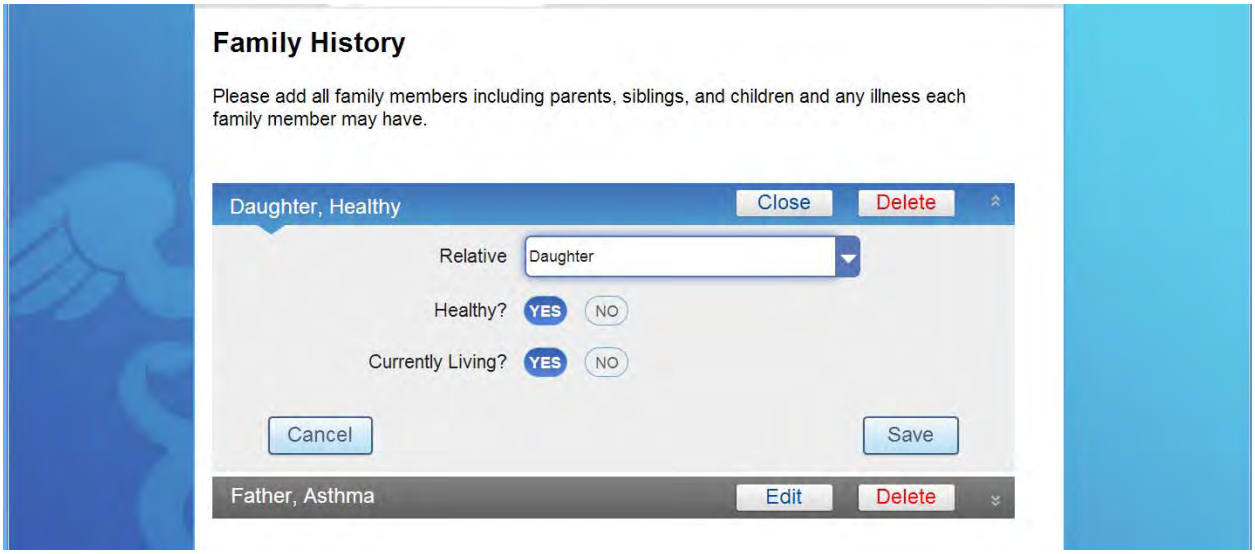


Figure 6: Multi-Record Object Editing Panel

Conclusions

Making a web application work on an iPAD was a lot more work than what we expected. However, it ended up being possible to solve all the problems that we encountered, but the journey was neither as short nor as painless as predicted.

About the Author

Dr. Paul Dorsey is the founder and president of Dulcian, Inc. an Oracle consulting firm specializing in business rules and web-based application development. He is the chief architect of Dulcian's Business Rules Information Manager (BRIM®) tool. Paul is the co-author of eight Oracle Press books on Designer, Database Design, Developer, JDeveloper and Performance Tuning, which have been translated into nine languages as well as the Wiley Press book *PL/SQL for Dummies*. Paul is an Oracle Ace, and the first inductee into the IOUG SELECT Journal Hall of Fame. He is President Emeritus of NYOUG. In 2003, Dr. Dorsey was honored by ODTUG as volunteer of the year, in 2001 by IOUG as volunteer of the year and by Oracle as one of the six initial honorary Oracle 9*i* Certified Masters. Dr. Dorsey's submission of a Survey Generator built to collect data for The Preeclampsia Foundation was the winner of the 2007 Oracle Fusion Middleware Developer Challenge and Oracle selected him as the 2007 PL/SQL Developer of the Year. Paul can be contacted at paul_dorsey@dulcian.com .

Going Mobile: New ADF UI Components for the Modern Era

Peter Koletzke, Quovera

I'm goin' home
And when I wanna go home
I'm goin' mobile
Well, I'm gonna find a home
And we'll see how it feels
Goin' mobile
Keep me movin'

—Pete Townshend (1945–), *Going Mobile* (from the *Who's Next* album)

Oracle has joined the modern era.

I'm sure the folks who inhabit the green Oracle HQ towers in Redwood Shores, California would counter that Oracle has *driven* the modern era. My bold, mildly politically-incorrect pronouncement refers to some new user interface (UI) controls—more properly called "components"—that are now available to developers using Oracle Application Development Framework (ADF) in JDeveloper and Eclipse. These components fulfill current trends in mobile device interfaces and are therefore "very modern." (See the sidebar "A Word about ADF" for more about the term "ADF.")

This article describes and shows examples of three of these components so you can understand why they are not merely more tools in an already crowded toolbox—they are markers of the modern era. The descriptions and pictures in this article can go only so far.

So that you can experience how these components act and work, the conclusion of this article points to a sample test drive application that demonstrates these components. You can download and run this application in JDeveloper 12*c*, and even follow the hands-on practice included in the download to experience developing the test drive application.

Before launching into a demonstration of these components, a brief review of the evolution of Oracle user interface technologies is in order.

A Word About ADF

In this article as well as in current, Oracle-world parlance, the term "ADF" is used to refer to the technologies specific to Application Development Framework (for example, ADF Business Components, ADF Model, and in this situation, ADF Faces). Moreover, now that applications with most ADF technologies can be developed in either JDeveloper or Eclipse, when you work with ADF, you are considered an "ADF developer" rather than a "JDeveloper developer" or "Eclipse developer."

Oracle UIs Past to Present

For many years, Oracle Applications, Oracle's business applications suite, used Oracle Forms as its main user interface—first in terminal-server mode, then in client/server mode, finally in a web browser running a Java applet. After Oracle Applications was reworked into Oracle E-Business Suite (EBS), day-to-day data input screens built in Oracle Forms were supplemented with lightweight self-service, web browser, no-Java-applet-required screens using a proprietary Oracle technology: *UIX*—loosely expanded into "User Interface XML" (within which "XML" stands for "Extensible Markup Language").

In EBS, Oracle Forms still supplied the requirements of fast data input and rich display needed for screens like sales order entry because the web browser components at that time were not capable of that kind of productivity. Still, UIX supplied Forms-like functionality to the self-service screens through many innovative, highly-interactive components. An example

of these components is a table control whose contents update without refreshing the entire page when the user scrolls up or down or sorts records by clicking the column headings. Although these components were on the industry's cutting edge, they only came close to but could still not match the interactivity users had become accustomed to in Oracle Forms applications.

When the JavaServer Faces (JSF) standard became mainstream, Oracle had already migrated its UIX components to be fully compliant with the JSF specifications. This component set, named "ADF Faces" and first available with JDeveloper/ADF 10*g*, increased the number of rich, interactive components available for web interface development. It also offered *declarative AJAX* (also called "partial page rendering"), which allows ADF developers to set up fast round-trip server events using only component property settings. (Refer to the next sidebar, "About AJAX" for an overview of AJAX.) With declarative AJAX, the developer is freed from the normal AJAX development burden of coding custom JavaScript functions and servlet code.

About AJAX

AJAX (or "Ajax") stands for "Asynchronous JavaScript and XML," and describes a combination of techniques that allow components on the web page to communicate with the server in the background. A summary of typical AJAX steps follows:

1. A JavaScript function creates an XMLHttpRequest object and uses it to send an XML message to the server; for example, after a user enters a postal code, the function would send an XML message containing the postal code to the server.
2. A custom servlet processes the request and sends a message back to the page; for example, the servlet would query a table to find the postal code and would return the associated city, state or province, and country.
3. Another JavaScript *callback* function accepts the values from the return message and places them in the proper locations on the page; for example, the function would fill in the city, state or province, and country fields.

The benefits of AJAX events are the small programming footprint, speed, and interactivity. That is, a small message is sent to the server; the server runs a very specific, and hopefully fast, program to form a return value; and the return value fills the values into one or more components on the page without redrawing the rest of the page.

ADF Faces in ADF (JDeveloper) 10*g* begat ADF Faces Rich Client (ADF Faces RC) in ADF 11*g*. ADF Faces RC (hereafter shortened to "ADF Faces") introduced even more components as well as extremely rich visual components like graphs, maps, and hierarchy viewers. Declarative AJAX moved forward even more with this version, and these components are now being used in Oracle's newest flagship application product, Oracle Fusion Applications ("Fusion Apps").

Although Oracle is using ADF 11*g* to develop Fusion Apps, it is also adding features to a new version: ADF 12*c* (released in July 2013). Some of these features, such as the components in this article's test drive application, are being ported back and forth between 11*g* and 12*g* versions so both versions can offer the new functionality.

But What About Mobile?

That's the question you're probably asking since mobile didn't appear in the preceding history lesson. It's actually a question nearly everyone is asking; you've probably heard the prediction that the number of mobile devices in the world is predicted to exceed the world population in 2014.

Oracle is investing heavily in user experience (UX) research and design primarily for Fusion Applications; lately, much of the UX effort has been on design for mobile devices. As a result, ADF has evolved again to solve the mobile requirement by offering the following alternative user interface technologies, each of which uses a separate set of components:

- **ADF Faces components**: used for desktop, tablet, and touch devices. It is very strong on network connectivity (to and from a remote database for example). Current version: 12*c* (12.1.2).
- **ADF Mobile components**: used for Android and iOS mobile phones; this option also supports tablets and allows the application to access services on the device such as the camera, contacts, and GPS. It is not as strong as ADF Faces for network connectivity and may use a SQLite database local to the device that will be synchronized with a remote database. Current version: 11*g* (11.1.2.4).

Although the application shown in this article uses the first option, ADF Faces components, the available components, development techniques, and resulting user interface are similar for the second option.

Mobile Design Principles

Regardless of the ADF method, the application developer/designer needs to accommodate the shift in the human interface. This has always been the case when interfaces changed.

For example, in the days of terminal-server, character mode applications, character and function key presses were the only way to input data and navigate an application. When graphical user interfaces (GUIs) appeared on the scene, application designers needed to consider the effects of mouse clicks and mouse movements in addition to keyboard key presses. Applications could now contain graphical elements like buttons that users could click to perform a certain function.

Mobile applications require another shift in design thinking because the human interface now favors hand gestures (finger swipes and taps) rather than mouse and keyboard actions. Most mobile devices also offer on-screen keyboard interfaces so mobile applications can use a combination of “traditional” typing and “modern” hand gestures.

In addition to the paradigm shift for human interfaces, the application designer can be faced with a requirement that the application support multiple device types: desktop computer and tablet, for example. However, the newer components, especially the newer components in ADF Faces, are built to support multiple device types: for example, a mouse click on an icon in a desktop environment is the same as a finger tap on the same icon in a tablet environment.

Design Shift Example

Think about how the design of an application menu has changed over the years based on the current human interface. In the days of character-mode applications a menu would often display as a list of functions with a number or letter next to each; a single field would allow input of the number or letter, and submitting that page would navigate to the function indicated in the field.

In GUI days, a menu could have been a menu bar with pulldown selections, or a navigation bar with multiple icons or buttons; selecting or clicking a menu selection, icon, or button would navigate the application to the selected function.

In mobile days, a menu can be rendered as a set of icons indicating available functions, which could look something like this:

This view of options is particularly familiar to mobile “smart” phone users, who often see the available apps on their cellphone arranged in this way. In a desktop environment, the user would click one of the icons with a mouse to activate the selection; in a tablet environment, the user would finger tap an icon to activate the selection.

Once the selection is made (for example, Employees), the Employees page loads and the icons rearrange themselves at the top of the page as shown in Figure 1. The selected function is highlighted but the other functions are available for selection at any time, somewhat like a tab component.

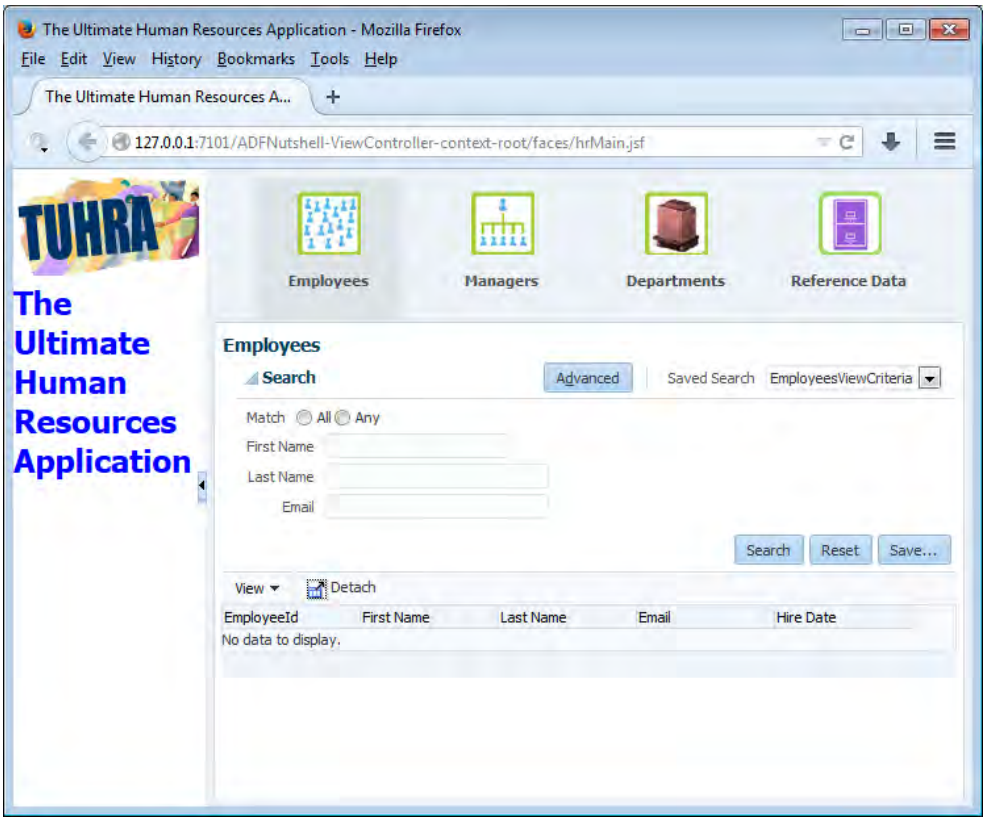


Figure 1. Employees Page

New ADF Components

These screenshots are taken from the test drive application mentioned earlier. This application showcases these three new components that were designed to take Oracle's user interface experience into the mobile realm:

- Panel Springboard
- Panel Drawer
- List View

Panel Springboard

This component, which appears in the source code as `af:panelSpringboard`, is responsible for the icon arrangement and rearrangement just mentioned. It is also available in ADF Mobile as `adfmf:springboard`. It holds another component, `af:showDetailItem`, for each icon displayed in the grid. For example, the code to render the four icon menu within the springboard component is the following:

```
<af:panelSpringboard id="ps2" displayMode="grid">
  <af:showDetailItem text="Employees" id="sdi1" icon="/images/peopleOff.png">
    <af:region value="#{bindings.empflow1.regionModel}" id="r1"/>
  </af:showDetailItem>
  <af:showDetailItem text="Managers" id="sdi2" icon="/images/managerOff.png">
    <af:region value="#{bindings.managerflow1.regionModel}" id="r2"/>
  </af:showDetailItem>
  <af:showDetailItem text="Departments" id="sdi3" icon="/images/buildingOff.png">
    <af:region value="#{bindings.deptflow1.regionModel}" id="r3"/>
  </af:showDetailItem>
  <af:showDetailItem text="Reference Data" id="sdi4" icon="/images/filecabOff.png">
    <af:region value="#{bindings.jobflow1.regionModel}" id="r4"/>
  </af:showDetailItem>
```

```
</af:panelSpringboard>
```

Notice that the displayMode property is set to “grid”; this defines the starting arrangement as a grouping of icons in the center of the display as shown in the first screenshot. You can alternatively set *displayMode* to “strip” to show the icons in a line as shown in Figure 1. Clicking an icon while the component is in grid mode will shift the display into strip mode automatically and will display the content in the `af:showDetailItem` as shown in Figure 1.
In the preceding code snippet, `af:region` defines a task flow that has been developed in another file. The task flows in this application are a single view (JSF page fragment) that displays data relevant to the selection, for example, Departments data for the deptflow1 flow.
To save space in this article, the Panel Springboard code has been abbreviated. Each `af:showItemDetail` can define separate icons for the “at rest” (initial) display, a mouse hover, disabled, and depressed (selected) as follows:

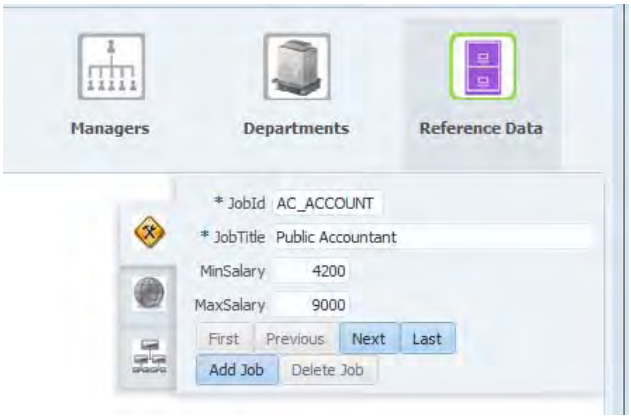
```
<af:showDetailItem text="Employees" id="sdi1" icon="/images/peopleOff.png"
                   depressedIcon="/images/peopleOn.png"
                   disabledIcon="/images/peopleOff.png"
                   hoverIcon="/images/peopleOn.png">
    <af:region value="#{bindings.empflow1.regionModel}" id="r1"/>
</af:showDetailItem>
```

Panel Drawer

The Panel Drawer, `af:panelDrawer`, component renders tabs attached to one side of its parent component. Clicking a tab opens the “drawer” and displays its contents. ADF Mobile has no exact equivalent but an ADF Mobile design could accomplish an equivalent effect with another embedded Panel Springboard.
The test drive application’s Reference Data option displays a Panel Drawer as shown on the right. The tabs can appear on the left or right side of the parent container (or start or end side, if you are working with an application that needs to support multiple reading directions).
Clicking a tab in this component displays some content just as clicking an icon in the Panel Springboard displays content. The code behind this component is very similar to the code behind the Panel Springboard:

```
<af:panelDrawer id="pd1" position="right">
  <af:showDetailItem text="Jobs" id="sdi5" icon="/images/toolsOff.png">
    <af:region value="#{bindings.jobflow1.regionModel}" id="r4"/>
  </af:showDetailItem>
  <af:showDetailItem text="Locations" id="sdi6" icon="/images/globeOff.png">
    <af:region value="#{bindings.countryLocflow1.regionModel}" id="r5"/>
  </af:showDetailItem>
  <af:showDetailItem text="Regions" id="sdi7" icon="/images/orgchartOff.png">
    <af:region value="#{bindings.regionflow1.regionModel}" id="r6"/>
  </af:showDetailItem>
</af:panelDrawer>
```

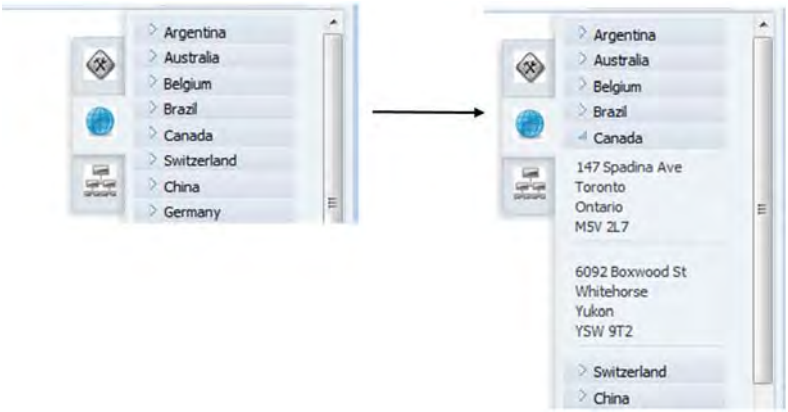
Clicking a tab, like Jobs (the top icon), pulls out the drawer and displays the region task flow (JSF page fragment) inside the `af:showDetailItem` component as shown next:



Subsequent tab selections retract the visible drawer and pull out the selected drawer. The `af:showItemDetail` inside this component offers the distinct icon properties mentioned before (at rest, mouse hover, disabled, and depressed).
With tabs on the right (or left) a tablet user can select an icon with the thumb on the hand holding the tablet. The size of the extended drawer will be the size of the initial content. If the content changes, the drawer will not resize but you can alternatively set width and height attributes to declare the size of the drawer.

List View

The List View component, `af:listView`, is responsible for displaying a collection of data (set of rows). It also supplies functionality familiar to smartphone users. That is clicking (or tapping) a selection in the list can reveal a panel with details about the item. It is available in ADF Mobile 11*g* as `adfmf:navigationBar`.
As a static demonstration, consider the following flow from the Locations page (second drawer of Reference Data) in the test drive application:

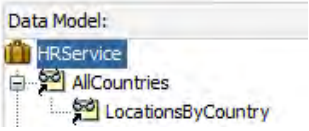


Clicking the show details “>” icon next to Canada, as in the snippet on the left, expands a panel under it showing the offices in Canada. The code to display this component looks like this:

```
<af:listView value="#{bindings.AllCountries.treeModel}" var="item"
           emptyText="#{bindings.AllCountries.viewable ? 'No data to display.' :
                      'Access Denied.'}"
           fetchSize="#{bindings.AllCountries.rangeSize}" id="lv1">
  <af:listItem id="li1">
    <af:panelGroupLayout id="pgl1" layout="vertical">
      <af:outputFormatted value="#{item.bindings.StreetAddress.inputValue}"
                          id="of1"/>
      <af:outputFormatted value="#{item.bindings.City.inputValue}"
```

```
                          id="of2"/>
      <af:outputFormatted value="#{item.bindings.StateProvince.inputValue}"
                          id="of3"/>
      <af:outputFormatted value="#{item.bindings.PostalCode.inputValue}"
                          id="of4"/>
      <af:separator id="s1"/>
    </af:panelGroupLayout>
  </af:listItem>
  <f:facet name="groupHeaderStamp">
    <af:listItem id="li2">
        <af:panelGroupLayout id="pgl2">
          <af:outputFormatted value="#{item.bindings.CountryName.inputValue}"
                              id="of5"/>
        </af:panelGroupLayout>
    </af:listItem>
  </f:facet>
</af:listView>
```

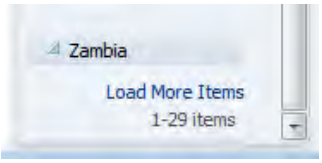
Naturally, this display requires queries to the database, and view objects in the ADF Business Components (ADF BC) layer of the application define these queries. The data model of the ADF BC application module looks like this for the queries used by the `af:listView`:



AllCountries represents a view object that queries the COUNTRIES table; LocationsByCountry defines a view object query on the LOCATIONS table and is joined to AllCountries with a view link (foreign key relationship) defined on the COUNTRY_ID column. The view link is responsible for displaying the relevant locations for the currently selected country.

Since List View is a multi-row (collection) object, it uses a tree binding with nodes for the country name and locations attributes. The component's `groupHeaderStamp` facet contains an af:listItem that displays the country name. The component's direct child component, `af:listItem`, displays the location information. The example in this application uses read-only, `af:outputFormatted`, components, but it could just as well use `af:inputText` components if data entry was required.

Notice from the flow screenshot that the component automatically renders a scrollbar to navigate through the list returned by the AllCountries query. It also automatically displays a "Load More Items" selection such as the following if the scrollbar cannot cover the entire list returned from the query:



This component can be used to find a value from a list and display details as in the preceding example. It can also be used as a navigation list: so when a value from the list is selected, appropriate data is selected in adjacent regions. The test drive application contains an example of this use in its Managers page as shown in Figure 2.

Selecting a manager's name on the left queries the manager's record on the right and under that record is a read-only list of employees reporting to that manager. The List View component in this example is simpler because it only needs to display the manager's name, without any "show detail" expanding panel functionality. The List View code looks like this:

```
<af:listView value="#{bindings.AllManagers.collectionModel}" var="item"
           emptyText="#{bindings.AllManagers.viewable ? 'No data to display.' :
                      'Access Denied.'}"
           fetchSize="#{bindings.AllManagers.rangeSize}" id="lv1"
                      selection="single"
           selectionListener="#{bindings.AllManagers.treeModel.makeCurrent}">
  <af:listItem id="li1">
    <af:panelGroupLayout id="pgl1">
      <af:outputFormatted value="#{item.bindings.FirstName.inputValue}" id="of1"/>
      <af:spacer width="7" height="10" id="s1"/>
      <af:outputFormatted value="#{item.bindings.LastName.inputValue}" id="of2"/>
    </af:panelGroupLayout>
  </af:listItem>
</af:listView>
```

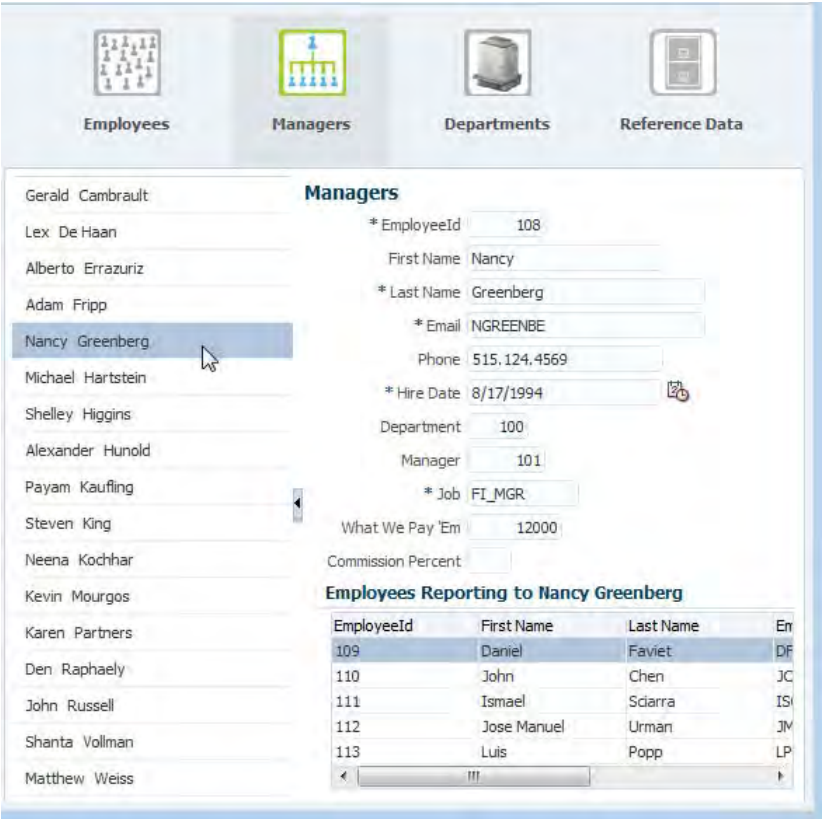


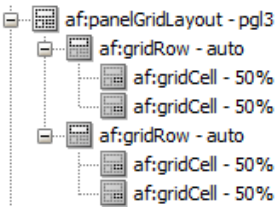
Figure 2. Managers Page

The data model for the List View component does not require a nested collection like the preceding country-location example. It is a simple list based on a single view object. However, the page includes a nested collection for the employee detail table. In addition, the page includes declarative AJAX (partial page rendering) settings for updating the form and table values on the right when a manager is selected on the left. Refer to the tutorial that accompanies the hands-on test drive application for implementation details.

Bonus Component: Panel Grid Layout

Although out of scope for this article's sample application, another important, new component was added with ADF 12*c* that can assist in solving design challenges in both desktop and mobile applications.

Panel Grid Layout is another choice in the group of *layout components*: a component that holds other components. Panel Grid Layout, `af:panelGridLayout`, allows you to specify a layout of a certain number of rows and columns, with the ability to use HTML-like properties such as column spans, row spans, and margins. It emulates the Java Swing GridBagLayout or GridLayout layout managers by using embedded grid rows and grid cells. For example, a simple two-row, two-column layout would look display as follows in the Structure Window:



Before this component was available, the grid functionality usually required embedding multiple layers of layout components, and this often resulted in inconsistent and sometimes unpredictable behavior. Panel Grid Layout greatly reduces the need for complex, multi-level layout component designs, and is now a highly-recommended layout component for most multi-cell needs.

Conclusion

This article has explained how Oracle user interface technologies have evolved and the benefits derived from each step in that evolution. It also explained how the relatively recent change in human interfaces has required a paradigm shift in designing applications. The latest shift has been to mobile applications, so the article provided examples of how Oracle is meeting the mobile challenge by explaining the functionality, effects, and some detail about the coding for three relatively new ADF user interface components: Panel Springboard, Panel Drawer, and List View. It also briefly mentioned the place of honor immediately taken by another new component: Panel Grid Layout.

The components this article introduces demonstrate how ADF has evolved to meet the human interface needs of a mobile application that may need to be rendered on many different display devices. The three components explained in this article are a small subset of all the ADF components available to build a mobile-aware application so this article is just a starting point for learning about ADF's mobile components.

The Next Step – a Test Drive

This stuff is kind of exciting. Even if you disagree, you may still want to try out the application mentioned in this article. The sidebar "Sample Application" contains details of where you can find it.

In addition, you may also want to try out some development work with these components. The test drive application also includes a tutorial you can use to create the application demonstrated in this article so that you can obtain hands-on experience with these components. A sample solution is also available (the finished test drive application) so you can check your work.

Running this test drive application and working on the hands-on tutorial will give you a good feel for how ADF handles the needs of a mobile application. With the overview in this article, the test drive, and tutorial, you will be equipped with some of the knowledge you will need should you find yourself goin' mobile.

Sample Application

You can find the application and tutorial mentioned in this article in at www.quovera.com: click White Papers and look for this NYOUG Technical Journal article. It may also be available at www.nyoug.org.

About the Author

Peter Koletzke is a technical director and principal instructor for Quovera, in Mountain View, California, and has 30 years of industry experience, 25 of which is in the Oracle arena. Peter has presented at various Oracle users group conferences more than 310 times and has won awards such as Pinnacle Publishing's Technical Achievement, Oracle

Development Tools Users Group (ODTUG) Editor's Choice (three times), ODTUG Best Speaker, ODTUG Volunteer of the Year, NYOUG Editor’s Choice (three times), and ECO/SEOUC Oracle Designer Award. He is an Oracle Certified Master, Oracle ACE Director, and coauthor (variously with Dr. Paul Dorsey, Avrom Roy-Faderman, and Duncan Mills) of eight Oracle Press development tools books including *Oracle JDeveloper 11g Handbook*.

Nothing hunts down Oracle performance issues like Confio Ignite™.

Over 50% of DBAs who try Ignite resolve a performance problem on the first day.

Start your **free trial at Confio.com/p-hog**

© 2012 Confio Software, Boulder, Colorado. (303) 938-8282

CONFIO SOFTWARE

NYOUG 2014 Sponsors

The New York Oracle Users Group wishes to thank the following companies for their generous support.

Dell Software (software.dell.com)
Oracle (www.oracle.com)

Contact Caryl Lee Fisher (execdir@nyoug.org) for vendor information, sponsorship, and benefits

#1 Middleware

- ✓ #1 in Application Servers
- ✓ #1 in Application Infrastructure Suites
- ✓ #1 in Enterprise Performance Management

ORACLE®

oracle.com/goto/middleware
or call 1.800.ORACLE.1

Copyright © 2009, Oracle. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.