Rene Antunez
RMAN in 12c - The Next Generation

Karen Cannell
APEX 5 New Features & Upgrade Advice

Kevin McGinley
Oracle BI Development

Tom Peters
Disable Database Triggers

Kellyn Pot’Vin-Gorman
A Roll of the Dice with Raspberry Pi

Board Focus - Tim Gorman
WE BUILD SOLUTIONS THAT LEAD TO SUCCESS

Rolta AdvizeX brings you a complete solution for all your application, data, and infrastructure needs. Our customers use technology to drive innovation and achieve their most important business goals because we first help them focus on a strategy that leads to success, then bring the data, applications and infrastructure expertise needed to create and sustain that success.

To explore how we can help your organization, contact Rolta AdvizeX at 800-220-8454 or visit Rolta AdvizeX online at www.advizex.com

#CloneAttack

What’s in it for you?

Going to RMOUG Training Days 2016? Try Delphix #CloneAttack free for 90 days. Install your own ready-to-use virtual machines and get hands-on experience with this important new capability.

Delphix experts will be ready to assist, demonstrate, or answer your questions in the lab at the rear of the exhibitor hall. Find out more about #CloneAttack on Twitter or contact us at cloneattack@delphix.com
On the Cover:
Cover photo by Sylvan Creach, taken in Twin Lakes a few months ago.

Monthly Features

4 Letter From Our President
5 Stan Yellott Scholarship Fund
26 Board Focus - Tim Gorman
28 RMOUG Board of Directors
30 November 2015 QEW

Featured Articles

6 Oracle BI Development
by Kevin McGinley
Making the Case for Adopting Agile and Continuous Integration

10 APEX Interactive Reports Deep Dive
by Karen Cannell
Part 2: New Features and Upgrade Advice

16 Raspberry Pi
by Kellyn Pot’Vin-Gorman
A Roll of the Dice

19 Database Triggers
by Tom Peters
How To Disable for a Single Session

20 RMAN in 12c
by Rene Antunez
The Next Generation
What is the Oracle ACE program?

The Oracle ACE program is a community advocacy program for Oracle technology evangelists and enthusiasts, sponsored by and managed by Oracle Corporation. As stated on the ACE overview page at "http://oracle.com/technetwork/community/oracle-ace", it is both a network as well as a resource for everyone in the Oracle community. It is not a certification program, and there is no course to study and no test to pass. Rather, the ACE program is a recognition program by one’s colleagues and peers, and joining is a process of nomination, review, and acceptance.

Who are Oracle ACEs?

They are your colleagues in the worldwide Oracle technology community, of which you and the RMOUG community here in Colorado are a significant part. There are now more than 500 people in 55 countries around the world who have been recognized by their peers and by Oracle Corporation. They are not employees of Oracle, but rather partners and customers of Oracle.

The ACE program is now 10 years old and comprises three levels...

ACE Associate - this is the entry-point for the program and an ideal form of recognition for technology community advocates who are building their reputations

ACE - established advocates for Oracle technology who are well-known in their community

ACE Directors - top-tier members of the worldwide community who engage with many user groups and advocates and with Oracle Corporation

The following past and current members of the ACE program are also members of the current RMOUG board of directors...

1. Bobby Curtis - ACE Director
2. Chris Ostrowski - ACE
3. Dan Hotka - ACE Director
4. John Jeunnette - ACE
5. Kellyn Pot'vin-Gorman - ACE Director (retired - now an Oracle employee)
6. Rene Antunez - ACE
7. Tim Gorman - ACE Director

So RMOUG is currently very well represented with ACEs with fully half of our large board of directors so honored, and in the past yet more ACEs have served with the board, such as David Fitzjarrel, Kent Graziano, Brad Brown, John King, and George Trujillo.

Moreover, at our annual Training Days conference, Denver is fairly overrun by yet more Oracle ACEs, with upwards of 35 in recent years. Over the past few conferences, RMOUG has initiated “Lunch with ACEs” with an ACE at each table so that conference attendees have a chance to ask questions, share experiences, and just generally interact with these experienced and skilled enthusiasts.

Colorado is also home to the senior manager of the global Oracle ACE Program, Vikki Lira, which makes for one terrific party at the RMOUG “Training Days” conference each year!

The ACE program is always growing, and the current members of the program are always happy to help you make the step up and jump to the next level in your career. Essentially, ACEs are expected to contribute to all or some of the following: by participation on technical forums like OTN or the ORACLE-L list, by publishing articles in newsletters, magazines, blogs, or in books, to tweet, post, and participate in community activities such as Oracle user groups (like RMOUG). Few people do all these things, but there are some who do, driving their careers above and beyond with the 21st century.

RMOUG provides several ways to build your qualifications for inclusion in the ACE program. You can speak at meetups, quarterly educational workshops (QEWs) or the annual “Training Days” conference. You can publish your expertise in the award-winning RMOUG newsletter, “SQL>Update”. You can join the board and help the community as an organizer and mover-shaker.

Joining the ACE program is not a one-time task, but an on-going commitment to participation and sharing to build your community, whether that community is local to Colorado or worldwide.

To find out more about the program, go to the web page at “http://oracle.com/technetwork/community/oracle-ace/become-an-ace” or just contact one of the Oracle ACEs near you and ask questions.

That’s why we’re here.

Tim Gorman
The Scholarship Fund started in 2001 to encourage future IT professionals in their efforts to broaden their knowledge. In 2007, RMOUG voted to rename the scholarship fund to honor the memory of Stan Yellott. Stan was a long time member of RMOUG where he supported the user community by serving on the RMOUG board. Stan focused on expanding Oracle educational opportunities. Stan’s vision was to include high school and college students as the next generation of IT professionals.

**Application Deadlines**
- March 15th for an April scholarship distribution
- June 15th for a July scholarship distribution
- September 15th for an October scholarship distribution
- December 15th for a January scholarship distribution for the following year in January

*For Details, Visit the RMOUG Website www.rmoug.org*
Making The Case For Adopting Agile and Continuous Integration In Oracle BI Development

by Kevin McGinley

What Is The Waterfall?

Close your eyes. OK, maybe not quite yet. Read the next few sentences, then close your eyes. Once your eyes are closed, imagine every IT project you’ve ever been a part of, whether as an employee of the company owning the project or a consultant contributing your valuable skills toward the successful completion of the project. Wrap your mind around each project, considering its flaws and successes, its challenges and hurdles. Go ahead and close your eyes now, and when you open them again, try to recall the singular thread that weaves through all of the projects you can remember.

Are you back? Was that a pleasant experience? A painful one? A little of both? For most of us, there are likely both pleasant and painful memories across the entirety of our careers. There are projects we liked, projects we didn’t like, projects that went well, projects that didn’t, and sometimes the painful ones are the ones we liked the most because we overcame a challenge or hurdle that was difficult yet rewarding.

If you focus on the instruction at the end of the first paragraph, “recall the singular thread,” I’d be willing to bet that at least some of you thought about the words, “change,” or “unknown,” or “unplanned,” or something similar. Perhaps a few more of you, upon reading these words, thought to yourselves, “Yes, actually that’s what I was thinking of, but couldn’t really articulate it in a single word or phrase.” While someone somewhere is bound to point out an exception, it is nearly universal across every project that something unplanned happened. Something changed that you couldn’t have foreseen.

If we can agree on that premise, it’s curious then that many organizations adopt a development lifecycle methodology that refuses to accept that change is the norm, not the exception. The methodology I refer to is commonly called “waterfall” and started sometime in the 1970’s in manufacturing and construction.

The basic premise of the waterfall methodology, as shown in the picture above, is that you gather every requirement necessary to build the thing in question, working under the premise that finding out new requirements later is costly to address. You then design every aspect of those requirements, build the things you design, then test them to make sure they are correct. If you think about it, using waterfall in manufacturing and construction makes sense because it can be physically prohibitive to change a building after you build it. Knowing every last detail about the building you’re going to construct is important because changes can not only be costly, but physically difficult or even impossible.

The world of IT adopted the waterfall methodology because it provided a nice structure to complete IT project work, but also because many IT systems and applications from yesteryear were monolithic and difficult to change on the fly. Data Warehousing and Business Intelligence fell in line with the rest of the IT projects because, frankly, at the time it was difficult to iterate and BI was viewed not as value-add, but rather as a necessary afterthought. The tools, the skills, the perception...waterfall made sense. As time went by, few really questioned the waterfall approach in BI and Data Warehousing until it became so ingrained in how we do projects, that as professionals we just accepted it, never questioning it. I’m reminded of a scene from “The Matrix” (edited and parodied for my own purposes here):

Morpheus: I imagine that right now you’re feeling a bit like Alice. Tumbling down the rabbit hole?
Neo: You could say that.
Morpheus: I can see it in your eyes. You have the look of a man who accepts what he sees because he’s expecting to wake up. Ironically, this is not far from the truth. Do you believe in running projects without project plans, Neo?
Neo: No.
Morpheus: Why not?
Neo: ‘Cause I don’t like the idea that I’m not in control of my BI projects.
Morpheus: I know exactly what you mean. Let me tell you why you’re here. You’re here because you know something. What you know, you can’t explain. But you feel it. You felt it your entire BI & Data Warehousing career. That there’s something wrong with the world of BI Development. You don’t know what it is, but it’s there. Like a splinter in your mind -- driving you mad. It is this feeling that has brought you to me. Do you know what I’m talking about?
Neo: The Waterfall?
Morpheus: Do you want to know what it is?
   (Neo nods his head.)
Morpheus: The Waterfall is everywhere, it is all around us. Even now, in this very room. You can see it when you build your star schemas, or when you gather requirements. You can feel it when you execute your test cycles, or when you promote to production or when you develop dashboards. It is the world that has been pulled over your eyes to blind you from the truth.
Neo: What truth?
Morpheus: That there is a better way, Neo. Like everyone else, you were taught a singular way, taught a process that you blindly accept. A prison for your mind. (long pause, sighs) Unfortunately, no one can be told what The Waterfall is. You have to see it for yourself. This is your last chance. After this, there is no turning back.
   (In his left hand, Morpheus shows a blue pill.)
Morpheus: You take the blue pill and the story ends. You wake up at your desk and believe whatever you want to believe.
   (a red pill is shown in his other hand) You take the red pill and you stay in Wonderland and I show you how deep the rabbit-hole goes. (Long pause; Neo begins to reach for the red pill) Remember -- all I am offering is the truth, nothing more.
   (Neo takes the red pill and swallows it with a glass of water)

My argument is simple. The waterfall approach is a methodology we have blindly followed in the BI and Data Warehousing space because of inertia. Our blind adherence to this methodology has even made us oblivious to the fact that other development methodologies even exist at all. In “The Matrix”, the malicious machines stimulate our senses with perceptions of a fake world meant to keep us from realizing that the human race is actually imprisoned. For BI & Data Warehouse developers everywhere, “The Matrix” is a perfect metaphor for how we have been conditioned to think that The Waterfall is required for everything we do. But as in “The Matrix,” we have the ability to wake-up, because there is another way.

As the author I make one simple request: open your mind. While reading the rest of this article, you may be tempted to mumble retorts based on conventional wisdom and long held beliefs. Resist this urge and remind yourself that you are likely reading these articles because your current process today, while painfully simple to continue, is probably not really meeting your needs, or your end users’ needs.

Over the course of this article, I will discuss the benefits of Continuous Integration and the Agile methodology when applied to Business Intelligence development (and in this case, specifically Oracle). While many readers have likely heard of Agile, Continuous Integration is probably a new term. While they are both complementary to each other, it makes sense to discuss the need and benefits of Continuous Integration first, so let’s go there next.

What Is Continuous Integration?

Martin Fowler, a software engineer who authored a paper on the topic more than a decade ago, defines it this way:

“Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly.”

Let’s break down that definition and examine why and how this should be done for OBIEE RPD development, just like in other development platforms. The primary theme in the first sentence of the definition is “integrate frequently.” This means that developed code should be merged with the entire code base on a regular basis; Fowler suggests daily, at the minimum.

In OBIEE RPD development, customers generally take one of two approaches. The first approach is to let developers access an “online” RPD simultaneously. An online RPD is actively deployed to a BI Server and changes made to the RPD are reflected on the running BI Server as the RPD is saved. There are two primary problems with this approach. When multiple developers work online, they tend to collide and lock each other out of parts of the RPD. This is because OBIEE “checks-out” and locks an object and all its related objects as you work on them. Those locks not only block developers from working, but they can lead to a lock-up of the entire RPD, destroying work in progress and requiring a redeploy of the RPD. The second problem is that there is no traceability of what was done, when, and by whom. There is no way to know which developer did what, changes can’t easily be undone, and bad code can wreck a live environment. As such, multi-user development on an online RPD is discouraged.

The other approach is “offline” RPD development. This approach can take one of two flavors: true offline development of an RPD that isn’t deployed to a BI Server, or online RPD development on an isolated OBIEE server. In both approaches, true integration must take place in the form of a merge. The “offline” RPD containing the new code must be merged with the master RPD and the master RPD must be redeployed to a live OBIEE server. The merge typically follows OBIEE’s “three-way-merge” process, which is done using the OBIEE Admin tool. A problem with offline RPD development is the merge; it’s simplistic, results in frequent conflicts, and generally gives everyone headaches.
Part of the reason for these headaches is actually a matter of development practice, not a problem with the tool. Yes, the three-way-merge is an inelegant merge technique, but the bigger problem is that OBIEE RPD developers wait too long to integrate their work. Offline developers will wait days, even weeks before integrating their code because it's not easy to integrate. The philosophy behind CI is that integrating more frequently means less integration at once, which leads to fewer integration problems and easier troubleshooting when they do occur. Adopting a simple process change of making fewer changes at once before merging can go a long way to improving development in OBIEE - if we can make it simpler to integrate.

Returning to Fowler's definition of CI, the other major theme to examine is automation, specifically automating the integration build and automating the integration test. Here is where CI pays dividends for those who use it and leads to the claims Fowler makes at the end of the definition: rapid development of cohesive software. To do this though, we must stray from CI as a practice and introduce CI as a technology: the CI server. The CI server handles the process of assembling code builds and running regression tests against the build. There are several CI server technologies in the marketplace and Oracle actually owns one of them: Hudson. The open-source Jenkins was forked from Hudson when Oracle acquired it and serves as a popular free option today. Atlassian also owns a CI products called Bamboo, and its elegant interface combined with integration to JIRA make it a popular choice as well.

Of course, the catalyst that really enables CI to do its thing is source control. Most software development shops use it, but when you look at companies using Oracle BI, few do. A well selected, designed, and executed source control system enables developers to check-out RPD code on task branches, merge it with mainline development code, and use CI to test it immediately. The key is smart branch design, which enables the ability to track everything, cherry-pick features for promotion, and simplify automated deployments of RPD code.

What does this look like when you put it together?

You'll notice that it starts with developers having their own full-stack OBIEE servers to develop on. This allows them to develop in on-line mode without fear of competing with other developers. They can check-out the latest code from the source control system and check-in their changes on their own task branches. Once that code is merged using source control (not OBIEE), the CI server automatically builds a working RPD, deploys it to an OBIEE build server, and runs a series of logical SQL regression tests to insure nothing has broken. Successes (no failed tests) result in new RPD code available in source control for developers to check-out when building the next feature, while failures result in notifications to a manager or developer lead, who can address the failures with the developer responsible for the code (something we know thanks to source control).

The key to making this work well is a development methodology that actually embraces frequent check-in of code: Agile. A key thrust behind Agile is to get working software into users hands as quickly as possible, and CI really helps to make that happen.

**Agile All The Way**

Even if you’re not familiar with the Agile methodology, you can probably guess based on the word “agile” that it’s designed to be both fast and nimble. Where a Waterfall project is planned to be fairly structured and predictable, an Agile one sacrifices some structure and predictability to be fluid and allow for change. It is important, though, to mention that there is a well developed methodology behind Agile - it isn’t the Wild West of development. It is not my intention, however, to explore all the intricacies and dogma of the Capital-A-Agile methodology, but rather consider the basic principles so your Oracle BI projects can be Lowercase-a-agile.

The Agile methodology was invented in 2001 with the publishing of the Agile Manifesto by 17 developers who wished to uncover better ways to develop software. Supporting that manifesto were 12 principles:

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances
The two primary things that tend to differentiate Agile from Waterfall are “what you know” and “when you know them.” Specifically, requirements and testing outcomes are staged at different points in the process for each methodology. In Waterfall, we want to know everything up-front, build all of those things to spec, then test them all to make sure they produce the desired results. There are two inherent flaws with this in the world of BI: we don’t always know what we want, and if we wait until the end to find errors, it becomes more costly to fix them. In Agile, we acknowledge that we don’t know everything yet, so we keep our unit of work (project) short so we can adapt quickly to change. We also work in a constant state of build/test, build/test, instead of one long state of build, followed by one long state of test. This allows two things to happen: users see outcomes faster and give immediate feedback, and errors or feedback are easier to address, because the unit of measure is small. It is much easier to fix days of code than months of code.

The second dramatic difference between Agile and Waterfall is documentation. It’s not that one has it and the other doesn’t, but rather the priorities and the styles are very different. The easiest way to contrast the two is to talk about how one might document the OBIEE RPD. In a Waterfall project, you would most likely spend countless hours creating a document that outlines hundreds if not thousands of dimensions, hierarchies, attributes, facts, measures, formulas, and more. This document would serve as the input for developers to build from, and they would go build all of those objects. Over time, maybe even within the project, some of the objects would change or new objects would be added, and the document would need to be updated, then the code would need to change. The document is only as good as the discipline to keep it consistent with the code, and many hours are spent building and maintaining the document.

In Agile, loose guidelines are added to a “backlog” or “queue” in the form of a user request, or “story” that states something like, “I want ABCD.” The story may not have outlined everything perfectly, but the developer will work closely with the user to uncover some of those details through face-to-face conversation. Some of the details won’t be uncovered, but that’s because user won’t know them yet or maybe hadn’t thought to communicate them. Through repeated iteration and constant testing, the developer will uncover additional characteristics through user feedback and the code will evolve toward a more perfect state. And what about that document outlining the entire contents of the RPD? Why do we need that if we have the RPD itself? We spend a lot of time creating an alternate version of the RPD, especially since the Admin tool we use to build the RPD can dynamically generate an alternate version on demand to be viewed in a tool like Microsoft Excel. In Waterfall, we spend a lot of timing creating the same thing twice, have a hard time keeping both in sync, and the main person who gets frustrated is the user.

So if we bring this all together, how does this work for OBIEE development? There are a lot of ways to orchestrate the details, but the process fundamentally has these 8 things:

1. What a user wants is entered into some sort of ticketing system, possibly by the user himself, and many tickets constitute a backlog of work to do.
2. Items on the backlog are prioritized and estimated based on need and complexity. Whether you play “planning poker” or not is irrelevant, as long as you understand what is most important and what capacity you have to complete work.
3. From there the team begins a unit of work, or a project, sprint, release, etc. Again, there are many ways to organize this effort, but the effort should be 1-4 weeks, no matter how you slice it.
4. OBIEE RPD and front-end development begins from the user stories and with user engagement. Do as much as possible in the front-end without necessarily having all of the RPD pieces in place. If you’re doing RPD development, check the RPD out of source control and work against feature branches aligned to the stories in your ticketing system. Check-in work frequently after you’ve built code and regression tests to check your code.
5. Build and test every day, night, or some other short duration. Continuous Integration can run the builds of the RPD and the regression tests automatically after check-in or in batch on a schedule. Front-end development should be ‘tested’ by users with them consistently having access to the catalog objects being built and face-to-face feedback should be consistently provided.
6. Communicate daily. Set aside a short amount of time for the team to discuss obstacles each day. Design time or clarifying discussions should take place 1:1 so as not to disrupt others.
7. Don’t worry if it’s not perfect. More difficult refinements or attempts to follow standards and best practices can be accomplished by adding new stories to the backlog to “refactor” code that isn’t ideal. Maybe a front-end formula needs to be added to the RPD. Perhaps a hard-coded filter should have a repository variable instead. All of these things should be done, but not at the expense of getting something to the user quickly.
8. Rinse and repeat. Take the time at the end of the cycle to discuss what went well and what went wrong, but don’t dwell on it. The backlog is growing. Begin prioritizing, estimating, and planning the next release.

Shifting from a Waterfall approach to an Agile approach backed by Continuous Integration is not a simple activity. It requires changes across people, process, and technology to make it happen. But if you recognize that we now live in a world where the business is more capable than ever to deliver their own analytics, and they aren’t going to wait long for IT to deliver, then you must also recognize that it’s not a matter of if you shift, but when you shift.
APEX 5 Interactive Reports Deep Dive
New Features & Upgrade Advice - Part 2

by Karen Cannell, TH Technology

Abstract
APEX Interactive Reports have undergone major changes in APEX 5.0, bringing big new features and significant changes behind the scenes, changes that impact all interactive reports. Understanding these changes is essential for any developer who has enhanced or customized interactive reports and is planning on upgrading to APEX 5. Part 1 of this series introduced the APEX 5 Interactive Report (IR) new features. This paper, Part 2 of the series, takes a deep dive into the structural and code changes behind the scenes that make these improvements possible.

Target Audience
All APEX developers should be aware of the APEX 5 new features in order to pass these features on to their users and to leverage new development efficiencies. All developers who have customized interactive report appearance, altered Actions, written IR dynamic actions or otherwise enhanced an IR in any earlier APEX version (most of us?) need to know the behind-the-scenes details of APEX 5.0 IRs. The changes are significant, and unless the APEX standard APIs have been used the customization may not upgrade smoothly.

Background
Introduced in APEX 3.2, IRs have become the default report option for APEX applications. IRs enable developers to deliver a wealth of end user functionality with a minimum of effort.

Note: Most developers and end users are familiar with the search, filter, sort and other interactive features. The paper assumes readers are familiar with pre-APEX 5 IR’s. For those who are not, a full description can be found in this Oracle Technet (OTN) introduction to APEX IRs: http://www.oracle.com/technetwork/testcontent/irrs-083031.html

In addition, many developers have added customizations via CSS to improve appearance, dynamic actions to mimic MS Excel behavior, and otherwise extended standard IR functionality. Developers have made these customizations through dynamic actions, plugins, JavaScript frameworks and other bits of code to effect the desired changes. This paper focuses on the new IR architecture, and how the new structures may affect existing customizations.

APEX IR
As usual for a new APEX release, APEX 5 introduces new IR features: new and enhanced Action Menu features, some cosmetic uplifts and some report management improvements. All of these are covered in Part 1 of this series. With APEX 5, the major IR changes are behind the scenes – APEX IRs have been rebuilt from the inside out. Both the “inside” changes – the JavaScript engine – and the “outside” changes- CSS classes and Ids - are significant. The important note for developers is that because of this re-architecting, even when developers used the APEX-standard dynamic action or plugin frameworks, if the customization code references the pre-APEX id’s and class elements, the customization code will need to be refactored to upgrade.

Why so many changes, and why such a drastic change? Several reasons. The revised IR code allows for:

- Multiple IRs on one page (perhaps the biggest new feature, to be described in detail in later sections of this paper)
- Modal dialogs
- The Universal Theme and Theme Roller customizations
- APEX overall usability and accessibility

These features, particularly multiple IRs on a single page, were just not possible with the pre-APEX IR architecture. The new code makes sense, when one considers that multiple IRs on a single page was not possible with the legacy IR structure – something had to change. With all of the other IDE and end-user interface changes in APEX 5, the IR changes make sense.

The downside is, developers who have tooled outside of the APEX sandbox must now invest some time in upgrading to the new IR structure and JavaScript. All IR customizations made outside of the standard APIs may not work in APEX 5. Developers who have made such changes will need to refactor their customizations.

The following sections discuss the APEX 5 IR structural changes in detail. The Know Your Users section applies to all developers, and is repeated here as a reminder that all configuration and customization should have one goal, to serve user requirements. The Changes section addresses the CSS and JavaScript changes such developers need to know to plan their upgrade to APEX 5.

Know Your Users
IRs are very powerful in that they deliver a lot of end user functionality with minimal developer effort. However, it is the developer’s responsibility to maximize IR effectiveness by using the declarative settings to tailor the IR to end user needs. This means the developer needs to be aware of such things as overall security needs, how user use the data set, which Action Menu features should be prohibited or restricted, how much training end users will require, which download options are required, and how users are likely to use Saved reports. It is the developer’s job to prepare and
deliver the appropriate IR query and action set to support them. If customizations beyond standard IR features are required, they should first be implemented using standard APEX APIs, and only as a last resort achieved through custom coding.

In short, **Know Your Users.** Watch what they do, because what they really do is not necessarily what they say the do or need. Deliver the functionality they need, restrict the features they should not have, and ensure the data set they receive is useful to them.

**Changes**

*Turn and face the strange ch ch ch changes …*

D. Bowie

APEX IRs have been reengineered to accommodate multiple interactive reports on one page, and in keeping with the APEX5 overall style, usability and accessibility improvements. The underlying architecture is quite different than previous versions. It is important the developers understand these changes, particularly when adding dynamic actions, plugins or any other customizations.

**CSS Changes**

The main CSS changes are in a different pattern of class and id nomenclature. The old apexir_<element> id constructs are gone, replaced by a series of a-IRR-<element> classes and revised id names. Figure 1 shows the pre-APEX 5 IR HTML and CSS structure. Note the apexir_<element> id and class names. Figure 2 shows the APEX IR structure. Note the new STATIC_ID_<element> id name, and the a-irr-<element> class names.

The APEX 5 pattern is readily visible: apexir_<element> ids are now renamed to STATIC_ID_<element>, where STATIC_ID is the static id of the IR region, whether it was declared by the developer or assigned by APEX, in which case it has the format R123456789012345. It is much easier to understand the underlying ID structure, especially when there are multiple IRs on the page, when declared static ids that make sense are used. There is a Static Id attribute on every IR – in earlier versions it was most often left blank. With APEX 5, the Static Id is important in passing filters to specific IRs, and in the structure of the IR itself.

**JavaScript Changes**

Pre-APEX 5, the APEX IR JavaScript is contained in the file widget.interactiveReport.js. Post APEX-5, the APEX IR JavaScript is contained in the file widget.interactiveReport.js. The similarity ends there… well, almost. Reviewing the two files, one recognizes the same functions – actually widget methods - that correspond to the action menu actions, but the construction of the functions – the definition of the widget methods - is different, as is their implementation.

**The Supported Way**

The SUPPORTED way to influence APEX IRs is to “use the standard APIs”. That includes apexrefresh, and the APEX_IR API. Let’s see what we can do with these.

 apexrefresh is the documented, supported way to refresh APEX components from JavaScript. The syntax is simple:

```javascript
APEX_IR.RESET_REPORT("#myIRRegionStaticID", apexrefresh);
```

For more specific IR settings, use the APEX_IR API (https://docs.oracle.com/database/121/AEAPI/apex_ir.htm for APEX 5). To user most calls of APEX_IR, one needs to get the region id of your interactive report. As simple example, the code sequence to programmatically reset an IR to its default configuration uses the APEX_IR.RESET_REPORT procedure: is:

```sql
DECLARE
  v_region_id APEX_APPLICATION_PAGE_REGIONS.REGION_ID%TYPE;
BEGIN
  SELECT region_d INTO v_region_id
  FROM APEX_APPLICATION_PAGE_REGIONS
  WHERE application_id = :APP_ID
  AND page_id = :APP_PAGE_ID
  AND static_ic = 'MY_IR_STATIC_ID';
  APEX_IR.RESET_REPORT( P_page_id => :APP_PAGE_ID,
                            P_region_id => v_region_id,
                            P_report_id => NULL);
END;
```

Of course such code would be improved before production use by wrapping in a procedure, adding error catching and validations as needed for your situation.

In addition to the APEX_IR possibilities, there always the declarative RIR and CIR settings, discussed in Part I of this series in the Summer 2015 RMOUG SQL> UPDATE.

So we can do all the basics using supported, documented means. But what about our more complex situations, where we
have already strayed into unsupported territory with IR enhancements in earlier APEX versions?

**The Unsupported Way**

In APEX 4.2 and earlier, when there is always only one IR on a page, the widget is attached to the gReport element – the IR – and there is always only one gReport element on the page. One can inspect the JavaScript on the page and readily see calls to widget functions.

For example, inspection of the Search Column icon, shown in Figure 3, clearly shows the onclick action is a call to gReport.dialog2(“SEARCH_COLUMN”).

Further inspection of the IR JavaScript and HTML reveals that gReport.dialog2(‘parameter’) is the widget method for opening the ‘parameter’ dialog window, where ‘parameter’ values correspond to the Action Menu options.

Inspection of the APEX 4.2 Go button on the toolbar shows us the gReport.search call, shown in Figure 22.

With APEX 4.2, because we can clearly see how these two widget methods, gReport.dialog2 and gReport.search are used, we are actually quite confident in using them in our customizations, even when we know their direct use is unsupported. This is jQuery in APEX - we trust that this stuff is solid.

Well, in APEX 5, the familiar gReport.dialog2(...) and gReport.search(...) functions calls are not there. In fact, the familiar gReport element is not there at all. Figure 5 shows the IR Toolbar Go button HTML in APEX 5. No gReport in site – no jQuery is visible here.

![Figure 3 – APEX 4.2 gReport.dialog2 Call on Search Column Icon](image)

![Figure 4 – APEX 4.2 IR Toolbar Go Button gReport.search Call](image)

![Figure 5 – APEX 5 IR Toolbar Go Button – No gReport.search!](image)

Note again that all out-of-the-box APEX IRs will automatically use the new jQuery widgets, seamlessly. The problem is, in APEX 5, all calls to the undocumented gReport stuff do not work. Any customization that makes gReport method calls, or extends the interactive report widget will need to be refactored.

So how do things match up? What replaces gReport? Let’s go back to some jQuery UI widget basics. The APEX team has used the jQuery widget factory to build (rebuild) the interactive report and action menu widgets (and many others within APEX 5).

The best reference I have found on jQuery UI widgets is on jQuery UI, How to Use the Widget Factory, at [https://learn.jquery.com/jQueryUI/widget-factory/how-to-use-the-widget-factory/](https://learn.jquery.com/jQueryUI/widget-factory/how-to-use-the-widget-factory/). If you are not familiar with jQuery widgets and the jQuery widget factory, this how-to article will help understand jQuery widgets in general, which will help you understand how the APEX interactive report and action menu widgets work.

Note: If you are not familiar with jQuery and jQuery UI widgets, then seriously consider NOT making any unsupported customizations to the APEX IR widgets, at least until you learn more and become entirely confident in your ability to support yourself and your code. This paper does not contain sufficient information to make your first customization attempt.

The best reference for explaining how the APEX 5 IR widget works is by John Snyder, at [http://hardlikesoftware.com/weblog/2015/05/12/apex-5-0-interactive-report-customization/](http://hardlikesoftware.com/weblog/2015/05/12/apex-5-0-interactive-report-customization/).

Armed with the information in these two articles, we can figure out our APEX 5 IR widget code.

Widgets are attached to DOM elements. Widget methods are the functions that define what the widget does – the actions. Methods prefixed with _ are private methods. Every widget has an options method that lists the options (attributes). To view all options, use the option keyword:

\$\text{(selector).widgetName(“option”)};

To view a specific option, simply state the option:

\$\text{(selector).widgetName(“option”, “option-name”)};

Adding a second parameter sets the value of the option:

\$\text{(selector).widgetName(“option”, “option-name”, 100)};

The basic call to a widget method takes the format

\$\text{(selector).widgetName(“method”)};

If the widget method has parameters, add the parameters after the method name:

\$\text{(selector).widgetName(“method”, “param_1”, “param_2”)};

Now let’s apply that context to our APEX IR.

The key information John gives us is that in APEX 5, the IR widget appends “_ir” to the static id of the IR, and “_actions_menu” to the actions menu widget. So if my IR has a static id of DEMO_IR, we know the ids for the IR widget and for the IR actions menu. To view all the options for these widgets, in the Console window use these commands:

\$\text{($\text{#DEMO_IR_ir}).interactiveReport(“option”)};

to show all the IR widget options, and

\$\text{($\text{#DEMO_IR_actions_menu}).menu(“option”)};

to show all the action menu options.

There are quite a few options for both – to learn more about the IR and action menu widgets, definitely run these jQuery commands in your browser Developer Tools Console window. View the options,
then experiment with calling some of the options. The IR widget options correspond to the IR attributes. The menu widget options correspond to the IR action menu options, and the items correspond to the menu actions. These will be familiar to developers who are familiar with IR settings.

The IR widget options include: actionsMenu, afterRefresh, aggregate, chart, columnSearch, compute, controlBreak, fixedHeader, flashback, groupBy, help, highlight, and more. Those familiar with IR action menu settings will recognize these as IR attributes, including action menu settings.

The action menu options include 15 items, with id’s, like irSaveReport, irSaveDefault, irReset, irDownload and irNotify. (There are more – inspect for yourself!)

Following John’s blog, and experimenting with various options in the Console, I was able to find these interesting things:

```javascript
$("#DEMO_IR_ir").interactiveReport("option","currentRowsPerPage");
returns the current rows per page setting.
$("#DEMO_IR_ir").interactiveReport("refresh");
refreshes the IR.
$("#DEMO_IR_actions_menu").menu("find","irDownload");
returns the irDownload object.
$("#DEMO_IR_actions_menu").menu("find","irDownload").action();
invokes the action method of the irDownload object – it opens
the IR Download dialog. Translated, it finds the irDownload object
of the DEMO_IR interactive report actions menu and calls its action
method.
I leave the rest to your own experimentation. Clearly we are
not totally in the dark here, once we know how the widgets work.
Note: The IR static id is optionally declared by the developer.
If a static id is not declared, APEX assigns a long ugly one – therefore, it is best to assign a static id when one plans on
using referencing it in JavaScript.

At the 2015 ODTUG Kaleidoscope conference, we heard from
the APEX Team that there will be a forthcoming JavaScript API
for IRs. At this time, it is unclear when and how extensive an API
will be provided to standardize and facilitate calls to the IR func-
tions – particularly those for Reset, pagination, and Search options.
Until then, remember that customizations that do not use standard,
documented APEX APIs are not supported.

As in earlier APEX versions, developers who customize the
APEX IR JavaScript widget are in unsupported territory. Which
does not mean altering the IR widget or using its calls directly
cannot be done, it means that doing so is not supported, and will
likely not upgrade smoothly or at all to future versions. (Sound
familiar?).

Note: In earlier APEX versions, there was less declarative
JavaScript capability and therefore more developer customiza-
tions. As APEX advances and incorporates more declarative
JavaScript, it is more important to stay within the standard
APIs when making customizations, to avoid difficulties when
upgrading.

A few code comparisons will illustrate further emphasize the
differences between the APEX 4.2 widget and the APEX 5 widgets.

The pre-APEX 5 RESET function is

```javascript
/**
 * Reset current worksheet report to initial state
 * @function
 * */
this.reset = function() {
  that.action('RESET', false, false, false);
};

The APEX 5 RESET function is a private method:

```javascript
/**
 * Reset current worksheet report to initial state
 * @function
 * */
_reset: function() {
  this._action( "RESET" );
},
```

Not exactly the same, but close, and note the APEX 5 private
method.

The pre-APEX 5 Search function is:

```javascript
/**
 * Runs the basic search functionality of the worksheet.
 * @param {String} [pThis] if set to SEARCH check
 * @param {Number} [pRows]
 * */
this.search = function(pThis, pRows) {
  var lSearch = that.item.search();
  var lSearch_Col = that.item.search_column();
  var lReport = $v('apexir_REPORT_ID');
  var lTemp;
  if (pThis='SEARCH') {
    if (pRows) {
      that.get.addParam('p_widget_num_return', pRows);
    } else {
      if ($x('apexir_NUM_ROWS')) {
        that.get.addParam('p_widget_num_return', $v('apexir_NUM_ROWS'));
      }
    }
  } else {
    lTemp = [this.current_col_id, '=' $v(lSearch)];
    pThis = 'FILTER';
    that.get.AddArray(lTemp,1);
  }
  that.action(pThis, 'ADD');
}
```

... then experiment with calling some of the options. The IR widget options correspond to the IR attributes. The menu widget options correspond to the IR action menu options, and the items correspond to the menu actions. These will be familiar to developers who are familiar with IR settings.

The IR widget options include: actionsMenu, afterRefresh, aggregate, chart, columnSearch, compute, controlBreak, fixedHeader, flashback, groupBy, help, highlight, and more. Those familiar with IR action menu settings will recognize these as IR attributes, including action menu settings.

The action menu options include 15 items, with id’s, like irSaveReport, irSaveDefault, irReset, irDownload and irNotify. (There are more – inspect for yourself!)

Following John’s blog, and experimenting with various options in the Console, I was able to find these interesting things:

```javascript
$("#DEMO_IR_ir").interactiveReport("option","currentRowsPerPage");
returns the current rows per page setting.
$("#DEMO_IR_ir").interactiveReport("refresh");
 refreshes the IR.
$("#DEMO_IR_actions_menu").menu("find","irDownload");
returns the irDownload object.
$("#DEMO_IR_actions_menu").menu("find","irDownload").action();
invokes the action method of the irDownload object – it opens
the IR Download dialog. Translated, it finds the irDownload object
of the DEMO_IR interactive report actions menu and calls its action
method.
I leave the rest to your own experimentation. Clearly we are
not totally in the dark here, once we know how the widgets work.
Note: The IR static id is optionally declared by the developer.
If a static id is not declared, APEX assigns a long ugly one – therefore, it is best to assign a static id when one plans on
using referencing it in JavaScript.

At the 2015 ODTUG Kaleidoscope conference, we heard from
the APEX Team that there will be a forthcoming JavaScript API
for IRs. At this time, it is unclear when and how extensive an API
will be provided to standardize and facilitate calls to the IR func-
tions – particularly those for Reset, pagination, and Search options.
Until then, remember that customizations that do not use standard,
documented APEX APIs are not supported.

As in earlier APEX versions, developers who customize the
APEX IR JavaScript widget are in unsupported territory. Which
does not mean altering the IR widget or using its calls directly
cannot be done, it means that doing so is not supported, and will
likely not upgrade smoothly or at all to future versions. (Sound
familiar?).

Note: In earlier APEX versions, there was less declarative
JavaScript capability and therefore more developer customiza-
tions. As APEX advances and incorporates more declarative
JavaScript, it is more important to stay within the standard
APIs when making customizations, to avoid difficulties when
upgrading.

A few code comparisons will illustrate further emphasize the
differences between the APEX 4.2 widget and the APEX 5 widgets.

The pre-APEX 5 RESET function is

```javascript
/**
 * Reset current worksheet report to initial state
 * @function
 * */
this.reset = function() {
  that.action('RESET', false, false, false);
};

The APEX 5 RESET function is a private method:

```javascript
/**
 * Reset current worksheet report to initial state
 * @function
 * */
_reset: function() {
  this._action( "RESET" );
},
```

Not exactly the same, but close, and note the APEX 5 private
method.

The pre-APEX 5 Search function is:

```javascript
/**
 * Runs the basic search functionality of the worksheet.
 * @param {String} [pThis] if set to SEARCH check
 * @param {Number} [pRows]
 * */
this.search = function(pThis, pRows) {
  var lSearch = that.item.search();
  var lSearch_Col = that.item.search_column();
  var lReport = $v('apexir_REPORT_ID');
  var lTemp;
  if (pThis='SEARCH') {
    if (pRows) {
      that.get.addParam('p_widget_num_return', pRows);
    } else {
      if ($x('apexir_NUM_ROWS')) {
        that.get.addParam('p_widget_num_return', $v('apexir_NUM_ROWS'));
      }
    }
  } else {
    lTemp = [this.current_col_id, '=' $v(lSearch)];
    pThis = 'FILTER';
    that.get.AddArray(lTemp,1);
  }
  that.action(pThis, 'ADD');
}
```
The APEX 5 SEARCH function is a private method:

```javascript
/**
 * Runs the basic search functionality of the worksheet.
 * @param {String} [pThis] if set to SEARCH check
 * @param {Number} [pRows] Optionally set to control the number of rows displayed, needs to be done with the search because the user could enter a new search, then select the rows which would issue the search
 */
_search: function( pThis, pRows ) {
  var lData, lFArrays,
      o = this.options,
      lSearch = this._getElement( "search_field" ).val();
  // If pRows passed, this has been changed and the new value used, but only allow if either actions menu
  // row select, or search bar row select is enabled
  if ( pRows && ( o.rowsPerPage || o.rowsPerPageSelect ) ) {
    o.currentRowsPerPage = pRows * 1;
  }
  lFArrays = this._utilGetFormElAttributes( this._getId( "toolbar_controls" ) );
  lData = {
    f01: lFArrays.ids,
    f02: lFArrays.values
  };
  // If there is a search and a new search is entered, use the new one
  if ( lSearch !== "" ) {
    this._pull( null, this.reportId, lData );
  } else {
    this._action( "QUICK_FILTER", lData );
  }
},
```

As one may have guessed, the APEX 5 _action function – a private method – is a wrapper for an AJAX get:

```javascript
/**
 * The basic AJAX call for ACTIONS for the IR, just a wrapper around _get
 * @function _action
 */
_action: function( pAction, pData ) {
  var lData = $.extend( { widgetMod: "ACTION", widgetAction: pAction }, pData );
  this.currentAction = pAction;
  this._get( lData );
},
```

The APEX IR interactiveReport widget _get function controls all the functions of the interactive report. The complete widget, interactiveReport.js?v=5.0.0.00.NN can be inspected from its location:<host server and port>/i/libraries/apex/widget.interactiveReport.js. Where the <host server and port> are the http/https and host server name and port of your APEX installation, and NN is the exact version number in your APEX 5 installation's widget, interactiveReport.js library.

The above code excerpts are not meant to be any sort of how-to – their purpose is only to illustrate that there are differences and any developer who has pre-APEX 5 customizations that rely on the pre-APEX 5 form of widgetinteractiveReport.js will need to review their code carefully and refactor when upgrading to APEX 5.

### gReport to Widget Method?

So, what to do with all those unsupported gReport calls? Refactor to use the APEX 5 widget calls and remain unsupported? I cannot tell you. Wait for the APEX IR JavaScript API (no telling how long that will be)? Perhaps more will be revealed in APEX 5.1?

### To Risk or Not to Risk?

Comparison of the main functions of the pre-APEX 5 and APEX 5 interactive report JavaScript reveals that a similar, but not exact *private* function or widget method exists for each of the APEX IR actions. Developers that have customizations that rely on the old IR JavaScript must review each APEX5 function carefully and decide how to update their code accordingly.

Again, it is not expected that customized use of the IR JavaScript functions will be supported. Developer beware...

The recommendation is to use dynamic actions that call the APEX IR API functions and procedures to achieve the desired results. This method may mean more coding, but this approach, using the APEX dynamic actions and the published APIs – will be supported going forward. Direct use of the IR and action menu widget methods is not supported. If you go this way, be prepared to refactor (again) going forward.

### Upgrade (refactor) Example

The following simple example demonstrates the changes needed in an existing dynamic action to work successfully in APEX 5. The refactor in this case was required due to the CSS changes in APEX 5 IRs. The old references needed to be updated to the APEX5 equivalents.

This simple Execute JavaScript dynamic action changes the background color of a row based on a certain value in the FLAGS column. Yes, this simple highlight could have been done with a preset IR Highlight action, however, to not clutter the control panel, to prevent users from editing this effect and for other business reasons this appearance change was done via JavaScript. (This dynamic action was one of several on the IR, each of sufficient complexity that it was not practical to implement all of them as IR actions).

```javascript
var rows = $('table.apexir_WORKSHEET_DATA tbody tr:gt(0)');
rows.each(function(idx) {
  var Flags = $(this).children("td[headers='FLAGS']").text();
  if( Flags == 'Outlier') {
    $(this).children("td").css("background-color","#FCF067");
  }
});
```

Simple enough, but this does not upgrade to APEX 5 because the apexir_WORKSHEET_DATA id no longer exists, so the table.
APEX 5 IR Upgrade Cheat Sheet

As promised, this paper includes a “cheat sheet” for mapping pre-APEX 5 CSS and IDs to their APEX 5 equivalents. The cheat sheet is included in the appendix. Note that a cheat sheet is just a guide – it does not replace doing your homework and learning your subject matter. If you have read the above sections, you have all the information you need to map out your refactoring – you won’t need a cheat sheet.

Every developer should inspect their code and their particular interactive report in APEX 5 and validate their own mappings. The reason is, each interactive report may be configured differently. Different configurations will incur different combinations of IR class and id settings. This author’s recommendation is to use the enclosed cheat sheet as a guideline in beginning your upgrade work, but always always inspect and double-check the mappings in your specific IR and in your specific APEX 5 environment.

Summary

APEX 5 IRs contain some noteworthy new features for end users and developers. For end users, the most significant of those are the improved look-and-feel, modal dialogs, the enhanced GROUP BY action, the new PIVOT action, and the option for multiple IRs on a single page. Part 1 of this series covered the APEX 5 new and enhanced features for end users. For developers, the APEX 5 IR construction changes are more significant as the underlying architecture has completely changed. Developers who have done customizations on IRs in previous APEX version may need to refactor their code to follow the new APEX 5 IR structure and widget methods. With APEX 5, the overall power of APEX IRs has increased, but when it comes to customizations, some upgrade work may be necessary.

References

Oracle OTN APEX Pages
The APEX OTN Forum
http://forums.oracle.com/forums/forum.jspa?forumID=137
Manually Refreshing APEX Components
http://docs.oracle.com/cd/E59726_01/doc.50/e39147/extend_app002.htm#HTMDB30267
APEX 5.0 Interactive Report Customization, John Snyders
http://hardlikesoftware.com/weblog/2015/05/12/apex-5-0-interactive-report-customization/
jQuery UI: How to Use the Widget Factory
https://learn.jquery.com/jquery-ui/widget-factory/how-to-use-the-widget-factory/
The APEX 5 APEX_IR API Documentation
https://docs.oracle.com/database/121/AEAPI/apex_ir.htm

Appendices

APEX 5 completely overhauls APEX Interactive Reports from the inside out. The CSS, class structure and JavaScript have all changed significantly. Developers who have made Interactive Report customizations that use pre-APEX 5 CSS, classes and JavaScript and do not use the standard APEX APIs will need to refactor their code to follow the new APEX 5 IR structure and widget methods. APEX 5 automatically renders all Interactive Reports using the new CSS, class structure and stylesheet. THERE IS NO OPT-OUT, and there is no upgrade action required by the developer. Fortunately, to those familiar with the legacy IR CSS, class and JavaScript, the new elements will be easy to follow.

The following tables are for comparison and guideline use only. Not all elements have equivalents pre-APEX 5 and APEX 5.

The exact IDs, classes and JavaScript functions used will depend on the complete settings and current state of your Interactive Report.

In General

apexir_ IDs are replaced by corresponding STATIC_ID_<element> IDs where STATIC_ID is the declared or defaulted Static Id of the IR Region. If the developer does not declare a Static Id, APEX assigns one.

apex_ classes are replaced by corresponding a-IRR-<element> classes

Table A1 – APEX 5 Interactive Report Upgrade CSS Cheat Sheet is available on RMUOG’s web page. Please see www.rmoug.org/rmoug-sqlupdate-newsletters-online/
RMUOG’s kickoff of the Raspberry Pi Meetup occurred at the Summer Quarterly Education Workshop at Elitch’s on August 7th 2015. The event garnered more than enough interest for a monthly meetup to be formed for a new special interest group and the family coding event was a real hit with all the attendees.

Now if you’re not familiar with the Raspberry Pi, it’s a small, very affordable micro-computer that allows the user to build out programs and runs on an ARM processor. I have two in my possession now, the first release and a B+ version that has more processing power, memory and USB ports. The Operating System is commonly a variant of Unix, called Raspian, a version of Debian from the main vendor, but you can even load a new Windows 10 version that was just released last month, (if that’s your thing!)

There are a number of additional boards, accessories, kits and programming languages that are available from different vendors, but I prefer to work with Python, a great little programming language. The following programs that will be discussed in this article are all in Python3, the latest version, but will work with Python2, the previous release, too.

**STEM and Raspberry Pi**

One of the main reasons I’ve become so engrossed in this little computer is the opportunity to introduce tech education to the younger generation. We have an estimated 1.4 million tech jobs that will be opening up by 2020 and one about 20,000 CS graduates in the US each year. If you start doing the math, you can quickly ascertain how few individuals we have in the states to fulfill these positions and why it’s essential that more kids become aware of the opportunities in technology. I have three kids that either have recently graduated or will be soon and I can verify they aren’t receiving the information about these opportunities in public school. Most career recommendation testing is still based off a workforce estimate from two decades ago. There are students being pushed towards history and literature majors that will leave college without any jobs, which considering the cost of secondary education these days, seems quite a waste when there are so many tech jobs out there waiting.

**The Raspberry Pi Without the Hardware**

If you still haven’t bought one of these great little units and would like to work on some of this code in the article you’re reading, there is still another option to get you hooked. There’s a blog post on how you can download a Virtualbox image and set it up to do everything that I cover in this article out on my DBAKevar site. Just go to the following URL for my blog: [http://dbakevar.com/2015/08/emulating-a-raspberry-pi-on-virtualbox/](http://dbakevar.com/2015/08/emulating-a-raspberry-pi-on-virtualbox/) and the instructions will assist you in setting it up. You’ll need to have Oracle Virtualbox already installed on your PC, but most Database Administrators and Developers already have it. If you don’t, Virtualbox is easy to download and set up from Oracle Technology Network’s download page.

**The Raspberry Pi Project**

So how can this little computer engage the younger generation? Science projects, robotics, motorized art projects and for today, we’re going to build out a couple of games. Well known dice games to be precise.
The ability to simulate a dice throw is engaging to any kid or teenager and to be able to do this with code is pretty cool. Our first code today will have the following requirements:

1. A player will roll the dice as many times as it takes to get a 7, an 11 or roll a double.
2. The scoring will be the number of rolls subtracted from the final number they rolled for their 7, 11 or doubles.
3. The game must display the rolls previous to the 7, 11, or double, as these runs must be subtracted from the total.
4. At the roll of a seven, eleven or doubles, the code will exit, as the player's turn will be over.

**Coding in Python**

To create this script, we’re going to use the RANDOM module, which comes built in with Python2 and Python3. Python3 is now the expected version to use and even though 2 is still supported, it is recommended that you code in Python3. You can tell what version of Python you’re using by typing the following at a terminal prompt:

```python
python --version
```

We’re also going to write all our code in a way that is referred to as Pythonic thinking, which is essential to Python scripting and a bit different than Java, Ruby, C++ or Shell/Perl. Pythonic thinking is not a set of finite rules that must be followed, but a set of standards and practices that were designed by those that have worked with the product since its conception and have documented the best way to work with the language.

The code for this program is very simple due to the module and looks like this:

```python
import random #Use the RANDOM module

for x in range(1, 31): #We'll throw up to 30 times to get to 7, 11 or doubles.
    dice_1 = random.randint(1, 6) #Our Dice has 1-6
    dice_2 = random.randint(1, 6) #Second Dice 1-6
    dice_throw = dice_1 + dice_2 #Our total for our throw is two dice
    print(dice_throw) #Print the total so we can see the value
    if dice_throw == 7: #Seven is a winner
        print('Seven!')
        break
    if dice_throw == 11: #Eleven is a winner
        print('Eleven!')
        break
    if dice_1 == dice_2: #Doubles are a winner
        print('Doubles Thrown!!')
        break #Break out of the dice throw if 7,11 or doubles thrown
```

**Playing the Game**

What the output during the play looks like is the following:

```
pi@joshpi ~/Downloads $ sudo python3 dice_gm.py
7
```

I hope the totals at the bottom make sense on how the players would then calculate for each “throw”, i.e. execution. For our game, we are going the winner of the game will be whoever makes it to a score of 100. With each player using a Raspberry Pi and the script running on it, they can just take turns executing and keeping score (hint: enhancement to the game!).

The next phase of this project is to use the current script as a base and then create more dice games from ones you already play at home with physical dice. (I don’t understand this previous sentence?)

**Pass the Pigs**

Oh, you know you want to play this in a coded game! Can we do it? Of course we can! Let’s start with the rules for playing the actual game. (See Figure on page 18)

**Coding in Pigs**

Now we need to code our dice to return as the pigs would. This is where your logical thinking skills and imagination come into play. For my example, I’m going to again use the RANDOM module that we used in our earlier coding example, but the requirements for this new game is quite different from the first.

This first script is pretty complex for one simple reason - we decided to keep all the combinations for scoring in the code. I want you to look at the combinations closely and see how this ends up being a real headache to keep track of. Is this the best way to design this?
import random

for x in range(1, 31):
dice_1 = random.randint(1, 6) #options
for pigout
    dice_2 = random.randint(1, 6) #options
for pigout
    print(dice_1) #Print each value
    if dice_1 <= 2 and dice_2 <= 2 and dice_1
        print('Razorback and Trotter
            print('Razorback and Trotter, 10
        break
    if dice_1 or dice_2 <= 2 and dice_1 or
dice_2 == 3: #T/R with Snouter
        print('Trotter or Razonback with
        print('Trotter or Snouter, 15 pts!
        break
    if dice_1 or dice_2 <= 2 and dice_1 or
dice_2 == 4: #T/R with Learning Jowl
        print('Trotter or Razorback with
        print('Trotter with Learning Jowl, 20pts!
        break
    if dice_1 or dice_2 == 3 and dice_1 or
dice_2 == 4: #Snouter
        print('Snouter with Learning Jowl, 35PTS!!!Woohoo!
        break
    if dice_1 or dice_2 == 5: #Pig Out
        print('Pig out Back to ZERO, Next
        print('Pig Out Back to ZERO, Next Players turn!
        break
    if dice_1 or dice_2 == 6: #Oinker
        print('Oinker, Back to ZERO, Next
        print('Oinker, Back to ZERO, Next Players turn!
        break
    if dice_1 or dice_2 == 1 and dice_2 ==
dice_1:
        print('Doubles- Razorback- 20pts!
        break
    if dice_1 or dice_2 == 2 and dice_2 ==
dice_1:
        print('Doubles- Trotter- 20pts!
        break
    if dice_1 or dice_2 == 3 and dice_2 ==
dice_1:
        print('Doubles- Snouter- 40pts!
        break
    if dice_1 or dice_2 == 4 and dice_2 ==
dice_1:
        print('Doubles- Pig Out- Oink, OINK! 1pt!
        break
    if dice_1 <= 3 and dice_2 <= 3 and dice_1
        print('Doubles- Mixed, Add up indi
        print('Doubles- Mixed, Add up indi
    break

Now this code will emulate what the game rolls of the Piggy's would! You can execute the code just as you would any python script from a Raspberry Pi Terminal screen:

> sudo python3 <script_name>.py

3
1 Trotter or Razonback with Snouter, 15 pts!

Clearing the Fat

So let's say we want to simplify this and “Pythonic” this script even more. How would we go about that? We can still use the RANDOM option, but this time we'll use it with the "pig falls" directly and we'll use a function to build out our script instead of all the if statements.

import random

pig_fall = ['Razorback', 'Trotter', 'Snouter', 'Leaning Jowl', 'Pig Out', 'Oinker']

def pick_pigs():
Pig1 = random.randint(0, len(pig_fall) -1)
    Pig2 = random.randint(0, len(pig_fall) -1)
return pig_fall[Pig1] + ' + ' + pig_fall[Pig2]

print (pick_pigs())

Notice I've removed all the logic for scoring and figuring out what are doubles, etc. We've simplified and instead of converting the numbers of the dice throws into the pig landings, we're just going to randomize the pig landings exclusively. I've gone back to a very simple script that allows me to play the game without adding any complexity into the code that isn't in the original "manual" game. Just random "pig throws" displaying the landings that were made!

> sudo python3 pignew.py

Leaning Jowl + Snouter. Woohoo, 35 pts!
Trotter + Pig Out Bummer...
Leaning Jowl + Trotter Not bad!

You now have three examples of game code that you can use, enhance or build out to create your own games. There are a ton of other projects, add-ons and events that you can do with this great micro-computer, so don't miss out.

If you're interested in learning more about the RMOUG Raspberry Pi Meetup and Special Interest Group, you can find everything you need to know on my blog, (and soon on the RMOUG website!) at http://dbakevlar.com/raspberry-pi-page/
How to Disable Database Triggers
For a Single Session

by Tom Peters

The Problem:
Database triggers can come in very handy and be used to solve a myriad of problems. However, at times, their effects may not be desired. As an example, say there is a before-update row-level trigger that prevents updates to a row when its status code equals ‘CLOSED’. We all know that sooner or later, a customer will need to change a row that has a status of ‘CLOSED’. Of course the trigger can be disabled prior to running the update script, such as:

```
begin
  execute immediate 'alter trigger bu_my_table disable';
  update my_table
  set x = 32
  where x = 3.2
  and status_code = 'CLOSED'
  and create_date = '14-jul-2014';
  execute immediate 'alter trigger bu_my_table enable';
end;
```

The problem with this approach is that the trigger logic is absent for updates occurring in other sessions as well. This may be problematic, or depending on the logic that gets skipped, catastrophic. It could help to just run your script when the likelihood of other updates occurring is lower. While that approach may reduce risk, it’s still far less than ideal.

There is a better, safer way! The remainder of this article will explain how you can effectively disable triggers for just the single session used by the script.

The Solution:
The solution takes advantage of the fact that package variables are stored in the user global area (UGA). Data stored in the UGA is specific to a user’s session and can only be seen by programs the run in that same session. Therefore, if an update script running in session XYZ stores a value in a package variable, then that variable’s value can be interrogated by a trigger also running in session XYZ to determine whether to run the logic in the trigger. The process is described in detail below.

1. You have determined that a script needs to skip trigger logic while executing DML.
2. The script sets a package variable to a prescribed value prior to executing DML.
3. The script executes the DML, which causes the trigger to fire.
4. Before the trigger logic executes, the trigger inspects the package variable for the presence of the prescribed value.
5. If the value is found in the package variable, the logic is skipped, otherwise the logic is executed.

To implement these steps, do the following:
1. Create a package with a PL/SQL table indexed by varchar2 to represent a list of programs.
2. Add two procedures to the package; enable and disable.
   - disable adds a value to list represented by the PL/SQL table
   - enable removes a value from the list represented by the PL/SQL table
3. Add a function called is_enabled to the package.
   - is_enabled checks for the presence of a value in the list of programs
4. Alter your trigger source code to call is_enabled.
5. Alter any future scripts that need to skip trigger logic by calling disable before executing DML, and then call enable afterward.

The source code for step 1 is below.

```
CREATE OR REPLACE PACKAGE program_control_pkg IS
  PROCEDURE disable (in_program VARCHAR2);
  PROCEDURE enable (in_program VARCHAR2);
  FUNCTION is_enabled (in_program VARCHAR2) RETURN BOOLEAN;
END; -- Package spec
```

```
CREATE OR REPLACE PACKAGE BODY program_control_pkg IS
  type program_tab_typ is table of varchar2(1)
  index by varchar2(4000);
  program_tab program_tab_typ;
  procedure disable( in_program varchar2 ) is begin
    program_tab( in_program ) := 'Y';
    end disable;
  procedure enable( in_program varchar2 ) is begin
    program_tab.delete( in_program );
```

cont’d on page 31
RMAN In 12c: The Next Generation

by Rene Antunez

The new generation of Oracle’s popular database has been designed for the Cloud and will enable you to make more efficient use of IT resources while continuing to improve your users’ service levels. These enhancements and new features will make it easier for customers to take advantage of the Cloud.

This 3 parts series article will discuss several of these new features specific to RMAN, Oracle’s backup and recovery utility.

The first thing that we have understand in Oracle 12c a new concept of Pluggable and Container databases.

What is a Container and a Pluggable Database?

A container database (CDB) is an Oracle database that includes zero, one, or many customer-created Containers or Pluggable Databases.

The CDB has:
- One ROOT container (CDB$ROOT) containing SYSTEM, SYSAUX, UNDO, and TEMP tablespaces, Controlfiles and Redologs.
- One SEED container (PDB$SEED) containing SYSTEM, SYSAUX, TEMP, and EXAMPLE tablespaces (used as a template to create new PDBs).

A pluggable Database (PDB) is a user-created container holding the data and code for your specific applications. A PDB:
- Has SYSTEM, SYSAUX, and TEMP tablespaces.
- Contains any number of other user created tablespaces.
- Writes to the container UNDO tablespace, controlfiles and redologs.
- Undo and redo are annotated with details of the PDB they belong to.

In the Oracle Multitenant architecture, the server parameter file still exists and it belongs to the CDB, though there is a new concept between a CDB and a PDB, called inheritance and it means that the value of a particular parameter in the root (CDB) is inherited by one or more PDBs. There are parameters that can be changed at the PDB level and override what is being inherited from the CDB.

Now we have to understand what RMAN is and how RMAN works. Recovery Manager, better known as RMAN, is an Oracle client utility that is installed with the Enterprise or Standard edition. RMAN can also be found using the Admin option when installing the Oracle Client.

Because it is a client side utility, it allows you to use one RMAN executable version to backup current and previous versions of the Oracle Database. There are some restrictions, which can be verified in MOS document RMAN Compatibility Matrix [ID 73431.1].

This RMAN executable, which is a “wrapper” that presents a command-line interface (CLI) for the DBMS_RCVMAN and DBMS_BACKUP_RESTORE packages present in a library file called recover.bsq. The executable uses this file to extract code from it to create PL/SQL calls to the TARGET database. This library file is located in $ORACLE_HOME/rdbms/admin. Basically what the executable does, is to interpret the commands you give it through the CLI, direct server sessions to execute those commands, and record its activity in the TARGET database control file that is being backed up. Other Oracle programs, such as Oracle Enterprise Manager, do not call the RMAN executable, but instead call the DBMS_RCVMAN and DBMS_BACKUP_RESTORE packages and functions directly.

There are two main SYS packages that do the work of backup and recovery, which are
- DBMS_RCVMAN (procedures which list your database incarnations, the set until time recovery window, list your backups, to name a few), and
- DBMS_BACKUP_RESTORE (which as you might have guessed is the one who does the backup and recovery operations, like create the control file snapshot, backup the datafiles, backup the spfile to name some).

The RMAN client directs the server sessions to execute the commands through channels. A channel represents one stream of data to a device, and corresponds to one database server session. The channel reads data into PGA memory, processes it, and writes it to the output device.

The work of each channel, whether of type disk or System Backup Tape (SBT), is subdivided into the following distinct phases:
- Read Phase
  A channel reads blocks from disk into input I/O buffers. The allocation of these buffers depends on the number of datafiles being read simultaneously from disk and written to the same backup piece. One way to control the numbers of files is the backup parameter FILESPERSET.
- Copy Phase
  A channel copies blocks from input buffers to output buffers and performs additional processing on the blocks, like the validation of the data blocks, as it verifies that it’s not backing up corrupt data blocks. The Copy phase also does the binary compression and the backup encryption.
- Write Phase
  A channel writes the blocks from output buffers to storage media. The write phase can be either to SBT or to disk, and these are mutually exclusive, meaning you write to one or the other, not both.
As you can see by the phases above, and what distinguishes RMAN from any other method, is that the backup is at the block level. This brings great advantages to the older-style user managed backups as it doesn’t have to backup empty blocks.

What do you need to back up?

The files that are critical to backup in order to restore and recover your database in case a disaster were to occur:

- Data files
- Control files
- Archived Redo logs (If Database running in Archive Log mode)

Additional files needed in case of a full disaster where you have lost your database as well as your site or your server:

- Parameter File (pfile or spfile)
- Block Change Tracking File
- ORACLE_HOME/GRID_HOME
- tnsnames.ora / listener.ora / sqlnet.ora /../WALLETS/<$ORACLE_UNQNAME> (In case of using TDE)
- Any other file that you and your business deem important to your DR

These additional files are not critical for RMAN to back up, but are very important in case a full disaster were to occur. If you do not have a backup of these files, you would need to rebuild your ORACLE_HOME / GRID_HOME from scratch, taking more time to restore completely.

Oracle RMAN 12c Newest Features

A) SYSBACKUP Privilege

In Oracle 12c the SYSBACKUP privilege was introduced. This privilege allows a user to perform backup and recovery operations in RMAN. You can use this new privilege by simply connecting with the “as sysbackup”.

If you want to connect using a password, a password file needs to be created or in place with the administrative privilege of SYSBACKUP. The file will have to be in a 12.1.x format (which is the default in 12.1.0.1 and 12.1.0.2) and the parameter REMOTE_LOGIN_PASSWORDFILE needs to be set to exclusive. In the example below you will see how the password file is created.

<table>
<thead>
<tr>
<th>OS System Group Name</th>
<th>UNIX/Linux</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSDBA</td>
<td>dba</td>
<td>ORA_HOMENAME_DB (for each specific Oracle home)</td>
</tr>
<tr>
<td>OSBACKUPDBA</td>
<td>backupdba</td>
<td>ORA_HOMENAME_DB (for each specific Oracle home)</td>
</tr>
</tbody>
</table>

By using these OS system groups, you allow a user to run and administer your backups without having to grant the SYSDBA privilege, preventing that user from access to any SELECT/INSERT/UPDATE/DELETE ANY TABLE commands.

Having this privilege allows you to perform the following operations and enables you to connect to the database even if the database is not open. The SYSBACKUP privileges are limited to:

- STARTUP/ SHUTDOWN
- ALTER DATABASE/ SYSTEM/ SESSION/ TABLESPACE
- CREATE CONTROLFILE/ ANY DIRECTORY/ ANY TABLE/ PFILE/ SPPFILE / ANY CLUSTER/ SESSION
- CREATE RESTORE POINT (including GUARANTEED restore points)
- DROP DATABASE/ TABLESPACE/ RESTORE POINT (including GUARANTEED restore points)
- FLASHBACK DATABASE/ RESUMABLE
- UNLIMITED TABLESPACE
- SELECT ANY DICTIONARY/ ANY TRANSACTION
- SELECT X$ tables (that is, the fixed tables)
- V$ and GV$ views (that is, the dynamic performance views)
- APPQOSSYS.WLM_CLASSIFIER_PLAN
- SYSTEM.LOGSTDBYSPARAMETERS
- DELETE/INSERT
- SYS APPLY$_SOURCE_SCHEMA
- SYSTEM.LOGSTDBYSPARAMETERS
- EXECUTE
- SYS.DBMS_BACKUP_RESTORE
- SYS.DBMS_RCVMAN
- SYS.DBMS_DATAPIUMP
- SYS.DBMS_IR
- SYS.DBMS_PIPE

ALT SYSTEM
AUDIT ANY
SELECT ANY TRANSACTION
SELECT ANY DICTIONARY
RESUMABLE
CREATE ANY DIRECTORY
UNLIMITED TABLESPACE
ALTER TABLESPACE
ALTER SESSION
ALTER DATABASE
CREATE ANY TABLE
DROP TABLESPACE
CREATE ANY CLUSTER
13 rows selected
B) SUPPORT FOR MULTITENANT CONTAINER DATABASES AND PLUGGABLE DATABASES

The RMAN utility provides full backup and recovery support, not only for your container database, but also for one or all of your pluggable databases. With this in mind the first thing that you want to do is identify how your database is composed. You can do this with the `report schema` command. When you use this command at a container level, you can see that your pluggable database files are preceded with the name of the database. When you use it at the pluggable level, you will only see the files that belong to that particular database. The example below first connects at the container level (cdb1), and then connects at the pluggable level (pdb1).

```
[oracle@oracleenespanol2 admin]$ rman target sys/oracle@cdb1
connected to target database: CDB1 (DBID=808250731)
RMAN> report schema;
using target database control file instead of recovery catalog
Report of database schema for database with db_unique_name CDB1
List of Permanent Datafiles
===========================
File Size(MB) Tablespace        RB segs
Datafile Name
--- -------- -------------------- ------- --------
1   790    SYSTEM               ***     /
2   880    SYSAUX               ***     /
3   125    UNDOTBS1             ***     /
4   560    SYSTEM               ***     /
5   590    SYSAUX               ***     /
6   260    USERS                ***     /
7   260    PDB$SEED:SYSTEM      ***     /
8   260    PDB$SEED:SYSAUX      ***     /
9   620    USERS                ***     /
10  505    PDB$SEED:USERS       ***     /
```

```
RMAN> exit
```

Recovery Manager complete.
[oracle@oracleenespanol2 admin]$ rman target sys/oracle@pdb1
Copyright (c) 1982, 2013, Oracle and/or its affiliates. All rights reserved.
connected to target database: CDB1 (DBID=808250731)
RMAN> report schema;
using target database control file instead of recovery catalog
Report of database schema for database with db_unique_name CDB1
List of Permanent Datafiles
===========================
File Size(MB) Tablespace        RB segs
Datafile Name
--- -------- -------------------- ------- --------
1    790    SYSTEM               ***     /
3    880    SYSAUX               ***     /
4    125    UNDOTBS1             ***     /
5    560    SYSTEM               ***     /
7    590    SYSAUX               ***     /
8    260    USERS                ***     /
9    620    SYSAUX               ***     /
10   505    USERS                ***     /
```

```
RMAN> exit
```
When you only connect to your pluggable (PDB1), you can only make a backup of the PDB1 database. RMAN can only allow you to recover one or all of your pluggable databases as well as your container database.
Another thing to keep in mind is that if you include your archivelogs (plus archivelog statement) when backing up your database, RMAN doesn’t do an archivelog switch. Archivelog switching will only happen at the container level, so when you do it at a pluggable level, RMAN will not perform the backup.

```
[oracle@oracleenespanol2 admin]$ rman target sys/
Recovery Manager: Release 12.1.0.1.0 - Production
on Sat Jan 25 16:54:00 2014
Copyright (c) 1982, 2013, Oracle and/or its affiliates. All rights reserved.
connected to target database: CDB1 (DBID=808250731)
RMAN> backup database plus archivelog;
Starting backup at 25-JAN-14
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=82 device type=DISK
specification does not match any archived log in the repository
backup cancelled because there are no files to backup
Finished backup at 25-JAN-14
```

```
Starting backup at 25-JAN-14
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00009 name=/u01/app/oracle/oradata/CDB1/EDDDC5E35CF7216DE043344EB2C0AB6F/datafile/o1_mf_sysaux_9c52f06t_.dbf
datafile file number=00008 name=/u01/app/oracle/oradata/CDB1/EDDDC5E35CF7216DE043344EB2C0AB6F/datafile/o1_mf_system_9c52f06f_.dbf
datafile file number=00010 name=/u01/app/oracle/oradata/CDB1/EDDDC5E35CF7216DE043344EB2C0AB6F/datafile/o1_mf_users_9c52f06o_.dbf
channel ORA_DISK_1: starting piece 1 at 25-JAN-14
channel ORA_DISK_1: finished piece 1 at 25-JAN-14
piece handle=/u01/app/oracle/fast_recovery_area/CDB1/backupset/2014_01_25/o1_mf_nnndf_TAG20140125T165412_9g8nq97_.bump
comment=NONE
Finished backup at 25-JAN-14
```

```
Starting Control File and SPFILE Autobackup at 25-JAN-14
```

As with previous versions of RMAN, one way to know what backups you have is with the command LIST. This report has added the Container ID record, which will allow you to identify which database your backup belongs to. Keep in mind that your container database will not have a Container ID reported; it is assumed that is the ID number is 1.

```
As with previous versions of RMAN, one way to know what backups you have is with the command LIST. This report has added the Container ID record, which will allow you to identify which database your backup belongs to. Keep in mind that your container database will not have a Container ID reported; it is assumed that is the ID number is 1.
```

```
In this version, Oracle recommends that you frequently backup your root database as it contains critical metadata for the whole CDB, while each PDB might fall into its own SLA’s depending on the lifecycle of it or of your business needs. The way to backup your root is to connect to the target and issue the backup database root command.

```
[oracle@oracleenespanol2 admin]$ rman target /
Recovery Manager: Release 12.1.0.1.0 - Production
on Sat Jan 25 17:12:31 2014
Copyright (c) 1982, 2013, Oracle and/or its affiliates. All rights reserved.
connected to target database: CDB1 (DBID=808250731)
RMAN> backup database root;
Starting backup at 25-JAN-14
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=82 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00003 name=/u01/app/oracle/oradata/CDB1/datafile/o1_mf_sysaux_9c52w6w_.dbf
input datafile file number=00001 name=/u01/app/oracle/oradata/CDB1/datafile/o1_mf_system_9c522mbz_.dbf
input datafile file number=00004 name=/u01/app/oracle/oradata/CDB1/datafile/o1_mf_undotbs1_9c524cnr_.dbf
input datafile file number=00006 name=/u01/app/oracle/oradata/CDB1/datafile/o1_mf_users_9c524bjm_.dbf
channel ORA_DISK_1: starting piece 1 at 25-JAN-14
channel ORA_DISK_1: finished piece 1 at 25-JAN-14
piece handle=/u01/app/oracle/fast_recovery_area/CDB1/backupset/2014_01_25/o1_mf_nnndf_TAG20140125T171238_9g8nq9h_.bump
tag=TAG20140125T171238 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:55
Finished backup at 25-JAN-14
```

```
Starting Control File and SPFILE Autobackup at 25-JAN-14
piece handle=/u01/app/oracle/fast_recovery_area/CDB1/autobackup/2014_01_25/o1_mf_s_837795277_9g8moy7n_.bump comment=NONE
Finished Control File and SPFILE Autobackup at 25-JAN-14
```

As with previous versions of RMAN, one way to know what backups you have is with the command LIST. This report has added the Container ID record, which will allow you to identify which database your backup belongs to. Keep in mind that your container database will not have a Container ID reported; it is assumed that is the ID number is 1.
Conclusion

In RMAN 12c, there are great new features that will help you if you are ever faced with a disaster. These new features will allow you to reduce your mean time to recover (MTTR) and ease your tasks in duplicating your databases. In part II of this article I will talk about Recovering tables and table partitions from RMAN backups, SQL Interface Improvements and Active Database Duplication.

References
Oracle® Database Backup and Recovery User’s Guide 12c Release 1 (12.1)
From http://docs.oracle.com/cd/E16655_01/backup.121/e17630/toc.htm
RMAN Recipes for Oracle Database 12c: A Problem-Solution Approach, 2nd Ed.
Darl Kuhn, Sam Alapati, and Arup Nanda (2013) Apress
My Oracle Support Notes:
1534487.1
1521005.1
1521524.1
1401574.1
1521075.1
RMAN | Pythian - Data Experts Blog
from http://www.pythian.com/blog/tag/rman/
SPFILE FOR A CDB VS. A PDB
from http://dbasolved.com/2014/02/14/spfile-for-a-cdb-vs-a-pdb/

Rene Antunez is an Oracle ACE; Speaker at Oracle Open World, Oracle Developers Day, IOUG Collaborate and RMOUG training Days; Co-President of ORAMEX (Mexico Oracle User Group) and Web Events Chair for IOUG Cloud Computing Special Interest Group (SIG); At the moment I am a DBA Team Lead at The Pythian Group
Formerly, he worked for Oracle as a Database Consultant and Solution Architect, which involved many challenging projects including, Exadata, ODA and EM 12c implementation, as well as upgrade projects specialising in backup strategy and high availability solutions.

In his free time Rene likes to say that he's a movie fanatic, music lover and bringing the best from Mexico (Mexihtli) to the rest of the world and in the process photographing it :)
The economics of ambition

My IT career started in 1984, when I graduated from Boston University with a degree in economics and a minor in mathematics. I had earlier chosen economics as a major only because I was mistaken in an assumption. I had believed that economics was the science behind business, similar to the way that physics might be considered the science behind most engineering. It was a foolish misconception, as I later came to realize that economics was more accurately described as the quantification of psychology, and not really a science at all.

In the end, I regretted choosing economics as a major, and subscribed entirely to George Bernard Shaw’s comment...

“If you took all the economists in the world, and laid them end to end, they still wouldn’t reach a conclusion.”

Needless to say, I ultimately found economics to be frustrating. It got worse when I started going to job fairs and interviews for jobs in economics. The final straw was an interview with the Federal Reserve Bank in Boston, where I sat across from a lethargic senior manager who instantly reminded me of a three-toed sloth, without the energy. She started our interview by stating in a tired monotone, “The first 3 weeks you’ll have 2 weeks vacation. Then after 6 years, you’ll have 3 weeks vacation. Then, after 12 years, you’ll have 4 weeks vacation. Then, after 20 years, you’ll have 6 weeks vacation. Any questions?” No ma’am, no questions. Where’s the door?

Then, I took a course in econometrics, which is a melding of statistics and economic research, and during that course I learned a statistical computing package called TSP (i.e. “time series processor”). Bingo! I didn’t so much care about the statistics or the economic research, but I loved the programming and the computer center. Soon I was a volunteer at the “batch window” at the computer center, helping other students with their glitches and problems. Computing and programming was an instant hit, and I began taking pure computer science classes. Since I already had a major, I didn’t have to waste time with prerequisite computer science courses, and could take the cool course like “operating systems design” and “artificial intelligence”. And so, I was in the process of shifting away from economics and toward computer science when I ran out of time and money and had to graduate. Boston University did not have a computer science program at the time, so all of the computer science courses were offered by the math department. So that is how I ended up graduating university with a major that I heartily disliked, along with a minor with a name that didn’t make sense because the field didn’t yet exist.

Sometimes you don’t start out with a plan, but if you pursue passion in everything you do, then often the plan finds you.

Tim Minchin once said the following during a commencement speech...

“Don’t ‘follow your dreams’ or hesitate to wait for dreams to emerge. Do whatever you’re doing, right in front of you, the best you can. Then, when that’s done, move onto the next thing and give that your all. Be micro-ambitious.”

My first job in IT in 1984 was a small startup company in New Jersey who were producing low-cost computer systems for small medical practices. I had learned to program in “C”, and was hired to deploy Informix’s C-ISAM library as a custom-built database management system. It was an intense introduction to systems and application programming, and focused on data from the very start.

Over the ensuing 31 years, my work life has remained intense. I joined Oracle Corporation in 1990, and by 1993 first became involved with RMOUG to present “PRO*C Tips and Techniques” at the 4th RMOUG “Training Days” conference at the US West training center in Lakewood. My senior director at Oracle at the time was a wonderful person named Valerie Borthwick, and she encouraged everyone on her team to share with the larger IT community, both inside and outside of Oracle. She believed that sharing expertise was a great way to firmly embed that expertise, and also to establish oneself as a skilled practitioner in the community. A few years later, I heard a colleague say it best...
meetups, where members of the Regis and RMOUG communities could meet on a Saturday morning or on a weekday evening for 2 hours to learn about a particular topic, using the workstation labs at Regis. It was a fantastic format, and resulted in a welcome boost in membership and involvement in RMOUG.

Over the past year, since I’ve joined Delphix, I’ve been attending meetups in the Denver/Boulder area on topics such as software testing, DevOps, SQL Server, and “R”, every so often encountering several other members of the RMOUG community. Meetups are easy, and that’s probably the point. The “meetup.com” website handles most of the logistics, providing the capability to easily create a webpage for the event, advertise via email and social media, with plenty of follow up messages to encourage the next meetup. You can RSVP for a meetup, and if someone comes up that prevents you from attending, it’s easy to cancel while letting the organizers know.

Meetups are fast and easy. In 2-3 hours, you can learn an entirely new topic that can help you in your job tomorrow, or help you in your job a few years from now. If you’ve been wanting to bend your career in a particular direction, then chances are there is already an established meetup community in that direction.

Meetups are small. It is not unusual for a meetup to attract only 4-6 attendees, and extraordinary to have a meetup attract 25-30. The intimacy of a meetup can be intimidating, but more often it is liberating. You can’t really hide very easily if you’re inclined to do so, but I’ve found that people are open to chatting if you wish, and respect your wish to just observe and learn in silence as well. Although “meetup.com” may have had its roots in online dating, it has come a long way since. Nowadays, an IT meetup is a group of motivated professionals, and the atmosphere is both casual and professional.

RMOUG is embracing meetups, because it increases the breadth and depth of educational opportunities for the RMOUG community. In addition to the recently-dormant Regis “DB Labs” meetup, new RMOUG board member Sruthi (Kumar) Annamnidu has established the RMOUG “Big Data” special interest group (SIG) and meetup, and there have been two successful meetups already. In addition, RMOUG member Ruslan Dautkhanov has attached a LinkedIn group to the special interest group, so that discussions about “Big Data” can take place and be archived on LinkedIn, visible to SIG and meetup members.

Coming soon, courtesy of RMOUG member and Oracle employee Sumeet Kabra, is a new special interest group and meetup for Oracle’s Engineered Systems. Watch the RMOUG website and social media channels on Twitter, Facebook, and LinkedIn for more information about Exadata, Exalogic, Exalytics, ODA, Zero Data Loss appliance, and other engineered solutions from Oracle.

Do you have a special interest which you’d like to explore more thoroughly? Do you want to share that enthusiasm?

To form a meetup for a special interest group, we need organizers. A subscription to “meetup.com” isn’t free, but RMOUG will handle that cost. What we need are folks who are willing to create the agenda for a 2-hour meeting by finding a speaker. RMOUG can help with finding speakers, but the meetup organizer would be expected to follow through, contact the speaker, and obtain the commitment. RMOUG can also help with finding a meeting location, but organizers are of course welcome to find their own locations for the event. Last, it is considered polite to arrange for some food and beverage at the event, since they are usually scheduled in the evening after work, when attendees might normally go home for their evening meal. It doesn’t have to be anything fancy, just something to tide people over. So organizers are also expected to find funding for about $30-$90 worth of food and beverage, depending on how many attendees RSVP. One common arrangement is to ask the speaker’s employer to provide that funding.

Overall, it is not a trivial effort, but the rewards are immense. Organizers establish themselves as leaders in the community, and have a front-row seat for all discussions.

Meetups are also an ideal way to learn how to give presentations. The setting is intimate, not much bigger than giving a presentation to colleagues at work. As we gain seniority in our careers, we are expected to share our expertise in such small settings. Presenting at a meetup or organizing a meetup is good preparation and training for taking that next step in our careers, moving from the plateau we’re on to climb to the next level.

As you can tell, I’m wildly enthusiastic about meetups.

I’m more wildly enthusiastic about RMOUG than ever, because I believe that RMOUG meetups are greater than RMOUG and meetups separately. RMOUG offers incredible benefits to the IT community in Colorado, including the world-class “Training Days” conference, coming next year on Tuesday 09-Feb through Thursday 11-Feb 2016, bigger and better than ever.

RMOUG also has the distinction of producing a high-quality quarterly newsletter, year after year, since the early 1990s. The only other local Oracle users groups doing so are NoCOUG (Northern California Oracle Users Group) and NYOUG (New York Oracle Users Group). You may have noticed that both those other Oracle users groups are located within enormous metropolitan areas (i.e. San Francisco and New York City, respectively), while RMOUG is located in the far smaller Front Range population. That kind of makes us the Green Bay Packers of Oracle users groups, which is another smaller community supporting something huge and world class.

And now RMOUG is evolving, encouraging meetups. We’re not seeking to control them or contain them, but rather to see them multiply, thrive and proliferate.

If you’re interested in finding or organizing a special interest group meetup at RMOUG, please send an email to “SIGdir@rmoug.org” or “peggy.king@rmoug.org”, and let the group know what you’ve got in mind. It will be time well spent, with enormous rewards for your career.

Because, as somebody wise once said, you can’t claim to know something well until you’ve tried to explain it to someone. Be micro-ambitious.
Meet Your Board

Tim Gorman  
President  
E-mail: President@rmoug.org

Bob Mason  
Vice President  
E-mail: VicePresident@rmoug.org

John Jeunnette  
Treasurer  
E-mail: Treasurer@rmoug.org

Chris Chase  
Secretary  
E-mail: Secretary@rmoug.org

Bobby Curtis  
Social Media Director  
E-mail: SocialMediaDir@rmoug.org

Susan Llanes  
Marketing/Vendor Relations Director  
E-mail: MarketingDir@rmoug.org

Dan Hotka  
Programs Director  
E-mail: ProgramsDir@rmoug.org

Art Marshall  
Web Director  
E-mail: WebDir@rmoug.org

Sruthi Kumar Annamnidu  
SIGs Director  
E-mail: SigsDir@rmoug.org

Mark James  
Scholarship Director  
E-mail: ScholarshipDirDir@rmoug.org
Karen Funston  
Training Days Technology  
E-mail: TrainingDaysDir@rmoug.org

Giri Venkata  
Board Member At Large  
E-mail: MembersAtLarge@rmoug.org

Komal Goyal  
Board Member At Large  
E-mail: MembersAtLarge@rmoug.org

Al Gonzales  
Board Member At Large  
E-mail: MembersAtLarge@rmoug.org

Vincent Giasolli  
Board Member At Large  
E-mail: MembersAtLarge@rmoug.org

Rene Antunez  
Board Member At Large  
E-mail: MembersAtLarge@rmoug.org

Kathy Robb  
Board Member Emeritus  
Arisant, LLC  
E-mail: BoardME@rmoug.org

Pat Van Buskirk  
Newsletter Director  
E-mail: NewsletterDir@rmoug.org

Ann Horton  
Oracle Liaison  
E-mail: OracleLiaison@rmoug.org

Peggy King  
Executive Director  
E-mail: Peggy.King@rmoug.org
Vlamis Software Solutions have partnered with Amazon Web Services to provide to you, free of charge, the opportunity to work, hands-on, with the latest of Oracle’s Business Intelligence offerings. By signing up to one of the labs below, Amazon’s Elastic Cloud Computer (EC2) environment will generate a complete server for you to work with.

RMOUG has invited the data warehouse experts of Vlamis Software Solutions to speak at the November QEW and bring their hands-on working labs for us to learn. Bring your internet-ready laptops. Can’t bring a portable computer? Not a problem. There will be laptops available for you to use.

These hands on labs are working with the actual Oracle software running on the Amazon Web Services EC2 environment. They each take approximately 2 hours to work through and will give you hands-on experience with the software and a tour of the features.

There will be an initial presentation on these topic; then you can get started on the walk-through of your choice. There MIGHT be time for you to do both! You can definitely take the materials with you and do the other from your office.

Business Intelligence

OBIEE

This test drive will present a quick overview of Oracle Business Intelligence Enterprise Edition (OBIEE) 11g and allow users to create reports and dashboards for themselves. The lab uses version 11.1.1.7.1 of Oracle BI and showcases the following features: adhoc reports, graphs, maps, and color-coding. In addition, this test drive will show you how to use OBIEE’s new map views to reveal location based patterns in business intelligence data sets. You will see how to create new map views, add chloropleth layers, add variable shapes, and display different measures at different zoom layers. You’ll learn how OBIEE can incorporate spatial analytics and integrate large geo-spatial data sets for location based analysis.

Big Data

Work with Oracle R and Big Data in this useful hands on lab. This test drive walks through some basic exercises on the Big Data Lite Virtual Machine. The Oracle Big Data Lite Virtual Machine provides an integrated environment to help you get started with the Oracle Big Data platform.

Watch for RMOUG’s email announcement for your opportunity to register for the November QEW
Tom Peters, cont’d from page 19

end enable;

function is_enabled( in_program varchar2 )
return boolean
is
  v_dummy varchar2(1);
  answer boolean;
begin
  answer := true;
  begin
    v_dummy := program_tab(in_program);
    answer := false;
  exception
    when no_data_found then
      answer := true;
  end;
  return answer;
end;

Next, any triggers that may need logic skipped are altered as follows.

create or replace trigger bu_my_table
  before update on my_table
  for each row

  update_not_allowed exception;

begin
  if program_control_pkg.is_enabled('bu_my_table')
  then
    if :old.status_code = 'CLOSED' and :new.status_code = 'CLOSED' then
      raise update_not_allowed;
    end if;
  end if;
end;

Now, any script that updates my_table can skip trigger logic for only its session:

begin
  program_control_pkg.disable('bu_my_table');

  update my_table
  set x = 32
  where x = 3.2
  and status_code = 'CLOSED'
  and create_date = '14-jul-2014';

  program_control_pkg.enable('bu_my_table');
end;

Credit goes to Steve Jun for the idea. I was merely the code monkey responsible for implementation.

Tom Peters is an Oracle/Java developer currently working for Pinnacol Assurance in Denver, CO. He gave his first RMOUG Training Days presentation this year entitled “Consuming RESTful Web Services with Oracle ADF.” Tom lives in Denver, CO with his wife and two children.
RMoug Training Days is Oracle Without Limits (O.W.L.) - technical sessions in all areas that interact with database technologies.

**Tracks**
- Application Express and ADF
- Business Intelligence
- Database Administration
- Database Tools of the Trade
- Development
- Emergent
- Middleware
- Oracle Apps
- Professional Empowerment

**Highlights**
- Two full days of training with a bonus half-day of 30-minute tech tips and two-hour deep-dive sessions
- More than 100 technical sessions with new tracks
- Full exhibition
- ACE lunch, WIT, and career panel
- Networking opportunities to meet and talk with Oracle ACE Directors, ACEs, and ACE Associates, product managers from Oracle, and other Oracle professionals

Bring your technical knowledge to the largest regional Oracle-related user group conference in North America.

- Abstract Submission Deadline: September 30, 2015
- Conference Registration Opens: October 15, 2015

Visit www.rmoug.org to submit an abstract, register for the conference or sign up as a sponsor or exhibitor.