Arisant focuses on understanding business requirements to ensure the most effective implementation of the right solution. Our strategy is aimed at providing an honest and expert brand of consulting services for both the Private and Public Sectors. Contact us today to discuss any of the following services, and more.

- Consulting Services
  - Architecture
  - Analysis
  - Design
  - Implementation
  - Project Management
- Staffing/Support
- Managed Services
- Managed Hosting Services
- Oracle Software Sales
- Oracle Hardware Sales
features

6 Migrating Critical Business Applications Successfully
by Karen Cannell
What Works, What Doesn’t, Lessons Learned

16 The Challenges of Hyperion Support
by Chris Chase
Part 1: Metadata Management - Governance is the Key

22 Oracle Golden Gate
by Bobby Curtis
How Many Ways To Monitor

monthly features

4 Letter From Our President
5 Stan Yellott Scholarship Fund
28 RMOUG Board of Directors
30 RMOUG Calendar of Events
30 August 2014 QEW

On the Cover:
From Jim Egan, a data warehouse consultant working for Mantis Technology Group, a ProKarma company - “The photo was taken in September, 2013 when I was hiking to the top of Mt Massive - the second tallest 14’er in Colorado. A group of three mountain goats crossed my path and then stayed on the path for almost 30 minutes. They would stop and look like they were waiting for me to take their picture. This happened a number of times and I took advantage of it each time.”

A special Thank You to Heidi Kuhn, Tim Gorman and Kellyn Pot’Vin, without whose continuing help, this publication would not be possible.
RMUOG’s calendar year matches the fiscal calendar year of Oracle Corporation, ending on 31-May, so another year at RMOUG has ended, and a new one has begun. At the May quarterly educational workshop (QEW) at the Oracle/Sun campus in Broomfield, a new board of directors was elected for RMOUG, and they have taken on their new roles and have hit the ground running, because RMOUG projects continue on a year-round tempo.

The RMOUG newsletter, SQL>Update, which you are reading, is produced quarterly, and in addition to the hard-copy editions mailed to members, downloadable copies are available at RMOUG.org > Newsletter > Newsletters Online.

RMUOG also holds quarterly educational workshops (QEWs) in May, August, and November, and then in February we hold our massive annual conference, Training Days, now approaching its 26th year. Information about each of these events are front and center at RMOUG.org during the months leading up to the event.

In addition to these projects, RMOUG now has five special-interest groups or SIGs:

1. Hyperion/EPM
2. Big Data
3. MySQL
4. Women In Technology
5. Fusion Middleware

Each of these groups is centered around their own LinkedIn groups or their own Meetup.com groups, and more information about each of these SIGs, how you can participate, and their activities can be found on the RMOUG website at RMOUG.org > SIGs.

Furthermore, because as IT professionals we owe it to our community to help train our own replacements, RMOUG has the Stan Yellott Scholarship Fund which is funded by donations and by profits from RMOUG’s other projects, and which dispenses small scholarships on a quarterly basis. Additionally, the scholarship fund also promotes attendance at the Training Days conference by area high-school students, and more. More information about the Stan Yellott Scholarship Fund can be found at RMOUG.org > Scholarship.

For all of these activities, RMOUG needs volunteers, people like you, for the following roles...

**Board of Directors**

We need volunteers for the board of directors, to make things happen. If you’re at all interested in understanding what the board does or how you can help, please contact RMOUG’s executive director Heidi Kuhn or any member of the board, found online at RMOUG.org > About > Current Board Of Directors. There are three groups of people on the board of directors:

1. Directors, voting members elected annually at the May QEW
2. At-large, not elected, appointed by majority vote of the directors
3. Emeritus, not elected, appointed former directors who serve in an advisory capacity

The directors are elected each year and they run the organization. At-large members are appointed by the directors to help the board in any way. Emeritus members are long-time former members appointed by the directors as advisory members.

If you are interested in joining the board, there is a contact form at the bottom of the page at RMOUG.org > About > Current Board Of Directors.

**Training Days volunteers**

Each year, the Training Days conference needs volunteers for three primary jobs...

1. Registration
2. Room ambassadors
3. Airport drivers

The registration volunteers staff the registration desk at the conference to welcome attendees and provide their badges and materials, as well as answer questions. The room ambassador volunteers manage each of the speaking sessions at the conference, helping and answering questions in the room, enforcing fire codes, introducing the speaker, and collecting session evaluations. Airport driver volunteers provide a friendly face to arriving speakers from out-of-town, meeting them as they arrive at the airport and seeing that they get where they need to go, as well as providing the same service outbound to the airport as the conference concludes.

Each of these volunteer positions may be eligible for a discount at the conference, so if you are interested, please fill out the contact form at RMOUG.org > Volunteer.
QEW or SIG volunteers

Help is always appreciated at the quarterly educational workshops or special-interest group events. If you are interested in helping out at any of these events, please fill out the contact form at RMOUG.org > Volunteer.

A question always arises: Why should I volunteer for RMOUG? I don't always know how to answer that, but my own personal inspiration was my mentor and former manager at Oracle Consulting, Valerie Borthwick, who was certainly one of the smartest and successful people I've had the pleasure of knowing. Valerie got her start at Oracle here in Colorado, and eventually she rose to the very highest levels at Oracle Corporation as an executive vice president, and then left Oracle to take on entrepreneurial leadership roles at start-up companies. At present she and her husband have retired to Aspen. Valerie knew that it was in Oracle's best interest to retain the best, and that the best get restless, and that being the best involves doing, not talking. She encouraged us to do and become famous in our community as people who acted and got things accomplished. There are people with great ideas and lots of talent and fun stories, but ideas and talent also require action. She wanted to be surrounded by solid people who could be counted on to act. For my part, I decided that RMOUG was a good place to enhance my career skills as well as share and grow, and I came to that conclusion over 22 years ago, and some of the best people I've known I met through activities at RMOUG.

Your mileage may vary. But I strongly encourage you to give it a try. You may drop it later, but you won't regret having tried.

Stan Yellott Scholarship Fund

RMOUG Scholarship Mission

To provide educational opportunities to members of the organization about the information technology industry in general, and in particular the technology of Oracle Corporation to include databases, storage, networking and application development, specifically the products and services of the Oracle Corporation.

To collect, post and distribute information about Oracle technologies and other related technologies to members.

To provide members with the ability to help their peers maximize their knowledge and skills working with products in information technology and in particular Oracle products.

To provide a consolidated channel of communication, conveying needs, concerns, and suggestions, for members of the organization, to Oracle Corporation and other vendor corporations involved with Oracle related technology.

To encourage members to present their information technology experiences using Oracle and other products and services.

To provide a consolidated channel of communication between members of the RMOUG and other communities in related information technology industries.

To promote educational opportunities for students of information technology through directed funding and services for educational purposes.

RMOUG is committed to supporting others in the pursuit of technical knowledge.

The Scholarship Fund started in 2001 to encourage future IT professional 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.

For Details, Visit the RMOUG Website

www.rmoug.org
Migrating Critical Business Applications Successfully
What Works, What Doesn’t and Lessons Learned

by Karen Cannell, TH Technology

Migrating critical business processes from Microsoft (MS) Access, Oracle Forms or other legacy platforms to Oracle Application Express (APEX) makes a lot of sense for a lot of reasons, but undertaking such projects demands prior analysis and planning that is often overlooked and underestimated. APEX migration projects pose challenges and transitions that merit discussion and planning beyond the simple “Convert this to APEX” treatment. From management, political, IT staff and end user perspectives, there is a lot more to consider than it seems at first glance.

This paper addresses the highlights and pitfalls of replacing existing MS Access, Oracle Forms or other legacy application with an APEX solution. We’ll cover conception (which apps are candidates?), planning, strategy, database design, user interface, and project execution from design through deployment, training and maintenance. The APEX Migration Process section covers the APEX Migration Workshop and SQL Developer Migration Workbench and describes where and how these utilities help, so that you may plan accordingly to use these tools or an outside framework for your situation. The Lessons Learned section points out successes (what works), failures (what doesn’t) and caveats encountered along the way (what we would have done differently), based on the author’s experience over years of successful large-scale migration projects. This collection of pointers for migrating critical business applications from non-Oracle, non-APEX applications to APEX will help you:

- Decide if an APEX migration is right for your legacy application, and
- If so, to plan life cycle time, resources and expectations to achieve a successful migration.

Migration to APEX

So, you have seen APEX applications, heard that one can covert spreadsheet applications within minutes, that APEX is database-centric, web-deployed, rapid application development tool and can be used for a wide variety of applications. You may have heard how APEX allows fast development of applications that are scalable and support essential web security features such as session state protection, LDAP authentication and complex authorization schemes. You have some in-house PL/SQL expertise, and have heard the APEX learning curve is not steep. So you think, let’s convert to APEX! Your data will be secure within an Oracle database and your application will be magically, rapidly (that is the RAD buzz-word, right?) converted to APEX, and all will be happy.

Well, as with any software project, nothing happens magically. It all needs to be carefully considered, planned and executed. Most important, the operative word here is not “Convert”, it is “Migrate”.

Whether you are moving from MS Access or Oracle Forms or another database platform or a home-grown legacy FORTRAN application, there simply is no tool that performs a conversion to APEX. What we are talking about is migrating critical business functions to an APEX application. To do this successfully requires a lot of up-front assessment, discussion, decision-making and planning. In most cases, a migration project is one of re-engineering: taking a business function on one platform and rebuilding an equal and better design on a new platform, in this case APEX.

Conception - APEX as a Migration Platform

Why consider APEX as a migration platform? The answer to that question depends on many factors particular to your current and future business plans. If one is interested in the security, scalability and features of an Oracle database, plus the advantages of web deployment are attractive, APEX may be good choice. In general, if you have existing, supported Oracle database(s), SQL and PL/SQL resources, in-house or readily available APEX development and deployment resources, an APEX migration may make sense.

A migration project is not the best time to learn APEX. There are many design and implementation judgments to be made during the migration process (in terms of application flow and page design) that require at least an intermediate level of APEX experience. If your organization is brand-new to Oracle and/or APEX, consider partnering with experienced resources for guidance.

APEX is a rapid-application-development tool that, when used properly, produces secure, scalable web-based Oracle database applications. The APEX Migration Workshop, a utility within APEX, provides a clearly-defined APEX migration path for those migrating from Oracle Forms and Reports or MS Access. The SQL Developer Migration Workbench assists in 3rd party database migrations to Oracle from SQL Server, DB2, Sybase, Informix, Teradata, MySQL and MS Access. There is support for the migration process available through Oracle Support and more readily through the Database and Application Migration forum.

Even with all that help, migration to APEX is not a cookie-cutter task. An APEX migration project is a process – it is not a one or multi-step conversion. A successful migration process, based on this author’s experience, requires more than just following the steps outlined in the Oracle Application Express Application Migration Guide, or those in the SQL Developer Migration Workbench. A successful migration project also requires:

- Experienced resources, legacy and APEX, to support the migration process
- Commitment to the process from all levels of stakeholders: Management, IT resources, and business/end users
Not every application is a good APEX candidate – but often the size and scope of legacy MS Access applications lend well to migration to APEX, usually affording better security, the ability to store and access greater amounts of data, improved page flow, and more modern user interface design. Persons often opt to migrate Oracle Forms and Reports applications to APEX due to in-house SQL and PL/SQL expertise, and the perception that the Forms-to-APEX learning curve is smoother than the transition to pure Java applications. The key factor on whether an APEX migration is right for your organization depends on whether APEX is a good fit for your organization. Will your developers adapt to APEX? If you have the infrastructure, Oracle database license and IT resources to support an APEX application, there may be a good fit. If your organization is not Oracle based, or if all your IT resources are Java programmers, APEX may not be a good fit. Scope is of course a consideration. If your legacy application aims to be the next Facebook or Twitter, APEX may not be your best platform. Assuming the application you tend to migrate is data-centric and slightly less ambitious than the next social media star, read on.

Migration

Migration generally means moving from one platform to another. Unlike birds, when we talk about software migrations, we mean migration in one direction: forward. We do not plan on migrating back at the end of the season.

In general, the original intent is to move the current functionality forward. Period. In reality, - in my experience – in successful migrations, this is seldom the case. A successful migration is a step forward in platform, functionality, and in user experience. In many cases, the resulting APEX application will little resemble the original. The APEX application will deliver the same business functions, and then some, in a more efficient, more secure, more user-friendly manner. The migrated product must be better than the original to be successful.

“But we don’t need more”, you might say. “We just need exactly what we have.” I have heard this from many budget-conscious managers, and I encourage you of this ilk to keep an open mind from the onset of migration discussions. Given the pace of today’s web technology advances, building the exact same functionality (as your legacy application) is a mistake. By the end of the project, one is behind. Consider the mobile phone you used two years ago. Would you buy the same one today? A few of you might. Of those, would your children use that older phone? Your end users won’t want to either. Accordingly, do not design your APEX application based on your legacy application’s limitations.

Moving from MS Access or Oracle Forms or other platforms to APEX - from client/server to web deployment, from desktop based to browser-based deployment - necessitates changes in how data is loaded, how data is stored, how data is passed, how data is presented and how data is reported. It necessitates changes in how your developers build and maintain the application, it necessitates changes in how your data is stored and in who manages it, and it necessitates changes in how your end users interact with the application. It is likely that as your migration project unfolds there will be a lot of re-evaluating and a lot re-engineering to accommodate business functions in the new platform. It is definite that as your migration project is deployed, there will be a lot of changes in how your end users work. Plan for this change from the beginning, and successful completion is more likely.

Goal

The end result must be better. A successful migration project will deliver more functionality that the original, and make end users happy. There must be value added for the business, and value-added for the end users, or they will not adopt it.

Planning Considerations

It is impossible to cover all the options for every possible migration project, however, the following points should be considered as guidelines for discussions and planning on your migration project. This list is by no means exhaustive – the particulars of your organization and applications may dictate more considerations not listed here. These points will serve as a starting point:

Small-Medium-Large

First, let’s have a realistic idea of what we are getting into. What is the scope of the migration project? One application or many? IF many, can they all share the same look-and-feel, or do they need to be branded differently? Identify all directly and indirectly affected applications. What connectivity issues are involved? What downtime will be involved, and how can the business accommodate that downtime? Planning is different for a small project than for one that changes the database platform and all applications for your entire organization. The small project can be handled in one phase; the larger project should probably be phased in both planning, development and rollout. Plan project phases according to project scope. Be realistic – changing business processes should be done with care.

The Business Process

Next, what is the critical business function or functions – the purpose –of the application(s) to be migrated? Do all functions need to be migrated? Do they all need to be migrated at once? Who performs these functions now? Who should be performing these functions? What is the data flow now? How should the data flow when it is stored in an enterprise database? Does an exact translation make sense here or not? An essential step in a successful migration project is determining exactly what needs to be migrated, and if necessary, building upon that later. Design and build only what is necessary, keeping in mind that the target platform is a web page or serious of web pages.

For example, one complex MS Access application loaded, consolidated and reported statistical results in a series of MS Excel spreadsheets. Each output spreadsheet contained multiple worksheets; each worksheet contained multiple tables and charts. The same was achieved in APEX by building one web page for each format of worksheet, and using select list to control the tables and charts displayed on each web page. This was possible because there was no business need to see every table and chart at once – that arrangement was simply to accommodate all the data in a single set of MS Excel spreadsheets. Since it is not practical – performance-wise and usability-wise – to display all the tables and charts on a single page, and because from a business perspective the end user would be only focused on one table and chart set at a time – the
APEX page design became very simple, and very efficient. In similar manner, every business form, every business report should be analyzed to determine what this equivalent web page function should be. This will require discussion between your business experts and your designers and your APEX experts, and it may be that the design will evolve as prototype versions are demonstrated and discussed. It may also be that a staged approach is necessary – migrate the critical business process first, then follow with a second phase that adds more advanced functionality at a later time.

**Database Design**

How your data is stored is very likely to change. When moving MS Access to an Oracle database, one is often moving from less-than-relational database structure to a – hopefully – true relational database structure. This means some database redesign. The same applies for other non-Oracle database. I have yet to encounter an MS Access database that is truly relational and mapped one to one into a third normal form Oracle relational database. I have encountered few Oracle Forms applications that migrated without some database changes – often the addition of views and security structures to better support APEX authorizations. Working with a database architect – or at least a resource who knows Oracle database design – is important at this stage. Working with an APEX resource or web designer who understands how web pages and database transactions interact is important here too. Do not assume that because your Oracle Forms application was a perfect relational database design that no changes are needed in the migration to APEX.

For example, I watched a talented Oracle Forms developer struggle with how to design an APEX page to replicate the Oracle Forms page. “I just cannot figure how to get APEX to handle a master-detail-detail-detail-detail relation.” He was perplexed, because the database design fit perfectly for building Oracle Forms blocks. He was right – there is no good way for APEX - or any intuitive web page – to handle that situation. What was required was a re-evaluation of what the page was doing – and why it all needed to be done on one page. In this case the task was three-fold: re-evaluate the business function – did it need to be done on one page, or was it best done in a wizard or in a series of transactions? – re-evaluate the database design, and re-evaluate the page design. Accordingly, the data structures - new views in this case – were adjusted to optimize the new APEX pages to serve the redesigned business process flow.

**Form Design – Web Page Design**

MS Access form design is not generally – good web page design. Oracle Forms page design is almost universally not get web page design. APEX pages perform best when the pages are designed to be what they are – stateless web pages. Each page should be one transaction. Each page should be simple and intuitive. I like to think, could I do this on my mobile phone or tablet? Like it or not, that is the technology measure we are faced with today. **(For an excellent discussion on user experience, see Lucas Jellema’s UXplosive Stuff, The Proliferation of User Interfaces, in the ODTUG Technical Journal Corner, http://www.odtug.com/technical-journal-corner).** Two years from now, will my end users still be able to use this application? I am not recommending building on the bleeding edge of technology. I am recommending that you be forward thinking and realistic. Your end users may not have advanced applications at work now, but they are using them in their home, on their TV remote, their X-box, their iPhone or Android. They can handle a little advancement at work.

Putting some thought into the overall application flow pays off. More important, NOT putting any thought into this aspect of the migration stands out like a sore thumb. Again, what “works” in MS Access or Oracle Forms does not necessarily work in a web environment. Case in point, one application of a series of migrated business functions was done by an outside developer, on contract, who did not have communications with the core development team who had migrated the previous 20 or so applications. Because there was no communication of “This is how we build apps” other than “Here is the template we use,” that developer copied the original MS Access app, almost form by form. This resulted in too much on one web page, and a very non-intuitive, confusing flow. Because there was no communication with the more seasoned APEX developers, this application got into production. The end users hate it. They have to use it, and suffer through it. Worse, the bad experience speaks louder than the other successful cases, causing a damage control problem. “That APEX app stinks…” carries like voices in a campground. End users are impressionable, and end users talk. If we are changing their jobs, we need to pay attention to design and build something that makes their job easier. Good web page design, and good web page flow makes a big difference. Here is where conversations between the business expert and a web designer and the APEX expert are important. Prototyping of APEX pages and page flows is a very helpful technique. I make sure to prototype early and often, demonstrating key features to the business experts to make sure their needs are being met in a manner they can work with.

**Security**

Whatever security was employed in your legacy application, moving to APEX affords the full suite of security features of the Oracle database. That is quite powerful, and opens many options. Some organizations opt for security in the database, at the lowest level possible. Some opt for database accounts, some for integrating with their organization’s LDAP security, then building authorization schemes based on functional roles on top of the basic corporate security. The right decision depends on your organization, on the nature of the data, the business functions begin perform and, whether the application is intranet or deployed to the external world, and of course corporate and legal security requirements. Questions such as, who has access to the data now? Who should have access to the data? Who can change the data now, and who should be able to change the data? How is the data loaded, and does it need to be encrypted on the way in the database, or both? All need to be asked and answered.

Using an Oracle database affords many opportunities to secure critical data in the database, where it is stored. How one builds an APEX page can determine how and where security is applied on data loading and on data reporting. That is not the end of it – security needs to be build into the data entry or data loading processes, into every web page, and into every report and data exchange operation. This means that security needs to be designed into and built into every APEX page. The APEX developers need to be aware of not just general web security practices, but also of specific security requirements of your business functions. For example, a web page that solicits a user’s name and email address for a mailing list has one set of requirements. A form that captures that same user’s billing and credit card information for a sale has - or should have - more secure requirements on passing and storing the credit card
information. When discussion the migration project, business users, designers and developers must be prepared to define and build in these security requirements.

**User Interface**

Look and feel of an application is important. You are changing how end users work, and if they do not like the new application, then creates more problems that if you had never changed the application at all. Buy in from end users can be more important than buy-in from managers and accountants. Sometimes user interface is dictated by corporate themes, style and usability requirements. When these exist, adhere to them. If they do not exist, create them and adhere to them. Create and use a common APEX theme when there are multiple APEX applications, so that users get the same experience from page to page from application to application. Common, well-designed interfaces promote end user confidence (“Oh, I know how these pages work...”) and are actually easier for developers to build and maintain. If no one in your organization has experience with user interface and web design, consider consulting for this advice, or at least adopt one of the APEX themes (Note that as of APEX 5.0, developers will have fewer core themes but easier opportunity to customize these via ThemeRoller – see APEX 5.0 Early Adopter for more information). User interface is too important to be practicing on – end users are very impressionable. They deserve an intuitive, reliable modern user interface to work with. If they don’t get it, they get cranky.

**Report/Output Design**

I mention report design in addition to web page design and user interface design, because reports are often misused. It is interesting to ask users what output they need, then to actually watch what they do with that output. Many times users will tell you they need a very wide set of data, or all the data, when what they really use is a small subset of it. In one situation, I have a request for all the data they meets these requirements. “All” the data amounted to tens of thousands of rows. It is plain not efficient to present tens of thousands of rows on a single web page. “OK,” they conceded, “but I need to download all those rows.” OK, so we built the download. In reality, what they users were doing was downloading all the data to Excel, then filtering, then graphing. What not just deliver the graph? Far smarter, efficient, secure and accurate to build and deliver a graph of the subset than to perform the huge download and desktop filter. Not to mention the fact that once data is downloaded, it is outside the database, and anything generated from the downloaded data is suspect.

Another common request is the need to print all the data. When we get over a couple hundred rows ... really? What is the cutoff in” really need a printout” versus just plain killing trees? In this situation, we had to build in the capability of printing – to PDF – tens of thousands of rows. We have no control over whether someone actually prints that whole data set. We did learn that the need to print stems from the habit of accessing the data offline – managers would download a set of data to have handy for the monthly review calls. They got in this habit because with the old app they were not always connected to the servers where they could access the spreadsheets – that contained all of the data. Part of the training – and hopefully learning – process of this application was to train end user is filtering for the data set that they need, then saving that smaller data set, printing a portion of that if they had to. The end user learning curve again is in gaining confidence that the data is there, accessible when they need it, and there is no need to download and save everything to their desktop.

Did I mention that in general, it takes seasoned end users a long time to change their work habits? In the self-help book “The Secret”, the authors suggest that one can change habits in 30 to 60 days:

 [...] changing the habit will take 30 days, re-affirming it further for another 30 days will definitely fix it and you’ll have no problem to continue from there on. That may be true for motivated individuals changing a habit they want to break. My experience with end users is that changing established work habits takes much longer. And MS Excel user? It takes months to years to wean MS Excel users off of their habits. In any migration project, we are changing how end users work. Changing the set of data a user has their hands on – i.e. all the data all the time - is a big change, one they likely do not want to make. My recommendation is to work with the end users to repeatedly demonstrate that their data really is available at all times, and to repeatedly demonstrate how to filter and save subsets of data – often in the form of saved interactive reports. Reassure them they can achieve the same results with less effort, less ink and less paper. And have time for more sophisticated work that downloading and filtering.

**Standards**

I touched on standards above while discussing User Interface – but standards merit a special mention. It is likely that your organization has standards for database design, for object naming conventions, for coding practices, for development practices, for user interface design. Your organization may have certain practices for source control and change management. If your organization has them, adhere to them. If your organization does not have them, make them, then make use of them and enforce them. Even when there is one developer, having and applying standards means that months from now, when called to revisit some long-forgotten feature of an application, it will be easier to decipher what was done then, and thus easier to implement what needs to be done now.

**The Overall Migration Process**

So far or discussion has been based on performing a migration within your organization, using APEX alone. There are organizations that specialize in APEX migrations, and others that have a particular framework for building APEX applications that simplifies the development process for certain features, still others have certain plug-ins that extend basic APEX functionality and may just suit your business needs. Include considering these offerings as part of your migration planning, particularly if your in-house APEX expertise is new or non-existent. A migration process is not the time to learn APEX. There is too much to learn, with too much at stake. Use of a framework may speed the development time significantly, enough to outweigh its costs. Use of a consulting firm experienced in migrations may serve to concurrently train your development staff, without the time and expense of off-site training.

Planning for your migration process may in fact take as long as execution of the migration itself. It is that important. The above list likely triggers additional topics that are essential in your organization.

**Resource Requirements**

The resource requirements for a successful migration from your legacy platform to APEX can be summarized in these points:
• A solid management decision and funding to do the migration.
• Commitment – buy-in – from all stakeholders, all levels, management to end users.
• One or more business experts in the application and in the business functions the application serves.
• One or more technical experts in the legacy application.
• One or more intermediate to advanced APEX developers.
• One or more technical business analysts familiar with APEX, to assess, guide and assist in translating business functions into APEX.
• Hardware and software licenses sufficient to host the application, including development, test and production instances.
• Database and Web Server support services.
• Support services, for when the application is in production.
• Clearly defined acceptance criteria – a test plan for confirming all requirements are complete.
• Training services, to transition end user from the legacy interface to the APEX application.

Communication.
This list covers the essentials for most organizations and is intended as a guideline. Your mileage may vary. Your organization may require more or fewer items, depending on the layers of management, the size of IT organization and the nature of the application.

Decision
The decision to do the migration must be solid and supported. There are many reasons to migrate, and sometimes just as many to NOT change a thing. The decision making process is outside the scope of this paper. What is important is that the decision is made, is a solid one, and has committed funding behind it.

Commitment
Decision does not mean commitment. In the case of a migration, a solid commitment from all stakeholders is required. This commitment must hold through the entire project. This means commitment from all levels of management, all IT resources, all levels of the business, managers to end users. The legacy developer needs to be willing to communicate with the APEX developer; they both need to communicate with the business expert and with the end users, they all need to communicate with management to convey questions, get clarifications, demonstrate prototypes.

This commitment needs to be solid all the way through to deployment and beyond. It will need to be campaigned for at the start of the project, then nurtured, coiled, revived and fostered throughout the project. Doubts, setbacks, bad vibes, bad press, delays, rumors, apathy – all these, at any stage of the project, will cause commitment to dwindle. The project manager needs to evaluate and reinforce the commitment at all stages. Yes, cheerleader is part of the job.

Business Expert(s)
At least one person needs to intimately know the business functions the original application currently supports and the business functions as they will be in the APEX application. This persons needs to know not just how things function now, but how they should be how they cold be improved. A migration project should not just replicate the current functionality, it should provide something better. The business expert is critical in saying what will work and what

Legacy Technology Expert(s)
A least one person needs to know the ins, outs and whys of the legacy application. Why was it built the way it is? How long ago was it built? What functionality is missing? What are the core algorithms that need to be carried forward? What features are deprecated? The legacy technology person – often the “owner” or the person who developed the application – has the best perspective on what the application does, how and why it operates the way it does. Often the lessons learned in development of the original application are helpful if not essential in migrating the functions to a new platform.

APEX Expert(s)
The APEX expert must be versed enough to envision how best to construct APEX pages, the page flow, and overall APEX web application design. This includes security, theme, page, report and item templates, component construction, overall application completion and delivery. This is not always building the same pages, forms and reports as in the legacy application. In fact, as mentioned above, it is most often NOT rebuilding the same pages and reports. Web deployment – and the option of mobile devices – opens the door for improved methods of accomplishing the same functionality, often in a more streamlined manner. The APEX developer must be able to envision and recommend improvements, prototype and demonstrate them, then complete and deploy them.

The business expert, the legacy technical expert and the APEX expert need not be three separate persons – if one person knows it all, that’s great. But don’t expect an APEX developer – even an expert APEX consultant – to perform the functions of the business expert and the legacy application expert. The point is, there needs to be communication between these three areas. If there is overlap – great. Business knowledge is essential to ensure the end product serves the business; Legacy application knowledge is essential to understand what the applications does, and to carry forward lessons learned; the APEX expert is essential to carry forward the existing functionality in the best, most forward-thinking manner possible. These resources work as a team.

Hardware and Software
Whether new or expanding on existing installations, ensure that sufficient hardware and software licenses, database and application server, exist for development, test and production environments. If your organization already has established APEX applications, this part is relatively simple: estimate the anticipated disk space and load requirements and make sure the existing hardware and licenses suffice. Include backup and recovery requirements, and be explicit about downtime restrictions.

If new hardware and licenses need to be purchased, plan for funding, procurement and installation time. If additional licenses for related products such as PL/PDF for printing are required, plan for procurement and installation of those too.

Database and System Support
Database and Web Server support services will be required, if your organization does not do this yourself. Plan ahead, as some-
times establishing these services often require planning quarters in advance. Set expectations on acceptable support response times, and be sure the planned support system can meet those requirements.

**End User Support**

Once the application is deployed, end users will need a means of support. If there is no established support center in your organization, you will need to create one. End users will need support to transition no matter how intuitive the application is. Again, set parameters for what is acceptable end user support response time. You don’t want to leave your users out in the dark.

**Acceptance Criteria**

As with any software project, define the acceptance criteria up front – this is the end goal. Have a test plan. The acceptance criteria define the key functionality that must be met for the migration to be successful. The test plan ensures the acceptance criteria are met. Your acceptance criteria may be purely functional, or may also include key performance and workload metrics. Without defined acceptance criteria, and an accepted test plan, your migration project can never be a success.

**Training**

Training will be required to transition end user from the legacy interface to the APEX application. Who will do this training? Will it be online or in-house? Will it be integrated with help integrated in the application? Who will train the trainers?

Training will also be required to transition in-house developers to APEX and the migrated application. If you employ a third-party organization for the migration process, be sure to include technical documentation and developer training, at least a technical walk-through, as part of the contract.

**Communication**

While not strictly a “resource”, communication is so essential is it included here to drive home the point: Communication is essential between all team members throughout the migration project. Weekly or bi-weekly team meetings, even when they are brief, keep everyone abreast of questions, decisions and progress (or pitfalls) made along the way. Short demonstrations of key features on a regular basis are important to keep the communication open and keep the image of progress strong.

Keep it positive, and keep it realistic. Lack of communication can kill a project just as well as rumors, bad news or inaccurate communications. Apathy, once it sets in, is hard to erase. I have seen an entire organization ignore a project from the mid-point on because the development team did not communicate any news – the project was proceeding like a black box. After awhile nobody cared about the black box in the corner, they went about their own work and ignored it. The developers, off in their own world, did not know this apathy had set in until it was too late. That migration project failed before it was ever deployed due to lack of communication. The home team had moved on, the developing team was left holding the box – a product that had no home. Communication on many levels – any level – could have saved that project.

**The APEX Migration Process**

As mentioned above, migration of any legacy application to APEX is a process – not a conversion. There is no one-step conver-
tor to move legacy applications to APEX. In very general terms, the APEX migration process is:

- Export the legacy database and application metadata (Exporter for MS Access, SQL Developer or manual means for other platforms)
- Migrate the database objects to Oracle (SQL Developer Migration Workbench, or other SQL Developer data load means)
- Migrate the applications components – processes, triggers, forms and reports – to APEX pages
- Organize and consolidate the APEX pages into a cohesive APEX application

There are some Oracle-provided tools to help. The APEX Migration Process actually entails, depending on the source platform, optional use of these tools:

- The Oracle Exporter tool, to extract the application definition and data from an MS Access application (MS Access versions up to 2007 )
- The Oracle Forms2XML Utility, to extract Oracle Forms application
- Components to an XML format.
- The SQL Developer Migration Workbench, to convert legacy database data and objects into Oracle Database data and objects (MS Access, Oracle Forms, SQL Server, MySql, Sybase, DB2, Informix, Teradata, as of SQL Developer 4.0)
- The APEX Migration Workshop, to manage the migration of all components of your application (MS Access and Oracle Forms)

In addition, as of Oracle 12c and SQL Developer 4.0, the SQL Translation Framework is available to assist with Sybase, SQL Server and DB2 database migrations:

- The SQL Developer, SQL Translation Framework, to translate non-Oracle SQL statements into semantically-equivalent Oracle syntax (Sybase, SQL Server and limited DB2 only, as of SQL Developer 4.0)

The following sections outline the highlights and basic operation of these tools.

**The Exporter Tool (MS Access)**

The Oracle Exporter is a helper tool tailored to earlier MS Access versions. Its sole purpose is to export the MS Access objects to an XML format, for import of objects into the APEX Migration Workshop, and the database objects into a .SQL file for import into the SQL Developer Migration Workbench. For cases where the Exporter tool does not apply, conversion of database data and objects will need to be done by other means – explicit copy to flat file then import, use of SQL Developer data loading options, or even SQL*Loader. The best option depends on your legacy database. The reader is directed to the SQL Developer Migration Workbench documentation for details migrating database objects from your source database platform.

Which version of MS access you are starting form makes a difference. The Exporter works for MS Access 97, 2000, 2002, 2003 and 2007. There is a separate Exporter download for each MS Access version. For MS Access 2010, there are currently no plans for a new Exporter version. The options are to save the MS Access database in an earlier MS Access database format (an option that may not be possible, if your database uses the latest database fea-
The general steps in migrating an MS Access or Oracle Forms application to APEX include:
1. Export the MS Access metadata using the Exporter tool (if MS Access).

The goal is to get the data moved to Oracle, stored in a relational database structures that support the as-migrated application. As mentioned above, this often means redesigning the table structures and reloading data accordingly. SQL Developer offers numerous ways of importing data from third-party databases. Once the data is accessible, it is a simple matter for a knowledgeable database architect or SQL and PL/SQL developer to refine the database design, extract and load the data into the new structures. Consult the SQL Developer Migration Workbench documentation and tutorials to determine the best approach for migrating your source database to Oracle.

The APEX Migration Workshop

The APEX Migration Workshop is for MS Access and Oracle Forms applications only. The APEX Migration Workshop is more of a migration management tool than a step-by-step process. It is essentially a project tracking tool tailored to typical MS Access and Oracle Forms application migration processes. It is not a how-to-conversion tool. It does not convert Oracle Forms form to APEX forms, MS Access reports to APEX reports, Oracle Forms code modules to APEX processes or database stored procedures. It does not convert Visual Basic macros to PL/SQL procedures and functions. There is, at present, no Oracle-based utility to do such conversions. The APEX Migration Workshop simply lets you track which components need to be converted, and map these components into APEX pages.
Use the SQL Developer Tools → Migration → Migrate interface, following the wizard for the options that suit your source database. Figure 4 illustrates Step 1 of the Migration Wizard. There are options for both Online and Offline database migrations. MS Access migration is offline, using the .XML output from the Exporter.

Hint: For MS Access, I had better luck using SQL Developer 3.2. In SQL Developer 4.0, The Microsoft Access Exporter is not there (I had to download from the APEX Migration Workshop links), and my Migrate process failed. It seems that SQL Developer 4.0 is geared more for SQL Server, Sybase, DB2 and other larger database platforms migrations. This makes business sense for Oracle, but makes migrating you MS Access application more tedious.

If migrating from Oracle Forms, the database is already Oracle, however, analyze the database design and adjust as necessary to best serve the migrated application. Just because it was an Oracle database does not mean it was well-designed.

If migrating from MS Access, analyze the database design and adjust as necessary to best serve the migrated application. Just because it was an Oracle database does not mean it was well-designed.

1. Create an APEX workspace to hold the migration project.
2. Create an APEX migration project within the workspace, using the .XML file output from the Exporter tool.

![Figure 5 – Creating the APEX Migration Project](image)

Figure 5 illustrates the Create Project interface.

The created project lists the tables, queries, forms, reports, modules and pages in the source application, as shown in Figure 6.

3. Create an APEX workspace to hold the migration project.
4. Create an APEX migration project within the workspace, using the .XML file output from the Exporter tool.

![Figure 6 – The Created Migration Project](image)

Figure 6 illustrates the Create Project interface.

The created project lists the tables, queries, forms, reports, modules and pages in the source application, as shown in Figure 6.

5. Analyze the MS Access/Oracle Forms application, using the APEX Migration Workshop.

This step identifies application tables, queries, forms, reports, code modules and pages. This analysis step is where the important work of the migration process begins. The developer needs to identify which forms, reports and code modules are to be brought forward, which should be retired, which should be redesigned into new APEX pages, which need to be recoded into PL/SQL packages or other code modules, etc. This is where the true transformation of the application happens.

This is where the business expert and the APEX expert need to collaborate. The business expert knows what function(s) needs to be achieved. The APEX expert needs to listen, envision and advise on how the function may be created in APEX. Unless the application is very simple, there is likely to be some back and forth here, even when there have been extensive planning sessions. Discussions here may lead to better layouts, streamlined page flow, or subtle changes that increase usability.

Communication, collaboration and prototyping are important here so the business expert sees how the APEX application is unfolding. It is better to have communication and validation at these early page design stages than to wait until later in the process when it is more difficult and more costly to redesign and refactor pages and other application components.

6. Generate the new APEX application.

Once all objects to be generated are validated, one is ready to generate the application. This step entails generating simple APEX pages from the components identified in the analysis step. These generated pages are basic, and most always need beautifying and some item rearranging. These pages form the base application. A big step – but there is ore to do.

7. Customize the APEX application.

In this step the application becomes real and useable. It turns into something you are proud to deliver to your end users. Depending on the scope and complexity of the application, this step may take a significant amount of time.

8. Plan for enhancements! A developer’s work is never done ...

These later phases of the migration process – Analyze, Generate and Customize – are where in-house knowledge of the legacy application and in-house PL/SQL and APEX knowledge are important. The APEX Migration Workshop does identify program modules to be converted, but a live developer must do the actual conversion, then record its completion in the Migration Workshop. The Migration Workshop can also identify, in the migrated database structures, tables with no primary key, syntax of queries, or other object dependencies, but it is up to the APEX developer to address them. Finally, the APEX Migration Workshop can identify pages to generate, but it is up to the APEX developer to turn those basic pages into an awesome application. If you chose to user a third-party to assist with the migration process, this is where that investment will pay off.

If one is not migrating from MS Access or Oracle Forms, direct use of the APEX Migration Workbench is not possible. However, the steps outlined above remain the same. One still needs to migrate the database to Oracle, assess the source applications tables, queries, forms, reports, modules and pages to migrated, analyze those components to be migrated, generate an APEX application that includes all components to be migrated, and refine the newly-generated pages into a cohesive, professional APEX application. The key difference is, one must track those components to be migrated without use of the APEX Migration Workshop, and one must therefore build the corresponding APEX components from scratch. The APEX Migration Workshop is a time-saving and process tracing application – a helper. Its use is not a requirement. In fact, once the components to be migrated and their corresponding basic pages are generated, many organizations abandon use of the Migration Workshop at that point. The remaining project phases are closer to normal APEX development, enhancement and maintenance projects. Where the APEX Migration Workshop leaves off, various
project management features of the Team Development utility may serve for continue component tracking. Of course these project management tasks could also be ported to your organization’s usual project management tools.

**SQL Translation Framework**

The SQL Translation Framework does not directly help you migrate your application to APEX. However, the Framework is mentioned here to prevent any misunderstanding – this is not a tool that translates application components to PL/SQL modules. Available as of Oracle 12c, the SQL Translation Framework translates non-Oracle SQL statements of a client program to Oracle-style SQL statements. This enables existing client-side application code to run unchanged against an Oracle database.

The SQL Translation Framework is for applications that use ODBC, JDBC, OLE DB or .NET drivers to connect to the database. In these cases, using the Translator allows one to connect to an Oracle database without significant application changes. In this sense, it is more an alternative to a migration to APEX (or any other development tool). It will be interesting to see how the use of the SQL Translation Framework is adopted as Oracle 12c becomes more widely deployed.

![Figure 7: SQL Translation Framework at work](image)

Figure 7 illustrate the SQL Translation Framework at work. Note that code is not changed – instead, a translator is applied to the existing, non-Oracle SQL. The result is “Oracle” execution with no SQL changes.

**Future of Oracle Migration Tools**

Tools currently exist for the most common migration platforms – MS Access and Oracle Forms for the APEX Migration Workbench, and the most common commercial databases, for the SQL Developer Migration Workbench. It is unclear at this writing how many APEX-specific migration tool improvements will be released in the near future.

Note that there have been few recent (APEX 4.0 and 5.0) advances in the APEX Migration Workshop. This is presumably due to higher demand for other APEX enhancements (HTML5, mobile design, developer productivity), and the fact that most APEX migrations are not component by component conversions (which the APEX Migration Workshop tracks beautifully), they are really re-engineering projects. In this sense, a migration project need not be tracked in the Migration Workshop – it can just as easily be tracked in the APEX Team Development utility, the difference being that the components need to be entered (as with any project tracking utility), as opposed to being uploaded.

In SQL Developer, where the bulk of data and data structure migration takes place, the most recent enhancements are geared to assist in a wider variety of third-party database (SQL Server, Sybase, DB2, MySQL, Informix, MS Access, etc.) to Oracle conversions at the data and data structure level. These advances make sense, as the major addressable business problem, from an Oracle perspective, is to transition the data into an Oracle database. The SQL Developer Migration Workbench helps you do that. Regardless of the source platform, migration of the business logic components will still need to be manually re-engineered into SQL and PL/SQL components – your migration project plan still needs to determine which components need to be migrated and track by who, when and how they get migrated into the final APEX application.

**Help**

Oracle Support does assist with database migration issues, through SQL Developer Migration Workbench and through the APEX Migration Workshop. In addition, there are focused Oracle user forums – now called Spaces – for migration projects. These forums are monitored by knowledgeable persons, and if the answer to your question is not already there, you can expect a prompt response.

There is a separate forum: *Microsoft Access Migration to Oracle APEX*

**Alternatives**

If all of the above seems daunting, or you simply do not have the in-house APEX or Oracle database expertise, consider consulting one of the many businesses offering APEX migration services. There are several established, reputable companies that offer migration to APEX services, some specializing in MS Access or Forms migrations, others less platform-specific. All of these companies employ the concept of a migration repository, and generating or developing APEX applications based on the to-keep components of the migration repository – essentially the same process as the SQL Developer Migration Workbench combined with the APEX Migration Workshop. What these third-party companies add is expertise and experience – they are familiar with APEX, and they are familiar with migrations. They are familiar with web development, and they are familiar with common migration solutions regarding security, performance and tuning. Consulting with a third-party team to assist with your migration does not take you other of the picture, however. You will still need to provide legacy application knowledge and business expertise. You will still need to participate in the analysis and re-design processes. You will still need to keep communications open, and ask for frequent progress reports and prototypes as the migration progresses. Your will need to define the acceptance criteria, and you will need to perform the acceptance testing. You should be involved in the feedback process throughout the project. The trade-off is cost. If you do not have the in-house APEX expertise, consider using a third party for the APEX development – the overall costs over the duration of the project may be well worth it.

**What Works and What Doesn’t**

In a nutshell, the following are some key points on what works, and what doesn’t, and lesson learned from over 25 years of software “Conversion”, “Migration” and “Upgrade” (all words for the same thing – moving from one platform to another, in this case APEX) projects:

**What Works**

- Sound Relational Database Design – Don’t shortcut here, it costs in performance and page design in later stages of
the project.
- Re-engineer for an Oracle database, for web interfaces, for web deployment.
- Have Standards and apply those standards.
- Communicate, Evaluate, Reassess, Communicate, Repeat.

What Doesn’t Work
- Working in a vacuum – Don’t keep development separated from the business and end users. They are key stakeholders, and need to be involved. They actually know more about how the process should work than the developers do!
- Tacking the migration as an isolated project. If this is a critical business function, it is likely not isolated from the rest of your business. Keep all stakeholders informed at all stages of the project.
- Treating the migration as a straight conversion. Conversion is lateral. Migration is moving forward.
- Using your migration project as your APEX training project. Do not risk your critical business functions on a training run. If this is your first APEX project, get assistance, it will be worth it in the long run.
- Skimping on training – end user and developer. Do not assume that your end users will naturally migrate to a web-based application. Not everyone makes the transition gracefully. Give them training, and give them support, for as long as it takes.
- Skimping on communications. No project proceeds well in a vacuum. Keep communications open and honest throughout the project. “No surprises” is a good thing.

Lessons Learned
Perhaps the biggest lessons learned are:
- Know your end users. Know how they really work, not just what they tell you.
- Maintain communications with the business users throughout the process.
- Keep showing progress – Even small sprints of progress maintain a positive attitude toward the project, preventing both apathy and negative press. When there is a setback, demonstrate it, document it, discuss it and the plans to move forward. Then move forward.
- Keep validating progress – Don’t make assumptions, and don’t let the business slack off on their testing and validation, even when it seems all is going well. The migration project is a contract between the business, the end users, and the development team. All parties need to take part in that agreement.

Even the best of working relations can end up with problems if the cycle of communicate, build, test, and validate is broken. In one long-term project, development as been going well, new features were added in accordance to the project plan and briefly demonstrated at each sprint end. Then, for scheduling reasons, the business team passed on in-depth testing of a key deliverable. In the demo, we all agreed it looked good. However, weeks down the road, a key business expert exclaimed “That’s no good at all – it needs to do this.” Well, at that point, “this” meant rebuilding the entire page with a new set of constraints and business rules. In fact, there was a lot of customization to do to get the functionality as it should be. In both cases, the visual result was a tabular form – on the surface, it looked good. The business rules behind the scenes were significantly different – and that behind the scenes functionality had not been completely tested and flushed out at sprint end. The result was the time and resources – and lost sleep – in rebuilding the feature. The other result was closer communications between the business expert and the APEX developer for the duration of the project – in fact that communication is ongoing. It was a costly lesson learned, but one that won’t be forgotten.

Honesty
Honesty seems like an odd topic to address in a software paper – but not really. It is my experience that honest communication at all stages, even when the news is bad, is better than giving misleading information that leads to doubts and lost credibility. Honest time assessments, honest evaluations of results, honest feedback – these make everything easier in the long run. There is rarely-never a chance to go back and refactor. Spending the time, within reason and budget, to do it right the first time is a good investment. If there are delays, document them, learn from them, and adjust the features added if there is a hard time or budget deadline. Keeping honest communications flowing means there are no surprises at the end of the project.

Conclusion
Migration to APEX from any legacy platform is a process. The process involves re-evaluation of your critical business functions and most often re-engineering from the database structures on up to the completed web application. Assessment of essential business functions to be migrated and how best to migrate these functions to a web interface is an essential part of the migration process.

There are Oracle-based tools to assist in the migration process, depending on your legacy database. The SQL Developer Migration Workbench assists in tracking and migrating database elements, structures and the data itself. The Migration Workbench is most useful for earlier MS Access versions (97, 2000, 2003 and 2007), and more recently SQL Server, MySql, Sybase, DB2 and Teradata data migrations. The APEX Migration Workshop assists in tracking application components to be migrated for MS Access and Oracle forms migrations. The Migration Workshop assists in transitioning, or not, legacy application components into APEX pages and ultimately into a coherent APEX application. For migrations from other platforms, the APEX Team Development utility can be used to track components to be migrated and progress on those components along the way. Outside of these Oracle-provided tools, a migrations project is just like any other software project – it needs to be planned, tracked and monitored the same as any other software development project. Standard software management processes apply, in accordance with the size and scope of your project.

These migration tools alone are not enough, however. Any migration from any legacy database application to any Oracle development tool, APEX included, requires re-evaluation of business requirements, re-evaluation of user interface requirements, re-evaluation of data elements and data structures, and re-evaluation of application flow and managing user expectations. In short, plan on re-engineering much if not all of the application.

An APEX migration project is a process, toward the end goal of an improved business application. A successful migration process involves considering all of the factors mentioned in these pages,... and is worth it for all in the long run.
The Challenges of Hyperion Support

Part 1: Metadata Management – Governance is the Key

by The Bean Consulting Group

Introduction

When we were approached by the RMOUG to write something for the quarterly SQL Update publication, we felt that the unique challenges of supporting a Hyperion application environment would be an appropriate topic – particularly given that Hyperion applications are becoming more and more “enterprise applications,” supported from within the IT department, rather than部门al information silos, which have traditionally been managed from within finance or accounting. In addition, we thought this topic might provide some insight to help bridge the gap between the IT and application DBAs who typically support enterprise applications and the finance and accounting groups that typically support Hyperion applications.

Supporting Hyperion environments involves a unique approach to critical thinking and problem solving that requires an understanding of not just the technology being used but also of how the technology is being used. And, as with any system, troubleshooting issues in a Hyperion environment requires an identification of the primary issue, or root cause, before taking any action; however, this is particularly true when Hyperion environments are involved since an identified symptom can potentially have several different causes, and the solution to the problem can vary depending on the root cause. So, it becomes imperative that the people supporting the Hyperion environment understand both the technical aspects of the Hyperion system and how the system is used from a functional perspective.

With that as a “backdrop”, we decided that there were several key topics that should be addressed, and that a series of articles would best serve our goal of providing a better understanding of what is required to support a Hyperion system. These topics include Metadata Management, Smart View and User Support, Process Management and Workflow, Data Integration, Ownership, and Security, and Reports and Reports Studio.

Why did we choose metadata management as the first topic? Because metadata forms the building blocks for any Hyperion / EPM application, and it therefore made sense to us for it to be the first, and perhaps most important, topic we address related to the maintenance and support of Hyperion applications. Metadata, as we’ve all perhaps learned recently, thanks to news about the collection of telephone records, is defined as “data about data.” However, as Wikipedia notes, “the term is ambiguous, as it is used for two fundamentally different concepts. **Structural metadata** is about the design and specification of data structures and is more properly called ‘data about the containers of data’; **descriptive metadata**, on the other hand, is about individual instances of application data, the data content.” Metadata in the context of Hyperion applications and environments is considered structural metadata and it defines the dimensional hierarchies that are used to store, and more importantly, retrieve data from the Hyperion applications. As an example, the dimensional hierarchies for a Hyperion application might include Accounts, Periods, Years, Scenarios, Departments, and Products as shown below:

These structures are then used to store data defined by the “intersection” of the dimensions.
Furthermore, we believe that governance of structural metadata is of primary importance in a Hyperion environment. Wikipedia defines the process of governance as “a theoretical concept referring to the actions and processes by which stable practices and organizations arise and persist.” Put another way, specifically related to metadata management, the process of governance provides a mechanism for maintaining and controlling the data structures that define the enterprise applications. And, when these data structures are “tainted” in any way, such as through redundant members or inconsistent business terminology, the result can be incorrect information coming from the system being used to make enterprise decisions.

We have identified four (4) primary methods for managing and maintaining a Hyperion environment’s structural metadata. There may of course be other methods, but we believe the four methods we have identified encompass the best methods depending on the specifics of the environment.

**Classic**

Hyperion has been a powerful business tool for well over a decade, so perhaps a brief definition of “Hyperion” would be helpful at this point to make sure we are clear. Hyperion for our purposes is a suite of products, typically used by finance, that include Essbase, Hyperion Planning, Hyperion Financial Management, Hyperion Profitability and Cost Management and a several of other related, “edge”, products. When these products are used in a “classic mode”, the structural metadata is managed within that “product silo”. This makes it much more difficult to share that structural metadata across multiple applications.

**Enterprise Performance Management Architect**

Enterprise Performance Management Architect (EPMA) is a feature of Oracle Hyperion Foundation Services that integrates with Hyperion Planning, HFM, Essbase, Profitability and Cost Management, and Data Relationship Management. EPMA enables administrators to manage, create, and deploy Hyperion applications within one graphical user interface.

This provides the ability to make changes to a dimension and have it carried through to all applications which use, or subscribe, to that dimension. This, of course, can save a tremendous amount of time if there are multiple applications. Additionally, EPMA allows metadata management across product offerings such as Planning, Essbase, Profitability and Cost Management, and HFM. So, as companies realize the benefits of having multiple Hyperion products, metadata management can become more and more complex utilizing classic applications. With a graphical user interface and platform for updating metadata across multiple applications, Enterprise Performance Management Architect can be a very good choice.

**Enterprise Resource Planning Integration (ERPi)**

ERP Integration Adapter for Oracle Applications (ERPi) is a module of Oracle Hyperion Financial Data Quality Management (FDM); furthermore, it is an interface that sits atop Oracle Data Integrator (ODI) and leverages ODI as its processing engine to handle large amounts of data and metadata. At a high level, ERPi is a tool for centralized data and metadata that creates a seamless integration from source systems into several different types of EPM applications, including Hyperion Financial Management, Hyperion Planning – Classic, EPMA, and even Public Sector Planning and Budgeting – and Oracle Essbase. Supported source systems, include the following:

- Oracle E-Business Suite 11i and 12
- PeopleSoft Enterprise Financial Management 9.0 & 9.1
- PeopleSoft Human Capital Management
- J.D. Edwards General Ledger system

**Data Relationship Management**

As defined by Oracle, Data Relationship Management “is an enterprise change management solution for building and retaining consistency within master data assets despite endless changes necessary to support underlying transactional and analytical systems.” The key difference between DRM and some of the other tools discussed is that it acts as a central hub for maintaining chart of accounts, cost centers, legal entities, reporting structures, and analytical dimensions. It unifies cross-functional perspectives to a single master record by collapsing separately maintained structures into one structure. Validations and business rules that enforce certain enterprise-wide policies as well as user and object security all facilitate consistency and integrity with master metadata. Flexibility also remains a key feature. Users can create and manage alternate hierarchies through Versions and even Blend various elements of two or more versions to meet different reporting requirements. All related additions, revisions, and any other update is then synchronized with subscribing downstream systems, including BI/EPM applications, ERP systems, and even Human Resources Management systems.

**Metadata Management Options in Hyperion**

Classic

Essbase is a multidimensional database that provides a platform for building analytic applications. In a classic Essbase application, metadata management occurs in Essbase Administrative Services (EAS) which is a java based, graphical user interface.

**Oracle Essbase Administration Services**

EAS offers many benefits in terms of metadata management, but one of its most powerful tools is a Rules files. In EAS, you can build two types of Rules files: Data Load Rules and Dimension Build
Rules. In the case of metadata management, an Essbase administrator would use Dimension Build rules to build large hierarchies that have a significant number of members. The other option is to add members individually though a manually process, which may also require management of member properties on an individual, member-by-member, basis.

Hyperion Planning is a budgeting and forecasting tool that integrates financial and operational planning processes to improve business predictability. In a classic Planning application, metadata management occurs through the Dimension editor, which is a central location within each Hyperion Planning application. It offers a graphical interface for managing dimensions for that application, and that application only.

Hyperion Financial Management (HFM) is a consolidation and reporting tool that provides managers with the ability to rapidly consolidate and report financial results, meet global regulatory requirements, and reduce the cost of compliance. HFM operates as a multitier system, and its configuration contains three logical tiers: a database tier, application server tier, and client tier. In a classic HFM application, metadata management occurs in the client tier through the Load Metadata feature as shown below:

Once the application is open, perform one of the following actions to change metadata:

- Save the Application files and load them into the Financial Management application from the Web or Win 32 Client
- Make the modifications directly within the XML or APP files and load them into the Financial Management application

Similar to a classic Planning application, metadata management in a classic HFM application is specific to the application, which results in more maintenance and less consistency.

*Enterprise Performance Management Architect (EPMA)*

EPMA consists of three modules: Dimension Library, Application Library and Calculation Manager. Metadata management within EPMA occurs in the Dimension Library and the application deployments occur in the Application Library to sync any changes to the metadata back to the application. Updates to hierarchies and member properties may occur in two places, depending on the requirement – either in the Shared Library or with an application level “over-ride”.

If an administrator needs to add a member to a dimension used in multiple applications, such as scenario, the change would likely occur in the Shared Library. On the other hand, if the modification is to a specific property relevant strictly to the application, the change would occur at the application level. Multiple members are being added or a significant number of property updates are occurring, using Import Profiles with an ADS file is a more efficient method. An ADS file specifies parent/child relationships and related properties for multiple members.

In addition, if metadata between the shared dimension library and the application level becomes out of sync, or if a local dimension: a dimension specific to an application, needs to match a dimension in the shared dimension library, the administrator can use the synchronize feature in EPMA to achieve these types of functions. The synchronize feature is a powerful tool that allows an administrator two options to synchronize local dimensions in an application.

- Push metadata from a dimension in the Shared Dimension Library to the local dimension.
- Push metadata from the local dimension to the dimension in the Shared Dimension Library.

With either option you can Merge As Shared or Replace. Use the Merge As Shared option to merge similar dimensions from two different applications; essentially, you will be creating a master dimension. The synchronize feature is extremely helpful in managing metadata: it can help reduce maintenance, create consistency and minimize complexity.
Enterprise Resource Planning Integration (ERPi)

Prior to diving into the specifics of ERPi, it is important to distinguish it from its stand-alone, related tools to determine why it is the future of integration. As mentioned, ERPi is a module for Financial Data Quality Management (FDM) that leverages ODI as its processing engine, so how do these two products differ? FDM is strictly a data integration tool that provides drill-through capabilities from a Hyperion Planning or Hyperion Financial Management application. End-users, from EPM Workspace, can access further detail on the FDM relational repository as it relates to consolidated application. End-users, from EPM Workspace, can access further detail on the FDM relational repository as it relates to consolidated values on data forms or reports. However, it cannot manage metadata mapping rules are used to integrate. However, Oracle offers a central hub where database/reporting structures are maintained and validated before moving throughout the entire enterprise. Business Intelligence (BI) solutions, EPM applications, data warehouses are all maintained and updated by a single source. The tool is Data Relationship Management (DRM).

Since the focus of this article is metadata management, after all, how can ERPi specifically help an organization maintain governance over its metadata? As detailed throughout the article, the management of hierarchies is an ongoing process with frequent changes and updates to business functions and organizations. Certain dimensions may also be more dynamic than others; an Employee or Position dimension may require substantially more effort to maintain than an Entity or even Account dimension. Therefore, manual file manipulations may become too cumbersome and time consuming for system administrators. Additionally, as a best practice, a unique prefix or suffix may be used for each dimension to ensure uniqueness across all other members. So, it is crucial that consistency is created throughout the life of the application. Well, ERPi, through the use of metadata mapping rules, extracts dimension members and particular properties from the source system and loads them directly into the target application. A built-in task scheduler can then be configured to run updates as frequently as every hour or as infrequently as once per month.

When defining metadata mapping rules within ERPi, there are several features that strive to streamline the process and facilitate tight metadata governance. Rules can be established for each of the required dimensions based on the EPM application type (i.e. Account, Entity, Scenario, Version, Period, Year, etc...) as well as custom dimensions. For each dimension, prefixes or suffixes can be defined as unique identifiers, and ERPi will ultimately assign them to the source values. Furthermore, based on the dimension, certain properties can either be sourced directly from the ERP system or defined while creating the metadata rule. See Table 1 for the member properties of the Account dimension that can be set through ERPi. In addition to these properties, “Valid for Plan Type” and “Source Plan Type” options can be selected for the Entity and Account dimensions.

### Table 1: Metadata Attributes in ERPi

<table>
<thead>
<tr>
<th>Property</th>
<th>Available Options</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Type</td>
<td>Revenue, Expense, Asset, Liability, or Equity</td>
<td>Populated from Source Account Entity</td>
</tr>
<tr>
<td>Time Balance for I/S and B/S Accounts</td>
<td>Balance or Flow</td>
<td>Set when Creating Metadata Rule</td>
</tr>
<tr>
<td>Expense Reporting</td>
<td>Expense or NonExpense</td>
<td>Based on Account Type; If Expense, then Expense; Otherwise, Set to NonExpense</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Store, Shared, Dynamic</td>
<td>Based on Base or Non-Base Hierarchy</td>
</tr>
</tbody>
</table>

Segment hierarchies define those members that are extracted from the source system, and both base and alternate hierarchies can be created in the target application. For instance, if the source ERP is PeopleSoft, different Trees and Effective Dates can be selected to represent base hierarchies and/or necessary alternate reporting roll-ups. So, if an organization has different internal reporting requirements as opposed to what is required by banking institutions, ERPi can manage and maintain both sets of hierarchies.

Data Relationship Management (DRM)

Up to this point, the methods detailed have focused on updating strictly Hyperion tools, which heavily rely on the accuracy of the source information. Whether it is flat files, PeopleSoft’s hierarchical Trees, or any other relational table, the source must be maintained separately, and manual updates, file manipulations, or mapping rules are used to integrate. However, Oracle offers a central hub where database/reporting structures are maintained and validated before moving throughout the entire enterprise. Business Intelligence (BI) solutions, EPM applications, data warehouses are all maintained and updated by a single source. The tool is Data Relationship Management (DRM).

User security, or Permissions, is an important element to manage in DRM. In DRM, users access hierarchy nodes and their properties through Node Access Group Assignments. Individual users can be assigned to groups, which grant permissions to specific nodes and align with functional areas of an organization. Any node that is
not granted access is not displayed. For example, within legal entities, a group of users may only need to maintain the Information Technology structure; furthermore, similar access can be assigned to the chart of accounts structure or any other hierarchy. Access can also be granted to Versions and Property Categories. With various users and user groups accessing different hierarchies and nodes, it is important that appropriate logs are tracked. All DRM operations – data actions such as adding or removing a hierarchy or deleting a version and administrative actions such as adding a Node Access Group or updating Property Definitions – are logged in the Transaction History. The Transaction History records a timestamp, a username, the type of action performed, and any other relevant information.

Versions – sets of data categorized into hierarchies, nodes, and properties - drive all data-related operations within DRM. Typically, separate versions are used to maintain historical copies of hierarchies, nodes, and properties, create new data or model changes to existing data, and load data from different sources to be compared and/or blended. Versions also have a certain lifecycle to represent a period of time, and it is determined by Statuses. There are four different Statuses – Working, Submitted, Finalized, and Expired.

- Working – users are able to edit versions through different tools, including manual updates, Action Scripts (formerly Automators or Automator Scripts), and Blenders. Action Scripts allow users to process mass changes in an automated fashion and can be used to build alternate hierarchies or manage a departmental reorganization. When developing Action Scripts, there are two important elements to keep in mind. First, each record within the script represents a separate action to be performed and is process individually from other actions; and second, specific parameters are required in each flat file column. Blenders combine elements of two different versions into the same version, and they can be used to process adds, moves, or deletes to an existing hierarchy, create new hierarchies, and map nodes from a source hierarchy to nodes in a target hierarchy.

- Submitted – Validations are performed to ensure integrity of the data and comparison reports can be developed to determine the differences between the current version and any previous Finalized versions.

- Finalized – Exports are performed from the Finalized version to send hierarchy data to downstream systems, and can be output to flat files or relational tables. After all exports are complete and have been loaded to the destination system, all participating systems have consistent metadata as a basis for the period end reporting process.

- Expired – Versions can then be stored for possible future use in historical analyses or as an audit record

While users are adding, deleting, or updating hierarchies, nodes, and properties, Validations check for data integrity against critical, core requirements as developed by the organization. Validations can either be real-time or run explicitly by a user at the version, hierarchy, or node level. Real-time validations run at the time a change is being made and will prevent the change if the validation fails. Batch Validations (formerly Verification), on the other hand, are executed prior to pushing metadata changes downstream to subscribing systems, and verify changes that were not passed through real-time validations. See Table 2 for example validations that can be created for a Hyperion Planning application.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Real-Time or Batch?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MemberAliasLength</td>
<td>The alias of a HP dimension member must be 80 characters in length or less.</td>
<td>Real-Time</td>
</tr>
<tr>
<td>MemberNameLength</td>
<td>The name of a HP dimension member must be 80 characters in length or less.</td>
<td>Real-Time</td>
</tr>
<tr>
<td>SmartListRequired</td>
<td>A smart list value is required if the Data Type property is set to Smart List.</td>
<td>Both</td>
</tr>
<tr>
<td>UniqueNameAndAlias</td>
<td>The name and alias of a Planning dimension member must be unique across all dimensions in an application.</td>
<td>Batch</td>
</tr>
<tr>
<td>ValidForPlanCheck</td>
<td>A member of a HP dimension must valid for at least one plan type.</td>
<td>Both</td>
</tr>
<tr>
<td>SourcePlanTypeCheck</td>
<td>The Source Plan Type must refer to a valid plan type for a Planning dimension member.</td>
<td>Both</td>
</tr>
</tbody>
</table>

Table 2: Validations

Conclusion

In conclusion, it’s clear that metadata governance is extremely important in the support and maintenance of Hyperion applications. The more tightly that enterprise metadata can be controlled the less data scrubbing, cleansing, and mapping is required – which means less maintenance. With the various tools available for metadata governance, administrators are finding metadata management less strenuous and more flexible. Metadata governance is fundamental in building a system that provides one version of the truth.
Chris Chase is a Hyperion Planning, Essbase, and Financial Management professional with over 20 years of experience designing, developing, and implementing corporate IT solutions using OLAP and OLTP databases and object oriented programming languages. Chris currently serves on the board of directors of the RMOUG and has served as the leader of the Hyperion/EPM special interest group (SIG) for the past 3 years; in addition, Chris is also a member of the board of directors for the Colorado chapter of the Oracle Applications Users Group, and is also an active member of ODTUG. Chris founded The Bean Consulting Group, a Colorado based Hyperion consulting firm, in 2010; and he has remained the managing partner since then, overseeing the growth of the organization with local recruiting through the University of Denver and increasing both the local and national client base.

Become A Member For These Outstanding Benefits

- Quarterly Education Workshops
- Special Annual Training Days Rates
- Database Labs
- List Server Subscriptions
- SQL>Update Magazine

www.rmoug.org/member.htm

STILL USING SPREADSHEETS TO TRACK YOUR FINANCIALS?

We want you to enjoy your summer!

Don’t waste time wondering if your spreadsheet reports are correct or worrying that your data is still NOT coming from a single source. We provide confidence and peace of mind to a multitude of customers by making sure they have the right data at the right time.

So enjoy some sun and fun!

Contact us today, and enjoy summer with your family

720-557-1522 | 1-855-EPM-GROW | www.thebean.co
How Many Ways To Monitor Oracle Golden Gate

by Bobby Curtis, Oracle ACE

Oracle GoldenGate

In recent years, the usage of Oracle GoldenGate has grown to the point that, with the release of Oracle Database 12c, Oracle has decided to move away from Oracle Advanced Replication and Oracle Streams (Oracle 12c Upgrade Guide, Section 8.1.6 and 8.1.7). For many, this appears to be a great thing and a smart move for Oracle to provide a more heterogeneous replication product. At the same time this move raises a list of questions around setup, configuration, and monitoring within the Oracle community. The focus of this article will be the various methods available for monitoring Oracle GoldenGate in today’s enterprise.

Monitoring replication environments is critical for every enterprise. Moving data between databases with minimal downtime and maximum throughput are vital to a solidly implemented replication solution. In its simplest form, Oracle-to-Oracle, Oracle GoldenGate can be configured in a variety of ways to address many different situations. Before discussing the different configurations of Oracle GoldenGate, having an understanding of the working components will aid in understanding of the environment.

All Oracle GoldenGate environments consist of a manager, EXTRACT (capture and pump), REPLICAT (apply) and local and remote trail files. Brief explanations of these processes and trails files are provided below.

Manager – The manager process is responsible for communication between the local and remote hosts on a specific port.

EXTRACT – This is the GoldenGate capture process, is used to capture data that has changed within the environment where configured.

REPLICAT – This is the GoldenGate apply process, is used to apply data on the target environment. It reads the trail files on the system, reconstructs the DML/DDL and applies it to the target database. The REPLICAT uses dynamic SQL to create the SQL with bind variables so it can be executed over-and-over if needed.

Trail Files – Local and Remote trail files support continuous extraction and replication of database changes. Using trail files, replication processes (capture and apply) can happen independently of each other. This means that data changes can be captured, stored and later applied to the target system without concerns of losing data.

Having a basic understanding of these components will help in understanding how Oracle GoldenGate works and can be monitored. Below (Figure 1), you see how Oracle GoldenGate can be configured for an Active/Passive configuration.

Now, let’s take a look at a few different ways Oracle GoldenGate can be configured.

Architectures

Oracle GoldenGate is such a flexible replication solution that there have been six core architectures developed. These six architectures provide flexibility, transformation, and customer processing for numerous business requirements which can be grouped into these core areas:

• Business continuity and high availability
• Initial load and database migrations
• Data Integration
• Decision Support/Data Warehousing

Figure 2, provides a view at how these architectures appear for most business requirements. The first three (unidirectional, bi-direction, and peer-to-peer) provide for most business requirements around high availability and database migrations. The later three (broadcast, consolidation, and cascading) are architectures that are beneficial for data integration and decision support systems.
Although Oracle GoldenGate can be used in a wide range of architectures for many different reasons, there still appears to be a small problem. How does one monitor these environments? Let’s take a look at some monitoring solutions that are applicable for Oracle GoldenGate.

**Monitoring**

There are four ways to monitor an Oracle GoldenGate environment. The degree of interaction from an administrator varies depending on the method of monitoring selected. The four ways of monitoring Oracle GoldenGate are outlined below (Listing 1).

**Listing 1: Types of monitoring Oracle GoldenGate**

- Manual
- Script based
- Oracle GoldenGate Director
- Oracle Enterprise Manager 12c

Any monitoring solution should be robust and capable of notifying the administrator when there is a problem or perceived problem. Let’s take a look at each of these monitoring techniques in detail.

**Manual**

As with any environment the simplest way to monitor it is to do it manually. Oracle GoldenGate is no exception and when you begin working with Oracle GoldenGate it is a good approach until you are comfortable with the environment and the configurations that have been established. Over the long term however, manual monitoring of Oracle GoldenGate will become time consuming, but here are a few commands that are used to monitor Oracle GoldenGate manually.

You will need to become familiar with the parameters within parameter files that are used for monitoring and the associated commands that are available to provide information from the GoldenGate Service Command Interface (GGSCI). Most of the manual monitoring is configured when the parameter files are edited and assigned to the associated Oracle GoldenGate processes. For the EXTRACT and REPLICAT processes, the monitoring parameters are covered under the Report category. Reporting for overall lag time is configured at the manager process. These basic monitoring parameters can be seen in Table 1.

**Table 1: Reporting parameters for Oracle GoldenGate processes**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Process</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAGREPORT-MINUTES</td>
<td>Manager</td>
<td>Specify the interval at which Manager checks for EXTRACT and REPLICAT lag</td>
</tr>
<tr>
<td>LAGREPORT-HOURS</td>
<td>Manager</td>
<td></td>
</tr>
<tr>
<td>LAGCRITICAL-SECONDS</td>
<td>Manager</td>
<td>Specify a lag threshold that is considered critical, and to force a warning message to the error log when the threshold is reached</td>
</tr>
<tr>
<td>LAGCRITICAL-MINUTES</td>
<td>Manager</td>
<td></td>
</tr>
<tr>
<td>LAGCRITICAL-HOURS</td>
<td>Manager</td>
<td></td>
</tr>
<tr>
<td>LAGINFOSECONDS</td>
<td>Manager</td>
<td>Specify a basic lag threshold; if lag exceeds the specified value, Oracle GoldenGate reports lag information to the error log. If the lag exceeds the value specified with</td>
</tr>
<tr>
<td>LAGINFOMINUTES</td>
<td>Manager</td>
<td></td>
</tr>
<tr>
<td>LAGINFOHOURS</td>
<td>Manager</td>
<td></td>
</tr>
</tbody>
</table>

The parameters listed in Table 1, are defined during the initial configuration of the Oracle GoldenGate processes. These parameters can also be added after the configuration is established without impacting existing processing upon restart of the process where the parameter is added. The parameters also provide output to the Oracle GoldenGate error log (ggserr.log) when established thresholds are exceeded.

When you combine these parameters with commands that are run from the GGSCI command prompt, the administrator can gain valuable insight into what Oracle GoldenGate is doing. GGSCI commands that are valuable for manual monitoring of Oracle GoldenGate are listed in Table 2. These commands provide a range of information for each of the processes. The basics commands that can be run from GGSCI are: INFO, SEND and STATUS commands. Table 2 provides more details around these commands.

**Table 2: GGSCI commands for Oracle GoldenGate**

<table>
<thead>
<tr>
<th>Command</th>
<th>Process</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFO MANAGER</td>
<td>Manager</td>
<td>Determine whether or not the Manager process is running. If Manager is running, the port number is displayed. This command is an alias for STATUS MANAGER</td>
</tr>
<tr>
<td>STATUS MANAGER</td>
<td>Manager</td>
<td>Retrieve the status of the active Manager process or to retrieve dynamic port information as configured in the Manager parameter file</td>
</tr>
<tr>
<td>SEND MANAGER</td>
<td>Manager</td>
<td></td>
</tr>
<tr>
<td>INFO EXTRACT</td>
<td>EXTRACT</td>
<td>The status of EXTRACT</td>
</tr>
<tr>
<td>LAG EXTRACT</td>
<td>EXTRACT</td>
<td>Determine a true lag time between EXTRACT and the data source. LAG EXTRACT calculates the lag time more precisely than INFO EXTRACT because it communicates with EXTRACT directly, rather than reading a checkpoint position in the trail</td>
</tr>
</tbody>
</table>
This brings up the point of using scripts to monitor an environment. Using scripts is a time tested approach to managing any environment. Administrators tend to use scripts more in a Unix/Linux environment to monitor a wide range of manual tasks. For an Oracle GoldenGate environment, there is no reason to shy away from this approach. All the commands that are accessible from the GGSCI command prompt can be passed in shell scripts, minimizing the amount of typing administrators perform. Using scripts to monitor an Oracle GoldenGate environment provides an administrator with the following benefits:

- Write once, use multiple times
- Not having to memorize a list of commands
- Portability between environments
- Scripting Oracle GoldenGate commands can save an administrator time when trying to resolve issues within an environment.

An example of scripting an Oracle GoldenGate command is in Listing 2.

### Listing 2: Oracle GoldenGate command in shell scripts

```bash
#!/bin/sh
#info_all.sh
export GGS_HOME=/opt/oracle/product/ggate
$GGS_HOME/ggsci <<EOF
info all
exit
EOF
```

Using Oracle GoldenGate commands in a shell script is much like using the SQL*Plus interface from a shell script. Any command that is passed while inside of the end-of-file (EOF) indicator will be executed through the GGSCI utility. This approach simplifies the interaction between administrators and the Oracle GoldenGate environment.

Another benefit to using scripts for monitoring an Oracle GoldenGate environment is when the administrator needs to tune the environment. Administrators spend a fair amount of time tuning their database environments and Oracle GoldenGate is no different. A question often asked is "How much memory is GoldenGate consuming"? Using scripts, the amount of memory that GoldenGate consumes can be quickly determined. Listing 3 provides a working script that can be used for memory monitoring at the operating system level.

### Listing 3: Memory monitoring of Oracle GoldenGate

```bash
PROCESSES="mgr ggcmd extract replicat"
for GGPROCESS in $PROCESSES
do
  FLAG=`ps -ef | grep $GGPROCESS`
  if [ $_ ]
    then
      "No GoldenGate Process Found"
    else
      ps -C $GGPROCESS -o rss | awk '{print $2/1024, "MB", $12}' | sort -k 2
  fi
done
exit 0
```

As outlined in Table 2, there are a number of commands that Oracle GoldenGate provides an administrator to monitor the environment. Once the administrator becomes comfortable with the commands, identifying and resolving what is going on in the environment can be done quickly. Learning the commands and issuing them numerous times can become time consuming and slow an administrator’s ability to deal with issues on a timely basis. This brings up the point of using scripts to monitor an environment.
Using scripts, an administrator can do more with Oracle GoldenGate than what is expected. Between using scripts for operating systems and GoldenGate process monitoring, an administrator can ensure ready access to information and can even have outputs emailed to them on a regular basis using a mail utility.

Although scripts provide a powerful way to keep up with Oracle GoldenGate, Oracle provides a few graphical user interface (GUI) tools to aid administrators in monitoring.

**Oracle GoldenGate Director**

One such monitoring tool that fits the graphical user interface (GUI) model for Oracle GoldenGate is the Oracle GoldenGate Director. Oracle GoldenGate Director is a multi-tier, client-server application that allows you to manage Oracle GoldenGate instances from a remote client. Figure 2 outlines the architecture of the Oracle GoldenGate Director application.

![Figure 2: Oracle GoldenGate Director Architecture](image)

It is clear from looking at Figure 2, Oracle GoldenGate Director has many moving parts. On the client side there is the Oracle GoldenGate Director Client (Client). The client provides a GUI interface for managing Oracle GoldenGate instances. The client can be run on any system that supports Java, drag-n-drop manipulation, menus, toolbars and other desktop functionality expected within an application.

On the middle tier, Oracle GoldenGate Director Server is run and managed by an Oracle WebLogic Server Domain. The server consists of the following applications:

- Oracle GoldenGate Director Server application
- The monitor agent

These applications use a central repository to store information about users, groups, graphical diagrams created by users, consolidated events and other items to help manage Oracle GoldenGate. The components working together, gives an administrator a flexible and robust way to manage Oracle GoldenGate without the need to log into the servers hosting Oracle GoldenGate.

An added benefit of using Oracle GoldenGate Director is that the parameter files associated with Oracle GoldenGate processes can be edited without access to the servers. Interaction with Oracle GoldenGate is handled through the client interface and provides a wide range of flexibility to the instance being monitored. Email alerts and metrics can be assigned as well.

**Oracle Enterprise Manager**

The last way to monitor Oracle GoldenGate is to use Oracle Enterprise Manager 12c (12.1.0.1 or later). Oracle Enterprise Manager 12c, is quickly becoming the monitoring tool of choice for most enterprise environments. A key benefit of monitoring with Oracle Enterprise Manager 12c is the architecture supports plug-ins. As plug-ins are developed for various products they can be deployed to the Oracle Management Server and to the Enterprise Management Agent quickly, enabling a scalable monitoring approach. With Oracle GoldenGate, the plug-in can be deployed to the management agent and then monitoring of Oracle GoldenGate environments can be implemented.

Although Oracle Enterprise Manager 12c provides plug-ins for Oracle GoldenGate, there are a few things that an administrator needs to understand before implementing monitoring for Oracle GoldenGate. Monitoring Oracle GoldenGate requires the usage of the Java Agent (JAGENT). The JAgent provides access to the Java Message Exchange for reporting through the management agent to Oracle Enterprise Manager. The JAgent is turned on by using the parameter ENABLEMONITORING in the Oracle GoldenGate Manager parameter file. Once monitoring is enabled, you will see the JAGENT process within GGSCI when issuing an INFO ALL (see Figure 3).

![Figure 3: JAGENT Listed](image)

Once the JAgent is enabled and running, an administrator needs to make sure that the Oracle GoldenGate plug-in has been deployed to the Management Server and the Management Agent. This can be checked through the extensibility menu within Oracle Enterprise Manager 12c (Setup -> Extensibility -> Plug-Ins). Once on the Plug-Ins page, an administrator can find the Oracle GoldenGate plug-in under the Middleware drop down. (Figure 4).

![Figure 4: Oracle GoldenGate Plug-In](image)

From the Plug-In page, the administrator can see that the plug-in for Oracle GoldenGate is deployed on the Management Server and to one Management Agent. Once the plug-in is deployed, Oracle Enterprise Manager 12c can begin to monitor Oracle GoldenGate.

With the JAGENT started, plug-ins deployed on the Management Server and Management Agent, access to the Oracle GoldenGate page within Oracle Enterprise Manager 12c (Targets -> Oracle GoldenGate) (Figure 5) will be allowed.

From the Oracle GoldenGate Page, an administrator can see all the vital statistics for their Oracle GoldenGate environment (Lag, Lag Trend, Total Operations, Delta Operations, and Delta Operations Per Second). Using Oracle Enterprise Manager 12c to monitor Oracle GoldenGate is a nice and clean way to monitor the environment.
Combination

Besides the JAGENT, Oracle Enterprise Manager 12c, provided another way to monitor Oracle GoldenGate. Unlike using this JAGENT, this solution is a combination of the scripting approach and using Oracle Enterprise Manager 12c. By simply using Metric Extensions within Oracle Enterprise Manager 12c, monitoring of Oracle GoldenGate can be achieved at the host level. Let’s take a look at how this monitoring works.

In order to use the combination approach, you will need to have a script that is run at the operating system level to check on the status of the processes. In Listing 4, you will find a simple script that allows you to check the status Oracle GoldenGate from the command line using Perl.

Listing 4: Perl script used in monitoring

```
#!/usr/bin/perl -w
#
# use strict;
# use warnings;
#
# Static Variables
my $gghome = "/oracle/app/product/12.1.2/ggate";
#
# Program

my @buf = `$gghome/ggsci << EOF
info all
EOF`;

foreach (@buf) {
  if (/EXTRACT/||/REPLICAT/) {
    s/\s+//g;
    print $_."\n";
  }
}
```

The output from Listing 4, will provide you with a pipe (|) delimited list of Oracle GoldenGate processes that can be used for monitoring (Figure 6). Once you have the output in the format desired; Oracle Enterprise Manager 12c can be used to extend monitoring to the metric level.

With the output needed for monitoring, a metric extension needs to be created within OEM and deployed to the target where monitoring is desired.

To create a Metric Extension, access the Metric Extensions framework (Enterprise -> Monitoring -> Metric Extensions) within Oracle Enterprise Manager 12c (Figure 7). From here Metric Extensions can be developed, tested and deployed to various targets within the monitoring framework.

Once a Metric Extension has been established and deployed, the new metrics for monitoring Oracle GoldenGate can be found under the All Metrics for the associated target (Figure 8). As Oracle Enterprise Manager 12c executes the pre-defined Perl script, Oracle Enterprise Manager can evaluated the metric and notify, via notification rules, the administrator of any potential issues with Oracle GoldenGate.

The combination approach to monitoring Oracle GoldenGate from within Oracle Enterprise Manager 12c is a bit more texted based; however, metrics can be monitored more closely and frequently. Compared to using the JAGENT for monitoring, the combination approach provides a good grasp on whether or not the Oracle GoldenGate is up while avoiding configuration concerns associated with the JAGENT.

**Bugs**

Throughout this article, we have looked at four different ways of monitoring Oracle GoldenGate. With any monitoring solution,
there are bugs that will be found and could possibly cause false positives in the monitoring environment. There is really no way around these bugs other than either apply the patch for bugs identified or implement workarounds.

**Manual** - most bugs are actually associated the application software. With Oracle GoldenGate, many of the bugs are documented and can be looked up in My Oracle Support (MOS). A good starting point is MOS Note ID: 1298817.1

**Scripts** - most bugs that arise are due to end user errors within the scripts. Often times these types of errors can be minimized by code reviews and improvements of scripts used for monitoring.

**Oracle GoldenGate Director** - has a few bugs! Many of these bugs have been documented and can be looked up in My Oracle Support (MOS). A good starting point is MOS Note ID: 1298817.1.

**Oracle Enterprise Manager 12c** - a few bugs have been identified with the plug-in. Oracle has recommend that any monitoring with Oracle Enterprise Manager 12c should be on a version of Oracle GoldenGate 11.2.1.0.10 or later. One of the big bugs with monitoring with Oracle Enterprise Manager 12c has to deal with the datastore and Berkley database (MOS Note Id: 1495998.1).

No solution is 100% reliable and will still require some form of manual interaction. As with any application, there will be bugs. In the monitoring approaches identified, as with any application, there will be bugs.

**Summary**

Each approach to monitoring has its merits and its downsides; however, picking a monitoring solution is critical to how you will manage your Oracle GoldenGate environment. As discussed, there is a wide range of monitoring approaches depending on the level of control and time spent monitoring you desire. In the end, the primary goal is to keep Oracle GoldenGate running at maximum capacity while minimizing effort.

Bobby Curtis is an Oracle ACE, Senior Technical Consultant and member of various Oracle User Groups. He specializes in database monitoring and data integration technologies, both aimed at making usability simpler and easier. He is currently working as a Senior Technical Consultant focused on implementations and migrations of scalable databases while providing monitoring solutions for these environments. Bobby is a current member of the Independent Oracle User Group (IOUG), Oracle Development Tools User Group (ODTUG), Georgia Oracle User Group (GOUSER) and the Rocky Mountains Oracle User Group (RMOUG). He lives with his wife and three kids in Douglasville, GA. Bobby can be followed on twitter at @dbasolved and his blog at http://dbasolved.com.
Meet Your Board

John Peterson
President
E-mail: President@rmoug.org

Vince Giasolli
Vice President
Newsletter Director
E-mail: VicePresident@rmoug.org
NewsletterDir@rmoug.org

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

Ron Bich
Treasurer
E-mail: Treasurer@rmoug.org

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

Andreas Katsarlis
Marketing Director
E-mail: MarketingDir@rmoug.org

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

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

Chris Chase
SIGs Director
E-mail: SigsDir@rmoug.org

Bob Mason
Education Director
Scholarship Director
E-mail: EducationDir@rmoug.org
John Jeunnette  
Training Days Technology  
Vendor Support  
Board Member Emeritus  
E-mail: TrainingDaysDir@rmoug.org

Kumar Annamnidu  
Board Member At Large  
E-mail: MembersAtLarge@rmoug.org

Kellyn Pot’Vin  
Training Days Director  
Non-Voting Board Member Emeritus  
E-mail: TrainingDaysDir@rmoug.org

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

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

Mark James  
Board Member At Large  
E-mail: MembersAtLarge@rmoug.org

Heidi Kuhn  
Executive Director  
Voice Mail: (303) 948-1786  
Fax: (303) 933-6603  
E-mail: Admin@rmoug.org

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

Peggy King  
Board Member Emeritus  
E-mail: BoardME@rmoug.org

Pat Van Buskirk  
Board Member Emeritus  
E-mail: BoardME@rmoug.org
<table>
<thead>
<tr>
<th>Date</th>
<th>Organization</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/18/14</td>
<td>RMOUG</td>
<td>Board of Directors’ Meeting - TBA</td>
</tr>
<tr>
<td>7/14/14</td>
<td>RMOUG</td>
<td>SQL&gt;Update Call for Articles &amp; Photos</td>
</tr>
<tr>
<td>7/16/14</td>
<td>RMOUG</td>
<td>Board of Directors’ Meeting - TBA</td>
</tr>
<tr>
<td>8/8/14</td>
<td>RMOUG</td>
<td>RMOUG QEW at Elitch Gardens</td>
</tr>
<tr>
<td>8/15/14</td>
<td>RMOUG</td>
<td>SQL&gt;Update Article &amp; Photo Deadlines</td>
</tr>
<tr>
<td>8/19/14</td>
<td>RMOUG</td>
<td>Board of Directors’ Meeting - TBA</td>
</tr>
<tr>
<td>8/21/14</td>
<td>NoCOUG</td>
<td>Summer Conference</td>
</tr>
<tr>
<td>9/14</td>
<td>RMOUG</td>
<td>Board of Directors’ Meeting - TBA</td>
</tr>
<tr>
<td>9/15/14</td>
<td>RMOUG</td>
<td>SQL&gt;Update Publication Date</td>
</tr>
<tr>
<td>9/28-10/2/14</td>
<td>Oracle</td>
<td>Oracle Open World, Moscone Center, San Francisco, CA <a href="http://www.oracle.com/openworld">www.oracle.com/openworld</a></td>
</tr>
</tbody>
</table>

Please note dates are subject to change. For the most current events calendar visit our website at www.rmoug.org.

---

**SPONSOR A QUARTERLY EDUCATION WORKSHOP AND RECEIVE A HALF PRICE AD IN SQL>UPDATE**

<table>
<thead>
<tr>
<th>Breakfast</th>
<th>Discounted Ad Rate</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4 Page</td>
<td>$350.00</td>
<td>$175.00</td>
</tr>
<tr>
<td>1/2 Page</td>
<td>$350.00</td>
<td>$312.50</td>
</tr>
<tr>
<td>Full Page</td>
<td>$350.00</td>
<td>$500.00</td>
</tr>
<tr>
<td>Inside Cover</td>
<td>$350.00</td>
<td>$625.00</td>
</tr>
<tr>
<td>Back Cover</td>
<td>$350.00</td>
<td>$750.00</td>
</tr>
</tbody>
</table>

Join Us August 8, 2014 at the Elitch Gardens!

More Details www.rmoug.org

---

**Tell Us About Yourself**

Join us in sharing your Oracle experiences with other RMOUG members!

Tell us about your life, your job, or share your amusing Oracle anecdotes, tips and secrets!

Share your favorite photos with our members and have the chance to be published on our cover!

Prior to each issue, we ask members to present their favorite photo, regional to the Rocky Mountains and not previously published. Each issue has a theme - Spring, Summer, Fall and Winter - and we need a minimum format of 300 dpi at least 8.75” x 11.25”. So turn your camera vertically and imagine the words RMOUG SQL>UPDATE across the top!

Please submit all material to NewsletterDir@rmoug.org
Index To Advertisers

Arisant .................................................................................. 2
The Bean Consulting Group ............................................... 21
Confio ................................................................................... 27
DBAK .................................................................................... 2
Quarterly Education Workshop Sponsorship ..................... 31
RMOUG Membership .......................................................... 21
SQL>Update Opportunities ................................................. 30
Training Days 2015................................................................. 32

Reach A Targeted Oracle Audience • Advertise Now!

A full page, full color ad in RMOUG SQL>UPDATE costs as little as
70 cents per printed magazine and even less for smaller ads.

Contact NewsletterDir@rmoug.org

Quarterly Education Workshops
August 8, 2014

Join us for our next Quarterly Education Workshop in August for presentations at Elitch Gardens in Denver. Educational workshop sessions will be held in the morning, then invite your friends and family to join us for lunch at the Pavilion. Spend the rest of the day enjoying the theme park.

Members get in free! $15 each for friends/family passes

If you or your organization are interested in partnering with RMOUG to host an upcoming meeting, or to submit an abstract for presentation, please contact

Dan Hotka, Programs Director at ProgramsDir@rmoug.org

Watch RMOUG’s Web Page for August Training Topics www.rmoug.org
Tracks
- Application Development
- Business Intelligence
- Database Administration
- DBA Deep Dive
- Database Tools of the Trade
- Hyperion
- Middleware
- Professional Empowerment

Highlights
- Two full days of training with a bonus half-day of two-hour deep dive sessions
- More than 100 technical sessions
- Full exhibition
- Networking opportunities to meet and talk with Oracle ACE Directors and ACEs, product managers from Oracle, and other Oracle professionals
- Encore Women in Technology (WIT) panel

What Participants are Saying about RMOUG Training Days
- “Best bang for the buck.”
- “Excellent value for the amount of knowledge obtained.”
- “RMOUG Training Days punches well above its weight. Long-standing volunteers use their networks of connections to procure some very good content.”
- “RMOUG Training Days is a significant event on my Oracle Technology annual calendar.”
- “RMOUG Training Days was a great motivator. The speakers sparked interest and motivated me to start learning again.”

Interested in becoming a sponsor or exhibitor? Please contact Veronica Snow at 910.452.0006