Oracle Scene

A UK Oracle User Group publication

The way forward

Extending our E-Business Suite with Hyperion Planning

To gather statistics or not to gather statistics?

Top reasons to attend UKOUG Conference Series events

Issue 37 | Spring 2009 | £10.00 | www.ukoug.org

AN INDEPENDENT PUBLICATION NOT AFFILIATED WITH ORACLE CORPORATION

Welcome

Mark Rittman
editor@ukoug.org

Hello and Welcome!

Welcome everyone to this first Oracle Scene for 2009. This edition heralds a number of changes in the editorial team, with myself taking over as editor after three years of sterling service from Ophelia Dodds. I'm glad to say though that Neil Jarvis and Giovanni D'Alessio have stayed on as deputy editors, looking after technology and applications respectively, and I'd like to thank them together with Iyisha from the UKOUG for helping me deliver this issue of the magazine.

It's also "all change" for Oracle Scene this year in that we'll now be publishing three times a year, in March, July and November, and of course you can access electronic versions of Oracle Scene on the UKOUG website should you wish to download and read it on your laptop. As always, we're looking for authors on subjects spanning technology, applications and business management, so if you're interested in writing for Oracle Scene, drop me an email at editor@ukoug.org and I'll be happy to offer any advice.

This Edition

This edition of Oracle Scene has some excellent articles, going from statistics gathering in the database, through to service-orientated architectures and how to create the right controls in your business. From a technology perspective, Hugh Griffiths takes us through a SOA Case Study, whilst Robert Mycroft starts a two-part article on Using Services in RAC.

Fresh from a very successful presentation at conference, Julie Cave talks about Extending our E-Business Suite with Hyperion Planning, whilst in a topical article, Tracey Bleakley talks about ERP in a downturn.

Again, we had a massive number of submissions for this edition, and thank you to all of you who submitted articles that were held over – in most cases this was more down to creating a balance of articles in each edition and we'll look to run the best of those held over in future editions. We're also keen to hear from new as well as established authors, so if you're interested in writing for us then contact myself or Iyisha for details.

Well that's it for me, for this edition. Thanks again to Iyisha and the editorial team for their help in getting this edition together, and I hope you enjoy the contents. I hope all of you have a safe and prosperous 2009, and I'll see you all again in July, for the next edition of Oracle Scene.

Contents

Oracle Scene Editorial Team

Editor
Mark Rittman, editor@ukoug.org
Deputy Editor, Applications
Giovanni D'Alessio, deputy_apps@ukoug.org
Deputy Editor, Technology
Neil Jarvis, deputy_tech@ukoug.org
Marketing Assistant
Iyisha Rocke, IRocke@ukoug.org
Advertising & Subscriptions
For advertising/subscription information,
see www.ukoug.org or contact
opportunities@ukoug.org

UKOUG Office

UK Oracle User Group, User Group House,
591-593 Kingston Road, Wimbledon,
London, SW20 8SA.
Tel: +44 (0) 20 8545 9670
Fax: +44 (0) 870 9000 335
Email: info@ukoug.org
Web: www.ukoug.org

Production

Designed and produced by Anne Kotan
Tel: +44 (0) 1273 833 923
Email: anne_kotan@hotmail.com
Web: www.anne-kotan.moonfruit.com

Printed by Woking Print & Publicity
Tel: +44 (0) 1483 884 884
Web: www.wokingprint.com

The views stated in Oracle Scene are the views of the author and not those of the UK Oracle User Group Ltd. We do not make any warranty for the accuracy of any published information and the UK Oracle User Group will assume no responsibility or liability regarding the use of such information.

All articles are published on the understanding that copyright remains with the individual authors. The UK Oracle User Group reserves the right, however, to reproduce and article, in whole or in part, in any other user group publication.

The reproduction of this publication by any third party, in whole or in part, is strictly prohibited without the express written consent of the UK Oracle User Group.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates, used under license. This publication is an independent publication not affiliated or otherwise associated with Oracle Corporation. The opinions, statements, positions and views stated herein are those of the author(s) or publisher and are not intended to be the opinions, statements, positions, or views of Oracle Corporation.

Business and Management

Applications

Technical

The Guide to Fusion Development

Top Tip

Oracle News

Event News

UKOUG News

ERP in a Downturn – How to Maximise Value when Implementing Applications

by Tracey Bleakley, Edenbrook

In the current economic climate, it is more important than ever for systems projects to realise shareholder value and return on investment quickly. Unlike previous downturns, investment in Enterprise Resource Planning (ERP) applications and their underlying technologies has not suffered from blanket IT budget cuts, as competitive pressures now force companies to look for increasing operational efficiencies in their systems.

However, what is increasingly important is that projects are planned carefully to deliver immediate and quantifiable benefit, and that the resulting solutions can be supported in a cost-effective manner. Below are a series of measures designed to ensure projects deliver value for money.

Ensure Cross-Functional Coordination

When planning improvements to existing ERP systems, it is important to involve every group of users, whether they are an internal department or an external group – such as customers and suppliers. Due to the integrated nature of ERP systems, it is no longer possible to carry out, for example, a 'Finance' project without consulting the HR, sales, procurement, marketing, manufacturing and training teams who will also use the same system.

Coordination is one of the key success factors of ERP projects and should be planned in from day one, by designating an operational management group, a project steering committee, a design authority and user groups. These groups should meet on a regular basis to ensure that the operational and project needs of the system are being planned and actioned in the most effective way, for all stakeholders.

Success in this area will enable rapid, focused decision making, reduce project delays, and reduce testing and rework due to unforeseen conflicts with other departments. Whilst this may seem like an unnecessary overhead, time spent in well planned meetings making informed decisions will save on unplanned and costly delays later. Failure could result in implementation delays and organisational conflicts, while piecemeal implementations can negate the very purpose of an integrated package.

Implement a Method to Avoid Scope Creep

Once the cross-functional groups are in place, the project scoping and business case activities should be coordinated to make sure that every stakeholder understands and agrees with the project scope up-front. Changes to scope not only affect project timescales and costs, they can also impact benefits realisation, and therefore should only be authorised when there is a clear and well researched need.

Managing against a well-defined milestone plan and scope increases the probability of a project being delivered on time and to budget. Making rapid decisions at the proper levels in a well managed change control process (with checkpoints for authorisation and levels of research to be presented) avoids unnecessary scope creep and keeps implementation efforts on track.

Project Governance is essential (through steering groups and design authorities), as is a clear and timely decision-making and escalation procedure.

Implement Clear Project/ System Ownership and Transference of Knowledge

In order to minimise support costs and maximise the benefits from the technical expertise of external consultants, the project should be seen as an opportunity to build new capabilities internally through knowledge transfer.

This process should be carefully planned to transfer all necessary skills to affected employees by the end of the project. It is important to make sure that all individuals involved in maintaining the system and data are aware of the scope of their responsibilities, and are suitably trained during the project to undertake them in the most efficient way.

Make Time for Change Management

Change management is the process by which the organisation learns how to carry out the new roles and processes that the new system has been designed to support. If a project aims to improve operational effectiveness or reduce costs, then by implication something will need to change – be it the reorganisation of teams and the creation of new or different roles and expectations, or learning how to accomplish the same task in a different way.

Left to chance, it is unlikely that the organisation will adapt to the new systems in such a way that the required benefits will be achieved. In fact, resistance to change can be so severe that the project itself may fail to deliver anything.

Success in change management will make sure that the organisational and process changes are in place at exactly the point needed to enable the systems element to be seamlessly implemented. It should also reduce organisational disruption, change requests (from users not consulted on the project), user dissatisfaction with the system and the need for retraining. Failure could mean that the project is abandoned or simply ignored.

> "Success means that the benefit owners know what they need to accomplish by when, and are supported by the project and the user base to achieve their targets."

Focus on Communication

Tell people what you are doing and why! Resistance to change often comes from fear of the unknown, and a well planned communication strategy with tailored emails, web-site updates and roadshows is a simple and cost effective way to make sure everyone is onboard with the project.

Communication can also stimulate new ideas from the user base to bring additional benefits. Success means gaining everyone's commitment and support. Failure means that an opportunity is lost to make the users feel involved with the project and the ongoing improvement plans.

Involve Human Resources and In-House Change Management Expertise

Lack of user training and failure to understand how changes to ERP systems may impact on existing business processes are both classic factors in failed projects. The need to train users in both the systems and the new business processes is one of the key elements of success for an ERP project, however it could be a mistake to assume that this will be a costly exercise requiring large amounts of external assistance.

It can be useful to involve in-house HR/training functions in the project from an early stage so that these activities can be properly planned and coordinated, and many of the key users on the project may be able to assist in training. Also, look at automated tools such as UPK (User Productivity Kit) to enable repeatable desktop training. This will reduce employee travelling expenses, and allow training to be updated and repeated at regular intervals at a much reduced cost.

Success means having a plan for initial and ongoing training to cover the entire user base at a reasonable cost. Failure is scrimping on initial training and ignoring the need to budget for ongoing updates and refreshers – the effective use of the system and thus the realisation of benefits can only be achieved by trained users who know how to carry out the new processes!

Implement Performance and Benefits Measurement throughout the Project

When a project gets underway, it is easy to become embedded in its day to day management, and to forget the initial timescales and benefits that were projected from the outset. However, it is critical that projects deliver on their business case, and that quick-wins materialise as and when predicted.

However, this will not happen unless every benefit has a clearly defined owner, date and success criteria (i.e. what should happen when the benefit is realised). A performance and benefits measurement system could simply be an intranet page with a series of benefits against dates, owners and status. This will provide a way to track project work, identify gaps and deficiencies in performance, and recommend any additional work or resource needed to realise the benefit.

Success means that the benefit owners know what they need to accomplish by when, and are supported by the project and the user base to achieve their targets. Failure means that the project is implemented with no clear way of knowing if the benefits have been realised and why.

Align IT and the Business

It is important that an ERP project is not viewed as an IT project, as it can be easy to foster isolation between IT and business divisions, and misalignment between business strategy and IT strategy.

This can be avoided by leading each project with a well-defined business-centered need, and by ensuring that the project is sponsored, steered and managed by representatives from both IT and the business. Ideally, the teams will work together to develop long-term roadmaps so that the business and IT strategies are going in the same direction and remain flexible enough to support changing business needs.

Success means that the system supports the organisational changes, new processes and business needs and that the project benefits are realised. Failure means that the new system does not support the new processes, the benefits are not realised and the system is under-utilised, leading to costly ongoing development, or distrust between the teams.

Understand the Difference between ERP Software Configuration and Customisation

ERP packages may be configured to more closely fit a company's structure, business practices and workflow. Configuring the system can involve making compromises, given the adaptability of the software and the effort involved. This fine-tuning of the standard system is a key process in the implementation and requires translating business processes into appropriate parameter settings.

Customisation on the other hand is where the core system functionality does not meet the requirements of the client, and therefore the underlying system code is changed in some way to make the software fit the business. There is usually some element of customisation in any project – often around 20% (as one size usually does not fit all), however it is important to balance the need to customise with increased potential maintenance costs, difficulty in implementing upgrades and fixes and increased training needs.

Success means that the business is fully supported by an ERP system that is upgradeable, maintainable and has a reasonable (and predictable) cost of ownership. Failure means the business is either compromised by a 'vanilla' system (meaning that lots of manual off system processes are necessary), or it cannot

> "Implementing an ERP system is a careful exercise in strategic thinking, planning, and negotiating with departments and divisions ..."

support or upgrade a highly (and unnecessarily) customised system.

Do not Scrimp on Project Management

Implementing an ERP system is a careful exercise in strategic thinking, planning, and negotiating with departments and divisions – all of which require careful project management and methods.

Given that (as above) most customers customise around 20% of the system to meet the business need, planning the development effort is critical. Also bear in mind that the combination of hardware and software as well as numerous organisational, human and political issues all combine to make ERP projects complex, requiring significant project management skills.

Success means the project is delivered on time and to budget and realises the predicted benefits. Failure can not only impact the current project, but could compromise the relationship with the users and departments to such an extent that future projects could be negatively affected.

Consider a Phased Implementation Approach

A phased implementation approach is often less risky and not as dependent on the people involved. The different implementation approaches available are "Big Bang", "Phased" and "Parallel", and the choice is usually dependent on the project, timescales and budget.

The "Big Bang" approach is normally the most ambitious and difficult,due to the amount of users and departments involved, resistance to change, the amount of business processes that need remodelling, and the integration testing required to implement an entirely new system.
This approach is often appropriate for either small companies needing their first ERP system or new business start-ups.
The "Phased" approach often works better for large companies with existing systems.

Success means that the approach chosen delivers the project successfully without delaying any of the benefits and quick wins unnecessarily – this is often a fine balancing act.

Consider Parallel Running for Critical Elements (such as Payroll)

A "Parallel" approach ensures a fallback plan if for any reason the initial implementation is not successful, and allows a smoother transition to the new system. However, this must be balanced by the fact that maintaining data consistency between two systems introduces complexity, additional effort and cost.

This is a risk-minimising option. Success means that the cost of the additional data entry is much lower than the cost of failure (taking into account the complexity of the new system, the reliability of the migrated data and the sensitivity of the process).

Take Advantage of ERP Business Processes

ERP applications are built on best-practice processes, which are by definition designed to give companies competitive advantage. However, the level of competitive advantage realised is dependant on how organisations implement and use their ERP systems. A good rule of thumb is that if the ERP business processes fit with the current industry standard, then these should be kept, whereas those specific company developed processes that currently bring competitive advantage should ideally be replicated in the new system (ideally through configuration to minimise customisations). Once all other options have been exhausted, it is generally better to customise than to reduce the effectiveness of the business.

Each department involved should be prepared to change the existing business processes to the recommended best practices of the ERP system, because simply automating the existing processes will rarely realise the project benefits.

Selecting which processes to keep and change during ERP implementation is a challenging and daunting task; however these decisions will impact greatly on the success of the project.

Success means developing a system that will support the business and realise competitive advantage with the lowest possible overall cost of ownership.

Prioritise User Acceptance Testing

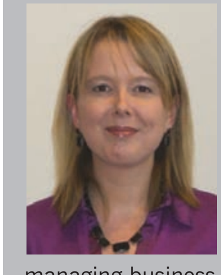
User acceptance tests are critical during the project phase, as ERP implementation success depends on the acceptance of the end-users and customers. Simply put, if the users don't like or cannot use a system, they will work around it rather than with it. Spending time here getting the system right reduces the amount of development projects post go-live and increases user satisfaction. In addition, if the system is simple and logical to use, it follows that productivity benefits will be realised and that data will be entered correctly (leading to meaningful reports and lower business intelligence costs).

Success is allowing enough time for user acceptance testing to fix all of the high priority bugs prior to go-live, with a clear plan to fix any left over 'snags' as soon as possible. Failure is skimming over user acceptance testing and finding serious issues following go-live or allowing UAT to foster scope-creep and derail the entire project timescale.

In Summary

In the current economic climate, it is more important than ever to realise the predicted value from ERP projects through sound implementation practices and the application of best practices. This article is intended to propose a number of sound business principles to ensure that such projects have all the ammunition they need to succeed – even in the toughest of markets!

About the Author



Tracey Bleakley is Head of Change Management at Edenbrook and a director of UKOUG. She has 12 years experience of designing, implementing and managing business transformation projects supported by Oracle solutions across a range of sectors including telco-media, financial services, retail and professional services.

In Control?

Top five common Oracle control issues

by James Mann and Will Drew, Deloitte

Many organisations are implementing or upgrading Oracle's E-Business Suite (EBS) to bring about a reduction in the cost of operations, effective and efficient business processes and enhanced quality of information. There are a number of challenges associated with implementations or upgrades, though one area frequently proves difficult to get right… controls.

Given the sensitive financial, customer, supplier and HR data stored within Oracle E-Business Suite, the implications of not getting the controls challenge right can be serious. The risks to organisations can range from data leakage and the associated reputational damage, through to fraud and the material misstatements of financial accounts.

It is surprising how little attention is given to application security and process controls during Oracle implementations. While some large organisations actively involve their audit and compliance teams, the majority of organisations overlook security, controls and audit requirements.

This article outlines our experiences of some of the most common control issues we have encountered while reviewing and assessing live EBS environments. These issues are unlikely to be new to Oracle security professionals or System Administrators but it is surprising how many organisations fail to manage these issues effectively – even those that claim to take security and controls seriously.

The Top Five

1. Super user Responsibilities

An area of difficulty for many organisations is the use of Oracle's out-of-the-box super user Responsibilities (access profiles). These Oracle Responsibilities, typically used by consultants and support staff, provide unrestricted access across a number of Oracle EBS modules. While the level of use may vary between organisations, the presence of users with these Oracle Responsibilities presents a risk due to the wide access they provide.

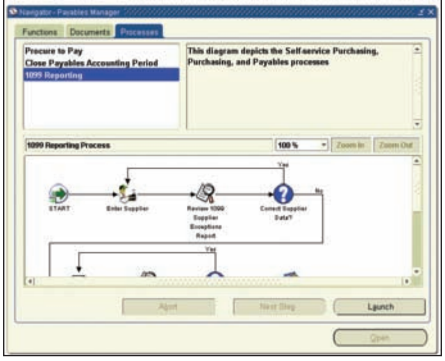
For example, the 'Payables Manager' Oracle Responsibility allows someone to create a supplier, assign their own bank details to the supplier, raise an invoice and process a payment. From the perspective of an auditor analysing your system, this would be a fundamental control weakness.

From a control perspective, no-one should have access to these Oracle Responsibilities in a live environment. A few key examples include:

- Payables Manager
- Receivables Manager
- Purchasing Super User
- Projects Super User
- General Ledger Super User
- Fixed Assets Manager
- UK HRMS Manager

Additionally from version 11.5.9, these default super user Oracle Responsibilities also provide access to functions via the 'Processes' tab within the Oracle EBS navigator. (See Figure 1).

Figure 1: Oracle navigator processes tab



Procure to Pay
Close Payables Accounting Period
1099 Reporting

This diagram depicts the Self-service Purchasing, Purchasing, and Payables processes

1099 Reporting Process

100 %

A user who has access to this tab can select any part of the process displayed in the pictorial window, and the corresponding Oracle form will open irrespective of whether the user actually has access to this function from the standard 'Functions' tab on the navigator.

We often find the super user Oracle Responsibilities used as a basis for organisations' own Oracle custom Responsibilities, and as a result this access frequently propagates through to a large number of end users in live environments. This is a significant risk because the Processes tab provides unrestricted access to end to end processes, undermining access controls and breaching the good practice of segregation of duties whilst increasing the risk of fraud.

2. Segregation of duties

Given explicit prominence by the Sarbanes-Oxley Act (2002), segregation of duties involves ensuring no single individual has control over two or more parts of a process. A simple example could be a payroll assistant who has the ability to create employees, assign salary details and enter employees onto the payroll. This level of access increases the potential for fraud, as the individual could potentially create a dummy employee assigned to a bank account they control and subsequently process fraudulent payroll payments.

In our experience, very few organisations manage this issue effectively. The root cause is often a failure to define a clear policy that sets out which activities should be segregated. This should be completed before even considering how these principles then translate into access rights in Oracle EBS.

Although management may have defined segregation of duties principles, enforcement of these principles often resides with the system administrators. These staff may know the Oracle EBS security model, but they often lack sufficient financial knowledge to fully understand segregation of duties requirements, or the implications of not correctly configuring Responsibilities to enforce these rules.

Even when segregation of duties principles have been defined and translated into access profiles, we often find a number of process failings which lead to conflicts within the system. Typical failings include:

- No consideration of segregation of duties when assigning users' additional access.
- Oracle Responsibility names not reflecting the access they provide.

- Out of date documentation that no longer details the access each Oracle Responsibility provides.

- No segregation of duties reports to help identify and monitor conflicts.

3. Inadequate system administration reports

Oracle provides a number of standard reports that enable system administrators to perform basic security checks. These include:

- Active Users report

- Active Responsibilities report

- Users of a Responsibility report

- User Profile Option Values report

- Functional Security Report request set

One failing often cited with Oracle EBS is the limitation of the standard reports, and security unfortunately is no exception. None of the reports listed above are exception reports which enable organisations to quickly identify and remediate system administration issues. It is little surprise therefore that many organisations relying on the standard reports struggle to adequately manage their user base. Organisations which most effectively manage their user base and proactively manage data quality have implemented a suite of bespoke reports.

Furthermore, the integrated nature of Oracle EBS means that several of the modules rely upon the accuracy of the user and employee information. Organisations using Oracle iProcurement and iExpenses will be aware how this information supports the approval routing of expense and purchase requisition notifications.

From our experience, effective system administration reports should address at least the following issues:

- Password expiry exceptions

- Missing user account descriptions

- User accounts not assigned to active employee records

- Employees with multiple user accounts

- Active user accounts assigned to inactive employee records

- Mismatching email addresses on an employee's HR and user account records

- User accounts with access that conflicts with the basic principle of segregation of duty

4. No auditing

By default, Oracle EBS records the username of who created and last updated a record via the standard 'Last Updated By' history. However, this information provides a limited audit trail. For example, if someone has reason to suspect several small payments had been diverted into the wrong bank account, using only the 'Last Updated By' history an organisation would not have any way of checking whether the bank account details had been temporarily changed.
The standard Oracle EBS 'AuditTrail' functionality can record a history of changes to data including; what changed, who changed it, and when it was changed. This can be useful when control breaches are identified, as analysis of numerous insider control issues have shown that had auditing been in place, the financial impact could have been reduced.

However, a common myth still prevails that auditing in Oracle EBS severely impacts system performance. This is often perpetuated because technical administrators often convince auditors that it is a "nice to have". By carefully assessing what is considered high risk, and only auditing certain tables and columns, the 'AuditTrail' functionality can be used effectively without a significant impact on performance.

Many companies will audit a few items such as salary information and bank account details but stop there, citing the performance excuse. Whilst these are high risk changes, they are unlikely to be the only ones. Any auditing strategy should be selective and effective, covering items such as supplier, customer and employee master data together with system configuration. It should also define appropriate monitoring reports that are reviewed and followed up when exceptions are identified.

5. Excessive access within support teams

Would you mind if a member of the support team looked at your salary or expenses? Have the system administrators made themselves exceptions to the corporate policy?

The assignment of excessive privileges to system administrators and use of generic accounts (e.g. SYSADMIN) is a common issue. IT staff often have free reign over the system, with managers often lacking a real understanding of what system administrators actually need access to.

In a well-controlled Oracle EBS environment, system administration staff typically only have the ability to administer users, workflow processes and module configuration. They shouldn't have the ability to enter transactions nor update sensitive master data. If asked to justify why they need transactional access you'll hear responses like "well I need to have access in order to recreate issues", "we've always had access to do that" and "I won't be able to do my job if you take that away". However, good practice suggests that support users are only assigned transactional access to the live environment when they need to resolve a problem that can not be re-created in the test and development environments.
Their access should be limited to the area concerned, removed once the problem is resolved and all activities performed are reviewed for appropriateness.

Without someone in a management position with the responsibility for casting a critical eye over the activities of the system administrators, it is likely that the team will have access well beyond the remit of their job description. Most of it probably assigned by themselves.

While these five issues are the most common, they are certainly not the only recurring issues we have seen. Insecure supporting infrastructure, security polices which are not applied to the system, sensitive data which is not scrambled in test environments and poor leavers processes are just a few further examples of other issues encountered by organisations.

> "The 'AuditTrail' functionality can be used effectively without a significant impact on performance."

> “Typically the focus will be on getting Oracle working, rather than ensuring appropriate controls are designed and operating effectively.”

The root cause

Oracle EBS is a complex system, and the number of potential configuration options can make it a challenge to control effectively. Whilst the majority of the issues outlined in this article could be classified as either a process or technology failing, an important aspect which should not be overlooked is people. Failing to allocate ownership for Oracle EBS controls is often the root cause with control issues falling in between Finance, IT and often outsourced providers. This is compounded by few individuals having the combined understanding of Oracle EBS, good practice controls and business processes to adequately ensure controls are implemented both efficiently and effectively.

However a more fundamental issue can be that an organisation lacks the culture and focus on controls. This often leads to a lack of awareness regarding control issues, and controls neither being designed up-front or assessed from a risk-based perspective. Typically, the focus will be on getting Oracle EBS working, rather than ensuring appropriate controls are designed and operating effectively. This frequently leads to a reactive approach to Oracle EBS controls, with an organisation’s auditors raising issues which could then result in a costly and time consuming re-design, and thus leaving the organisation exposed until it is resolved.

From our experience, a good place to start is to allocate ownership for Oracle EBS controls with a remit that encompasses application governance, business process controls, application security and database and infrastructure controls.

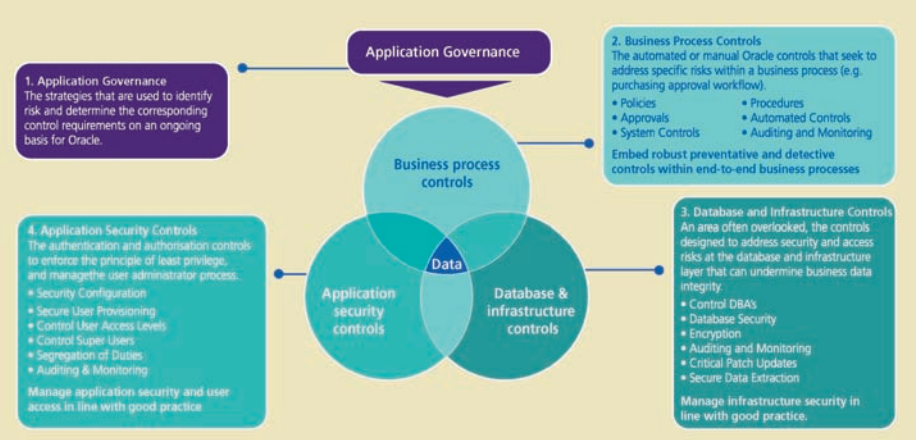
The diagram in Figure 2 shows these four key Oracle Control components. In each area, an organisation needs to address the balance between risk and controls, including the right mix of preventative, detective, automated, and manual controls suited to their organisation. It is easy to underestimate the complexity and interrelationships between the Oracle Control components. Focusing on just one area does not provide a complete control solution.

Those charged with the task of “owning Oracle Controls” might not be able to implement the changes in all areas themselves but they must communicate with both the Finance and Oracle IT staff, and importantly make sure that the interrelationships between the areas are addressed.

Conclusion

The issues presented in this article are certainly not new nor an exhaustive list, but they continue to cause control headaches for organisations using Oracle EBS. The good news is that Oracle themselves have recognised these reoccurring issues, and their recent acquisition of Logical Apps has greatly strengthened their available Oracle controls technology offering.

Figure 2: Key Oracle control components



Application Governance
Business process controls
Application security controls
Database & infrastructure controls
Data

1. Application Governance
The strategies that are used to identify risk and determine the corresponding control requirements on an ongoing basis for Oracle.

2. Business Process Controls
The automated or manual Oracle controls that seek to address specific risks within a business process (e.g. purchasing approval workflow).

- Policies
- Procedures
- Approvals
- Automated Controls
- System Controls
- Auditing and Monitoring

Embed robust preventative and detective controls within end-to-end business processes

3. Database and Infrastructure Controls
An area often overlooked, the controls designed to address security and access risks at the database and infrastructure layer that can undermine business data integrity.

- Control DBA's
- Database Security
- Encryption
- Auditing and Monitoring
- Critical Patch Updates
- Secure Data Extraction

Manage infrastructure security in line with good practice.

4. Application Security Controls
The authentication and authorisation controls to enforce the principle of least privilege, and managethe user administrator process.

- Security Configuration
- Secure User Provisioning
- Control User Access Levels
- Control Super Users
- Segregation of Duties
- Auditing & Monitoring

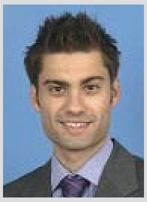
Manage application security and user access in line with good practice

This includes technology to automate segregation of duties identification and prevention and to implement auditing and granular security for configuration, master data and transactions.

However, even without this additional technology, a renewed effort is required to consider and address these common control issues. As a minimum addressing these top five control issues will be one step on the way to keeping the auditors happy and will help reduce the organisations risk exposure. Furthermore, it may even provide process efficiency savings which could reduce administration and support costs.

About the Authors

James Mann and **Will Drew** work in Deloitte’s Oracle Controls team who specialise in assessing, implementing and optimising Oracle EBS controls. This includes working as part of Oracle implementations to facilitate the inclusion of good practice security and controls. This experience is also used in performing Oracle control assessments and assisting organisations in improving and optimising their controls.



They both regularly present at Oracle User Group conferences and SIGs.

jamann@deloitte.co.uk

wdrew@deloitte.co.uk

www.deloitte.co.uk/oraclecontrols

No substitute for Practice

Throughout June I have been training the first few dozen of several hundred government employees who will shortly be using a new electronic Oracle-based Case Recording application. This follows several weeks spent writing course materials and user manuals.

Third party applications normally require a sizeable number of bespoke additions to configure in local requirements and this requires flexibility in training methods to include some non-tested functionality.

In the ideal world the training database will be a mirror image of the eventual live system but in reality this rarely happens. Where hundreds of employees are involved, staff training needs to start several weeks or even months prior to go-live which means a look-alike system won't be available. Go-live dates are always optimistic with tweaking and amending going on almost up to the final hour as new anomalies are discovered during testing and dummy runs. Implicitly, the training system will be already out-of-date by the time the first course sessions are delivered, with the gap getting wider as go-live date approaches.

Fortunately, none of the above really matters. The main objective of application training courses is to demonstrate the main functions and to provide trainees with exercises which help with screen familiarity, whilst the more complex details will be learned after go-live when the users are given the system. Even when pre-go-live versions are available for employees to practice on the majority don't take the opportunity as they never find the time. The few who make time usually get frustrated on discovering bugs which nobody will fix as they have usually already been rectified on the prototype 'live' system and there is no resource to apply every new patch to training and 'play with' systems. On countless occasions I've spent a day getting to grips with some software only to return to it a month later to discover that I've forgotten how to use it. Knowing this, I always write notes at the time to remind me how to do the difficult bits so I won't need to work them out a second time. This is common practice in training and emphasises the need for good documentation so that employees can look at the User Guides and be reminded of the instructions they receive in their courses, which may have been several weeks earlier.

Even the most timid or negative users will pick up new applications once they start using them as it only takes practice. I have made many on-site visits to supposedly struggling employees within a month of a go-live, to find them working away quite contentedly and requiring advice on no more than a few minor functions.

John McGhee is an Independent Training Consultant specialising in Oracle-based Social Care Applications. Formerly the European IT Manager of an American blue-chip company his experience includes 15 years of designing Oracle databases and implementing Oracle Financial applications. John won an Individual Excellence medal at the 2009 BCS Industry Awards.

Content supplied by:

Global Oracle Contractors Network

ORACLE contractors.com

Extending our E Business Suite with Hyperion Planning

by Julie Cave, Reading Borough Council

At Reading Borough Council (RBC) we are long term happy users of Oracle's E Business Suite (EBS), but simmering for the last few years has been our need to provide accurate mid and long term budgets for the council. Finally, Oracle appeared to have sorted out their budgeting solution with the acquisition of Hyperion and this article outlines our experience.

In 1998, RBC used GL, AP, AR, PO, CM and added iProcurement in 2003. Additionally, we have ADI plus Discoverer with NoetixViews and the GL Company for our reporting. Our original implementation was carried out by Fujitsu who continues to be our trusted partner. Like many organisations, we tried to look ahead when we invested in Oracle and included Oracle Financial Analyzer (OFA) in our licence bundle with the intention of rolling it out for budgeting later.

Our budgeting requirements are three fold: next year, mid term and long term. Next year Budgeting is easily managed using Financial Statement Generator (FSG) and Application Desktop Integrator (ADI). The challenge comes with the mid and long term (three and five years) which are complex calculations on top of next year's budget.

Since our initial implementation of EBS, Oracle has changed their strategy on budgeting at least twice. Several years ago, they recommended that those who had not already implemented OFA should look at their new product Enterprise Planning & Budgeting (EPB), but take up was slow. Through our attendance and the Local Government Special Interest Group, we had kept an eye on its maturity but everyone who had OFA kept saying EPB was not up to the job. At that time, we could wait but by early last year we had to start doing something.

We talked to Fujitsu about our options and although there were other solutions, we were keen to utilise our existing investment so were limited to either our initial OFA investment or EPB where we could simple exchange our licences 1:1. When we actually looked deep into EPB, it would do what we wanted, we could see why OFA was more popular but we were not migrating from that so decided to go with EPB, and commissioned Fujitsu to implement.

At that very point, Oracle purchased Hyperion and quickly advised that strategically they were going with Hyperion Planning. Great, it looked very good and was world class, but it was not a simple licence swap and RBC only had the agreed budget for the implementation of EPB available. Not easily put off, RBC talked again to Fujitsu and we came up with a plan based on the desires and needs of all stakeholders involved.

- RBC needed Hyperion Planning with a fixed budget.
- Fujitsu wanted experience with Hyperion.
- Oracle wanted a public sector reference for Hyperion as this was not a market they had traditionally sold into.

> "In RBC, we made people available to not only input into the process but also to learn and do a lot of the actual system population. And thanks to everyone's hard work, we completed on time and on budget."

Fujitsu were already Hyperion Enterprise users and had their own Hyperion implementation partner Paragon. They agreed to work with Fujitsu to gain experience with Oracle applications.

To keep to the budget required, lots of innovation in direct costs, hardware re-use and resourcing. In RBC, we made people available to not only input into the process but also to learn and do a lot of the actual system population. And thanks to everyone's hard work, we completed on time and on budget.

That sounds great and it is but it was not plain sailing, we learnt a lot on the way and just like we have learnt from others through the UKOUG, I hope to return the favour and am presenting our story at both the SIG and this years conference.

Our main learning points were around the many components of Hyperion Planning, how some of our calculations were too complex for the application and had to be configured within Essbase 'the cube' beneath the application. Traditionally, Hyperion has run on Windows and, although it is supported on UNIX, there was not as much knowledge around that. We needed additional hardware, with unexpected middleware requirements and although they were both at no additional cost, we have also had to take on the skills required for Oracle Data Integrator (ODI, previously Stellent) and Smartview the integration with MS Office.

Ongoing challenges are mostly around the integration with EBS. Currently, integration is down to process but we know full integration is on the product roadmap. Now, we need to maintain segment values in both systems and security rules from EBS do not flow over, so Essbase generates every possible combination. If we do utilise a new combination, it must be set up in EBS as there is no dynamic insertion for budgets. However, one of the good things about going first is that you get better access to Oracle and we are being tracked by product development who is listening to us and we believe over time these challenges will go away.

> “Accountants are happy because Smartview means they have that ‘spreadsheet’ experience...”

Ultimately, RBC would like to be able to create all new calculations ourselves and not rely on Essbase experts or have to learn that as well and we will soon start assessing the latest release of Planning which includes Calculations Manager which may help.

So, now that we have Hyperion installed, this is our process:

- Download current budget data (FSG into Excel).
- Load it into Hyperion (via Essbase load rules).
- Centrally apply inflationary elements.
- Using dataforms created during build user can view data and make adjustments.
- Once budget agreed by Cabinet, extract from Hyperion.
- Load back into EBS via budgeting spreadsheets.

Now that RBC has a working solution for mid and long term budget planning cycles, we also intend to have budget monitoring reporting in Hyperion, which provides better functionality than EBS. Centrally, we have visibility of process throughout, which is currently hidden in end user spreadsheets. Accountants are happy because Smartview means they have that ‘spreadsheet’ experience and what makes us most happy was that all this was achieved within budget.

We look forward to the planned full integration, but are very pleased we implemented when we did. The benefits outweigh the challenges and we have them now.

Oracle Hyperion Planning is a centralized, Excel and Web-based planning, budgeting and forecasting solution that integrates financial and operational planning processes and improves business predictability. Oracle Hyperion Planning provides an in-depth look at business operations and its related impact on financials, by tightly integrating financial and operational planning models. With Oracle Hyperion Planning, you can meet your immediate financial planning needs while enabling a platform for future cross-functional expansion and automated process integration.

UKOUG Conference Series Hyperion & BI 2009

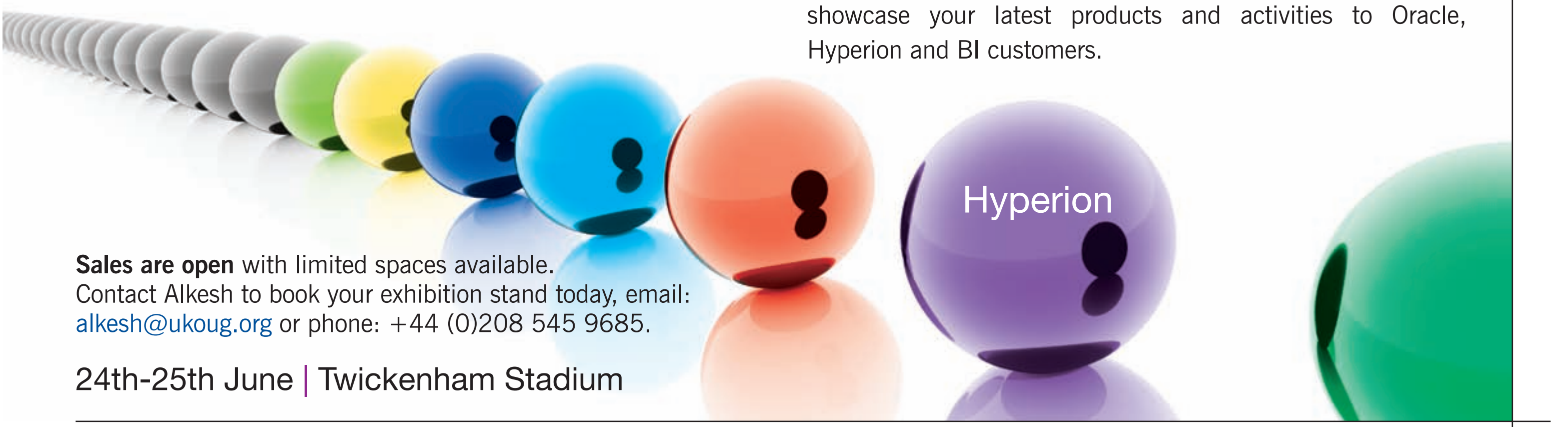
UKOUG Conference Series Hyperion & BI 2009 is the must attend event for the Hyperion, BI and Performance Management community in the UK.

The introduction of UKOUG Conference Series 2009 brings you the first independent two-day Conference representing the Hyperion and BI community in the UK since the acquisition by Oracle.

During UKOUG Conference Series Hyperion & BI 2009, you can also attend the exhibition, where delegates can meet with Oracle partners. This is the ideal place to see demonstrations of Oracle Technology and Applications.

More information about this event including registration details can be viewed at www.oug.org/hyperion

Interested in exhibiting? Do not miss the opportunity to showcase your latest products and activities to Oracle, Hyperion and BI customers.



Sales are open with limited spaces available.
Contact Alkesh to book your exhibition stand today, email: alkesh@ukoug.org or phone: +44 (0)208 545 9685.

24th-25th June | Twickenham Stadium

Compensation Workbench making life easier in HR

by Nina Cannon, iTrain Consulting Practice Manager

What is the range of your company's compensation plans? How are your company's compensation plans measured? Are they related to performance or are they standalone?

Whatever the size of your organisation, compensation plans are often quite complicated. At iTrain Consulting, we have found that Compensation Workbench can help HR departments become incredibly more efficient.

What is Compensation Workbench?

Compensation Workbench is a part of a larger suite of HCM applications that work seamlessly with Performance Management, Payroll, Human Resources and Finance.

The main functions of Compensation Workbench are as follows:

- Management and allocation of Annual Review Payments and Performance Awards.
- Bonus Awards based on targets or awarded on an ad-hoc or stand-alone basis. This is often utilised in strong sales driven environments, where short term incentives and awards are used to motivate and encourage employees.
- Blanket allocation of stock options to all the workforce or allocation to specific groups of people.
- Purchasing of additional employee benefits over and above the standard benefits such as annual leave. Examples of these benefits will vary between organisations and may include private medical healthcare insurance, additional annual leave, childcare discounts or vouchers, travel season ticket loans or discounts or retail discounts.
- The management of employee promotions such as the increase of salary or an increase to annual benefits that are received by the employee.

Under what circumstance would I use Compensation Workbench?

The following two scenarios reflect common instances that iTrain consultants encounter when meeting with HR departments and ways in which Compensation Workbench can help solve identified issues.

Scenario One

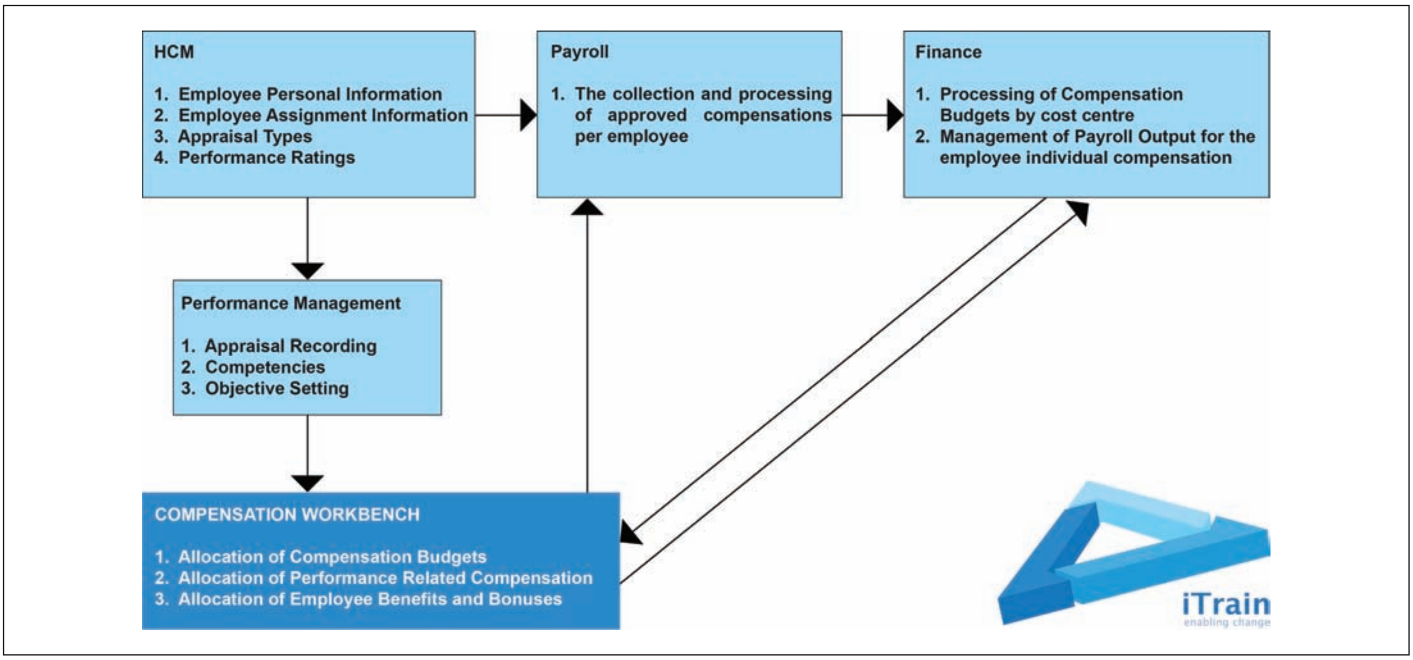
A new starter has joined your corporation recently. Within his grading, he is entitled to 4 weeks basic annual leave (20 days) and his basic salary as specified in his contract of employment. What his grade also entitles him to is £5000 a year of a selection of additional corporate benefits. Using compensation workbench, the employee can choose how to spend his entitlement – with options including Family Health Insurance Benefits and additional Annual Leave.

Scenario Two

It is the year end and every line manager in the organisation will be responsible for the completion of the annual performance review with each member of their teams.

With the support of Compensation Workbench, a suitable budget will have been allocated to each manager for the use of awarding performance related compensation. The rate of compensation can be linked via the performance rating achieved by the employee. This is a brilliant way of ensuring that the workforce is fairly rewarded in exchange for performance. How do the HCM modules link in with Compensation Workbench?

In the diagram below, you can follow the inputs into the compensation workbench (HCM and performance management), the two way interaction between Finance and Compensation Workbench, and the full cashable benefits as output from compensation workbench to payroll.


HCM
1. Employee Personal Information
2. Employee Assignment Information
3. Appraisal Types
4. Performance Ratings
Payroll
1. The collection and processing of approved compensations per employee
Finance
1. Processing of Compensation Budgets by cost centre
2. Management of Payroll Output for the employee individual compensation
Performance Management
1. Appraisal Recording
2. Competencies
3. Objective Setting
COMPENSATION WORKBENCH
1. Allocation of Compensation Budgets
2. Allocation of Performance Related Compensation
3. Allocation of Employee Benefits and Bonuses
iTrain
enabling change

Personal Details and Competencies are stored in HR. This information is shared with Performance Management where succession planning and objectives are outlined. The employee's appraisals are recorded against those objectives.

When the manager has completed the appraisal, the allocated budget and the compensation spine can be used to allocate the appropriate compensation (financial award) for the rating achieved during the appraisal.

At this stage, the relevant pay information is processed via Payroll so that the employee payslip reflects the new salary award/compensation.

Finally, the new pay or compensation information is seen by finance.

> "Compensation Workbench ... provides complete transparency to the organisation of how benefits and rewards are being managed ... ensuring that the organisation is investing wisely."

Can you restrict the eligibility of certain compensation schemes?

Yes, all compensation schemes can be restricted to particular staffing groups, such as bonuses only being available to management or a certain value belonging to a certain grade or employment group.

Do you have to be restricted to only one plan per employee?

No, Compensation Workbench allows you to set up a combination plan that will allow an employee to be a member of more than one plan.

A standalone plan restricts an employee to only having access to the benefits within that one plan.

What kind of things do I need to consider before implementing a plan?

The important tasks that need to be considered when implementing Compensation Plans include the following:

1. Budgets – How will the money be distributed?
2. Performance Rates – What performance Rates will be associated?
3. What kind of approvals and who will be the receivers of the approvals? Will the approvals be driven by a supervisor hierarchy or a position hierarchy?
4. Eligibility – Who will be a receiver of the Compensation and Benefits?
5. Cut off Dates – For example, with performance reviews, you will need to consider submission deadlines, review timetables, levelling timetables, and of course payroll timetables.

Summary of Key Benefits

Manage compensation across different areas of the business both locally and nationally

Allocate budgets to the varying areas of the business from local variations of terms and conditions or to different global entities.

Transparency and Fairness

Compensation Workbench ensures that each member of staff is benefitted using the same scoring structure, ensuring a fair reward system. It also provides complete transparency to the organisation of how benefits and rewards are being managed and which benefits are most commonly adopted, ensuring that the organisation is investing wisely.

Manage the various types of compensation as when required

CompensationWorkbench will manage a broad range of compensation types from pay awards based on merit or promotion, bonuses and commissions, one off rewards and employment benefit packages.

Meet the needs of the business

All businesses require a system that will do more than simply manage the established compensation plans. They require a system that is easy to manipulate and allows them to create new compensation plans in line with the changes and demands that a business will face on an annual basis. The Oracle Compensation Workbench is flexible enough to be configurable to cope with frequent movements and changes of a multi national or global business.

Control Parameters

Compensation Workbench will allow an organisation to apply parameters to a specific allocation of compensation ensuring that the correct group of people or individual receive a benefit or compensation for the right reasons, whether that is due to Performance or Grading. All relevant and pertinent Legal or Corporate Policies can be implemented within the plans as a means of control. Equally, the allocation and application of approved budgets using Compensation Workbench control where expenditure is being applied and ensure the process is transparent and traceable, both for the benefit of the employees and the organisation.

Summary

Irrespective of whether you are a global company that has to manage a number of varying compensation plans for each business unit or a mid-range company that is looking to apply some structure and reporting functionality to your company benefit plans, Compensation Workbench will support your needs. It will give staff visibility of what benefits they are entitled to and what they can aspire to achieve, it supports fairness in the distribution of compensation and benefits and finally allows the organisation to control its budget.

About the Author

Nina Cannon is an iTrain Practice Manager and Consultant with skills across both Oracle and SAP ERP applications. Nina's Oracle expertise covers Recruitment, HR, Payroll, Oracle Learning Management and Discoverer. Nina's knowledge of Payroll and HR systems, processes, rules and regulations was gained while working on major national projects for both the public and private sector.

w.itrainconsulting.co.uk

uksales@itrainconsulting.co.uk

To gather statistics or not to gather statistics?

By Tony Hasler, Anvil Computer Services Ltd

That is the question. Whether it be better to suffer the slings and arrows of outrageous execution plans or to take up arms against a sea of performance problems and by so doing end them.

How and when to gather optimizer statistics is a difficult issue, and there is no single answer that is suitable for all databases. This article explores the matters that need to be considered, so that you can decide what is best for you and your database.

I have to take a deep breath at this point. Not only because I feel the wrath of Hamlet lovers everywhere, but also because this is going to be a contentious article. Many of you will disagree with my analysis and with my conclusions, but then again there is no great consensus amongst the experts.

If, however, the article raises both awareness and debate I will consider it a success.

What are we trying to optimise?

The first thing to understand is that the performance of our databases needs to be *consistent* and *predictable*. This is *not* a matter of maximising averages.

This is a general point about performance management that is true not just for databases, and not just for IT. A few years ago The Royal Mail conducted a survey of its customers to see how important to them a morning delivery was. They wanted to know whether a lunchtime delivery would be acceptable if the price was lower. What they discovered was that their customers were concerned that the service was *guaranteed* far more than with the specifics of the service itself. For example, imagine that you had a car that did 200 miles to the gallon most of the time, but that unpredictably once in a while, for short periods, it started consuming one gallon for every two feet. The car would be useless to you, wouldn't it? The same is true for our customers. Sometimes, there is a formal Service Level Agreement (SLA) in place. If so, then it matters not if you exceed the service level by a long way ten months of the year, if you fail to meet it by an equally long way the other two. Even if a formal arrangement is not in place, database users have an expectation of "normal" performance and will make plans based on that performance level being met.

> "Oracle has given us at least six different technologies to help us with this problem that I will mention in this article but none are without problems."

In case you are wondering, this is not the contentious part of the article. You will find very few, if any, experts who will disagree with this point.

Why the Cost Based Optimizer is difficult to manage

The good thing about the old Rule Based Optimizer (RBO) was that it was *consistent* and *predictable*. Give it the same SQL statement and it would produce the same execution plan every time (unless you'd rebuilt some indexes at some point, and inadvertently changed their priority by creating them in a different order). Of course the plan may very well not be a good one, but if it was truly awful then you would likely discover it during your testing.

The Cost Based Optimizer (CBO) was introduced in Oracle 7 and operates in a radically different way. It identifies an execution plan not on a fixed set of rules, but on statistics about both the system where the SQL statement is running and the objects being accessed. This *generally* results in a better execution plan, but not always. In other words, even though its performance is usually better it is neither consistent nor predictable. This is one of the main reasons why many DBAs did not implement the CBO in Oracle versions 7, 8i and 9i.

Unfortunately this comparison between the two optimisers is over simplistic. For one thing, there are several reasons why it may not be possible to identify and manually correct the poorly performing SQL statements in your system. There may be too many, the source may not be available for a variety of reasons, or the users may have the ability to create their own SQL statements either directly with SQL or indirectly via SQL statement-generating tools of one form or another.

However, the biggest problem with the RBO is that the same execution plan may not give you the same performance. For example, when a system is newly deployed its database may grow dramatically in the first few weeks and months in ways that may not be easy to predict in advance. When this happens, it may be necessary to change the execution plan in order to *prevent* degradation in performance and achieve the consistency we are striving for.

For this reason the CBO is a much better optimiser than the RBO, and as of 10g you must use it to obtain support except in certain specific circumstances[1].

And so we are left with the unavoidable problem of having to manage the inherent unpredictability of the CBO's behaviour. Oracle has given us at least six different technologies to help us with this problem that I will mention in this article, but none are without problems.

Again, all of this is fairly uncontentious. It's when we start to discuss what to do about it that the fun starts.

It is a question of "how and when" rather than "if"

Let us make one thing clear right from the start: the CBO needs statistics to operate reasonably. You must at some point obtain object statistics for most if not all of your database objects one way or another.

In Oracle 9i, system statistics (i.e. statistics about the hardware response time) were introduced. You really should obtain these, but the CBO will operate with defaults if you do not. For more information on the importance of system statistics see the following articles from Jonathan Lewis:

[1] The CBO examines statistics stored in the data dictionary and so will not operate unless the database is fully open. If you query the controlfile when the database is only mounted you have no choice but to use the RBO.

http://www.oracle.com/technology/pub/articles/lewis_cbo.html

http://jonathanlewis.wordpress.com/2007/05/20/system-stats-strategy/

The normal way to obtain object statistics is to gather them. However, in certain circumstances you can copy them from another object, set them to some known values, import previously exported statistics, or some combination of the above. For example, you could gather statistics for a schema and then manually adjust the clustering factor for one of the indexes in the schema.

As of Oracle 10g the CBO will, by default, use *dynamic sampling* if statistics for an object are unavailable. You might, for example, use this with temporary tables as it is difficult to gather statistics for them, but relying on dynamic sampling all the time may be both inefficient and unpredictable.

So to be clear: this article is about how and when to obtain statistics, not whether you need them at all.

Why does the CBO sometimes get it wrong?

These types of article have a serious danger of misleading naïve people. I know you are not one of them, but in case one of your junior colleagues (or your manager) should read this article:

- The CBO will almost always generate a plan that is as good as or better than that created by any human, even an expert

- When the CBO does generate a bad execution plan there is a good chance that it is because the statistics are stale

- Another likely reason the CBO might seem to do a bad job is because of poorly written SQL. In theory it doesn't matter how bad the SQL is as the optimizer should do the best that is possible for that SQL. However, if your database has a column that has legal values of either "1" or "2", it will be better to use the predicate "c1 = 2" rather than "c1 <> 1" since the CBO will look in the latter case for rows that have "illegal" values. This is just one example of how two queries that appear identical might perform quite differently

Now we have that sorted out, there are a small number of cases where the CBO will get it horribly wrong even with well written SQL and the most accurate and up-to-date statistics. Unfortunately this small number of cases tends to be highly visible. Here are a couple of examples.

"The only time it is safe to gather statistics is when to do so would make no difference."

The CBO currently has no idea about buffer cache hits. If you perform 10,000,000 logical I/Os on 10,000,000 blocks you will probably at some point need to read most if not all 10,000,000 blocks from disk into memory. If, on the other hand, those 10,000,000 logical I/Os hit only 100,000 blocks that happen to be already cached, then you may be able to avoid physical I/O entirely. I have seen this happen when table data is sorted by the leading column of an index, but not by the other columns in the index. When this happened to me, I was able to improve the performance of an entire suite of queries by a factor of over one hundred by "misleading" the CBO![2]

Another problem with the CBO relates to correlated predicates. Suppose you have this query: (see Figure 1):

Figure 1

```
select * from star_signs s join birth_dates b
using (employee_name) where s.sign_name='Leo'
and b.birth_month='August'
```

What is the selectivity of the query? Assuming that the CBO is able to correctly determine that one twelfth of the rows in star_signs have a value of 'Leo' for sign_name, and one twelfth of the rows in birth_dates have a value of 'August' for birth_month, then the CBO will calculate the selectivity of the above query as 1/144, the product of the two contributing selectivities. Of course, the selectivity is actually about one in ten because most people born in August have a star sign of Leo and vice versa. When several tables are joined together even small errors in selectivity calculations like this one can be magnified into huge miscalculations. Incidentally, this is the sort of thing that is constantly improving in the CBO. In Oracle 11g *extended statistics* can be used to help identify correlations between columns, but as of yet only between columns in the same table, and only when the number of combinations is relatively small (254).

Even when the CBO makes an educated guess that turns out to be hopelessly wrong, it may stumble upon a decent execution plan by chance. However, if the statistics change even by a small amount then the CBO may select an execution plan that it now believes has just become slightly better than the one it used before. It is at that point the on-call DBA has his or her pleasant dreams disturbed.

The Dave Ensor Paradox

> *"The only time it is safe to gather statistics is when to do so would make no difference."*

This quote from Dave Ensor sums up nicely the views of those that feel that gathering statistics on a production system is inappropriate. The idea is not that you don't gather statistics at all but rather that you treat statistics like code; you subject all changes to an appropriate level of testing before deploying into production.

Suppose you were mucking about with "explain plan" on your test system, and thought that by adding a hint to a SQL statement you could reduce the run-time to below that of the un-hinted code. In most environments you would be in serious trouble were you to add the hint to the production code without telling anybody about it either beforehand or afterwards and without even actually *running* the SQL statement anywhere. You might even lose your job. Gathering object statistics on systems might well change execution plans to new and untested ones, so why do we let Oracle do something that we might get fired for?

This is a powerful argument, and in my opinion one that should be carefully considered when planning your statistics gathering strategy. In many ways, by testing statistic changes you get the best of both worlds. On the one hand you get

[2] There is an initialisation parameter *optimizer_index_cost_adj* that can be adjusted to help the CBO but I don't like using it as its effects are system wide and not object or query specific.

> "...if your users create their own "ad hoc" SQL statements then you are pretty much sunk..."

a set of object statistics that helps the CBO to come up with an execution plan that will probably be the same or better than that of the RBO, and on the other you minimise the risk of a problem by leaving the statistics alone until the next release of software. If you do not have a release of software planned, you may need a special release just to update statistics. Either way the statistics changes are tested before being deployed.

But be careful: this logic relies on at least three dubious assumptions. The first is that the same execution plan means the same performance. We have already discussed this and if the size of your database *and its tables* are not all fairly static this policy will be high-risk.

The second assumption is that the database is accessed by a fixed set of SQL statements, specifically the ones that you have tested. This is usually untrue. Consider this statement (see Figure 2):

Figure 2

```
select * from t1 where date_column = sysdate
```

This SQL statement changes every day, because the date changes. If you have column statistics on `date_column` then the CBO may have a histogram, and may create a different execution plan depending on whether it believes there are any values in the table that match the current date, and if so how many. In some circumstances, however, you may be able to identify and manage these situations.

Of course if your users create their own "ad hoc" SQL statements then you are pretty much sunk, and you must have up-to-date statistics to maximise your chances of getting a reasonable execution plan for these un-testable SQL statements.

There is a third false assumption, but it does not significantly affect the matter of how and when to obtain statistics. This assumption is that the same SQL statement with the same set of statistics will result in the same execution plan. Almost incredibly this assumption is also false. In Oracle 9i a feature called *bind variable peeking* was introduced. Without going into too much detail about this feature, it means that a SQL statement that uses bind variables can get one execution plan on one day and another on another – even with the same values for the bind variables! This is a classic example of maximising the averages. The resultant lack of predictability has earned this feature a well deserved bad reputation[3]. However, bind variable peeking is not a major factor in determining a statistics gathering strategy since it is equally unpredictable however you gather statistics.

What has Oracle done to help?

Oracle is not unaware of this issue and provides a number of features to try and improve the level of stability with the CBO.

In Oracle 8i, pretty much the only option you had available was the use of hints. This Oracle feature is still a crucial tool today but is inappropriately named. These so called hints are actually directives; assuming the CBO recognises the "hint" as legal it is obliged to act on it unless overridden by a Stored Outline, SQL Profile or SQL Baseline – we will come onto these features shortly. Hints allow you to control some or all aspects of the execution plan. The general advice here is that if you do decide to use hints then you should hint heavily to fix the execution plan completely.

The "Stored Outline" became available in Oracle 9i, and allows you to fix the execution plan for a specific set of SQL statements without adding hints to your code. In fact, as just mentioned if a Stored Outline is used then all hints placed in the code will be ignored.

One way to use this feature is to save all of the execution plans for all SQL statements run on your system. Any SQL statement that is run more than once will always use the same execution plan. The CBO should only devise a new execution plan when it sees an SQL statement that has never before been run on the system. This means that when you update statistics it will not affect the plans of any previously executed SQL statements. However, if a small change is made to an SQL statement then a brand new execution plan will be created.

In practice I have never heard of Stored Outlines used in this way so cannot comment on how effectively it works in practice. There is, however, an Oracle Whitepaper that describes (in Appendix A) how you might use this technique as part of an upgrade plan. It is entitled "Upgrading from Oracle Database 9i to 10g: What to expect from the Optimizer" and can be found here:

http://www.oracle.com/technology/products/bi/db/10g/pdf/twp_bidw_optimizer_10gr2_0208.pdf

The other way to use Stored Outlines is for a limited set of specific SQL statements. This is usually done when no access to source code is available, and hinting is therefore not an option.

By the time 10g came along, Oracle had obviously given this issue a great deal of thought. First of all, if you have the SQL Tuning Pack then you have access to the SQL Tuning Advisor and SQL Profiles. If the term "hint" had not already been used then SQL Profiles might well have been named SQL hints, since these are actually hints rather than directives. They do not fix a specific execution plan but reduce the risk that any new execution plan will be outrageous.

In addition Oracle made an implicit recognition of the dangers of changing statistics for the first time. Two of the new procedures added to the DBMS_STATS package were LOCK_SCHEMA_STATS and LOCK_TABLE_STATS. This allowed you to prevent automatic changes to statistics for specific objects.

Oracle 10g also provided an automatic statistics collection job for the first time. That job only gathered statistics on objects where the statistics were considered "stale". The ability to gather statistics only on stale objects has been around since Oracle 8i, but has never been the default option. The theory is that if there has been little change to the data, then updated statistics are not likely to cause the CBO to generate a much better plan. Statistics are considered stale if 10% of the rows have been inserted, updated, or deleted since the last time statistics were gathered. This figure of 10% seems ludicrously low and is customisable as of 11g.

In addition to the customisation of the definition of stale statistics, Oracle 11g introduced SQL Baselines. Without the SQL Tuning Pack the feature provides a direct replacement for Stored Outlines, which are now deprecated. With the SQL Tuning Pack you have the option to allow it to test new execution plans during a

[3] Oracle 11g introduces a feature called "Adaptive Cursor Sharing" which should reduce the risk of using an outrageous execution plan. Time will tell!

maintenance window, and then to evolve the baseline for an SQL statement if the cost is substantially lower. The definition of "substantial" is a factor of three. This un-customisable figure seems as ludicrously high as the 10% change figure mentioned above seemed ludicrously low. It does, however, suggest a growing understanding of the importance of stable execution performance. Perhaps this figure will be customisable in a future release?

Some advanced topics

The most common use of the **Partitioning Option** is range partitioning by date. Typically, new partitions for upcoming periods are created as older partitions are dropped, which can create some issues for statistics collection. If you were to collect statistics for the newly created partition just after it was created, then the statistics would accurately reflect the fact that the partition was empty. There may not be a sensible time to collect statistics on the partition as it will be constantly growing. Frequently the maximum size of a partition for one period will be about the same as that for another.

There is a procedure in the DBMS_STATS package called COPY_TABLE_STATS, that can be used to copy statistics from one partition to another. The procedure is documented for the first time in 11g but is available in earlier releases. By copying statistics from a full partition to an empty one, you can ask the optimiser to base its calculations on the worst case scenario. But once again we have to be careful; column statistics, including histograms, may exist for the partitioning column or columns and these may need to be manually adjusted. Similarly global column statistics may exist that indicate maximum and minimum values. You may wish to adjust or delete these as well.

Some tables, such as staging tables used as part of a bulk load operation, may have highly volatile statistics. In these cases it may be worthwhile to gather statistics each time the tables are loaded to make sure that the queries that follow behave optimally. This is an alternative to using a locked set of representative statistics. Remember, however, that you cannot gather statistics for temporary tables; you must either set them manually to representative values or rely on dynamic sampling.

Sometimes accurate statistics can be misleading and less accurate statistics can be more meaningful. In these cases you can adjust the automatically gathered statistics to less accurate values so that the optimizer is more likely to pick a better execution plan. For example, if you lower the clustering factor of an index you will increase the likelihood that it will pick that index over others or a full table scan. This was the technique I used to deal with the weird case I mentioned above where I knew that the Buffer Cache Hit Ratio would be very high with the index and very poor with the full table scan. Don't get carried away though; I have used this technique only once in my career. It should not be used because you know that "indexes are good and full table scans are bad". Only do this if you are sure both that the optimiser is getting it wrong and you know why it is getting it wrong.

> "... the risk of a performance related problem can never be entirely eliminated."

How can you devise a statistics gathering policy?

First of all you need to recognise that this is one of those cases where one size does not fit all. If you manage a large number of databases, you should probably not enforce the same policy on all of them. Most of the time regularly updating object statistics using the "GATHER STALE" option will be the best option, particularly if users generate "ad hoc" queries that cannot be tested in advance. If you have a limited number of critical SQL statements then my advice would be to fix their execution plans with hints, Stored Outlines, or SQL Baselines. If your database uses only a fixed set of statements (as is the case on many OLTP systems, for example) and the size of all of the tables remains fixed you may decide to gather statistics only as part of a tested release. In this case you will have to carefully examine your column statistics, make provision for any partition maintenance operations and generally expect to spend a lot of time getting it right.

A final note of caution

Bear in mind that the CBO creates an execution plan for DML and DDL statements, not just queries. This anecdote comes from an experience unrelated to Oracle, or to any database for that matter, but which is quite apt here. One day our system started to go desperately slow. The problem was tracked to a piece of code akin to the following, (see Figure 3):

Figure 3

```
delete from some_table where the_date_column < sysdate – 400;
```

The idea was to retain one financial years worth of data on the system (with about a month extra for safety). The data didn't seem to have changed that much. The execution plan for the statement hadn't changed at all from the previous day. But the previous day the job had run in under one second and now it was taking hours. Eventually somebody asked the question "When did we go live with this system?" You guessed it: exactly 400 days earlier!!

The motto of this story is simple. It doesn't matter how much time you spend designing and planning your system the risk of a performance related problem can never be entirely eliminated. So spend your time wisely and make sure your expectations are realistic.

About the Author

Tony Hasler is an independent software consultant specialising in helping companies improve Oracle related services. During his thirty years of experience Tony has led operating systems development teams, represented the British Standards Institute internationally, and filed a patent relating to optimisations of distributed transactions.

He can be reached at tony@aberdour.demon.co.uk

An Introduction to Services in RAC
Part 1

by Bob Mycroft, Fortissimo Solutions

This article should be of interest to the DBA and system architect who is interested in getting the best use out of their RAC databases. This article effectively forms the first part of a two part-er where I want to talk about the use of Services in RAC for workload distribution. In the second article I will be looking at server-side load balancing and client connection load balancing along with Transparent Application Failover.

What is an Oracle Service and why use them?

A Service is really just a way to take similar connections or workloads and put them together into a logical grouping so that they can be managed together as a unit. Services can be used to restrict the resources a session may use, to transparently enforce to which instances a connection is made, to control session failover (TAF) and to control which instances a parallel query can run on.

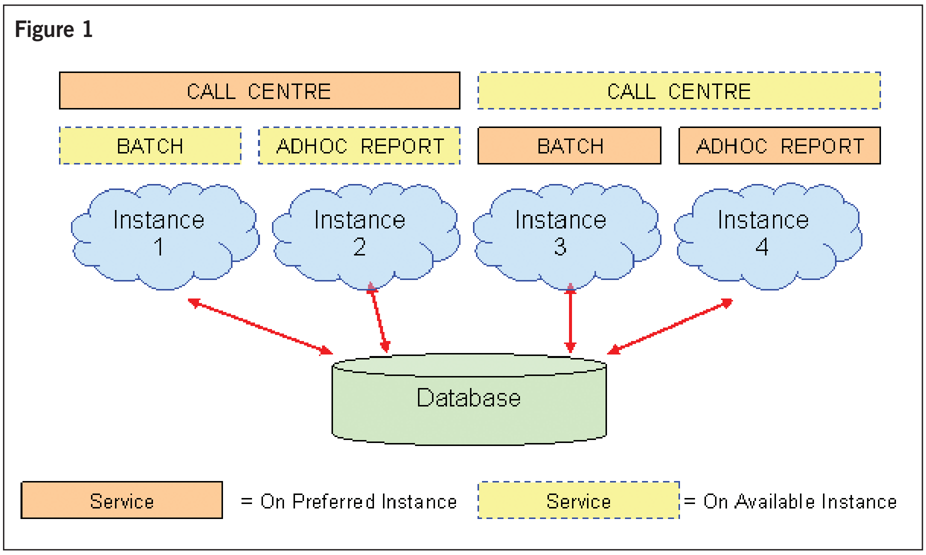
One of the real benefits of RAC is in its workload management – with a RAC database it is possible to partition the workload so that certain parts of the application are serviced only by certain instances, and to balance the incoming workload across those nodes. This can be done through the application of Services. Services can be associated with a single or multiple instances as seen in the following diagram, with a main “Preferred instance” and a failover “Available instance”. (See Figure 1.)

By default every 10g Oracle database and later has three database services; two are always called SYS$USERS and SYS$BACKGROUND and are created as part of the database creation (sql.bsq). You cannot manage these services directly as they are used internally by Oracle. Background processes connect to the SYS$BACKGROUND service and all other connections are registered against the SYS$USERS service unless explicitly connected using a user-defined service. You will also always get a third service that has a name of the format <DB_UNIQUE_NAME>.<DOMAIN_NAME>.

> “...you can partition the workload to reduce the impact between parts of the application.”

You can see which sessions are running in which service by querying the SERVICE_NAME column of V$SESSION. To use services in your system it is necessary to create your own services and decide what will connect to each service and which instances will run which services.

Figure 1



Using Services to partition the workload by business function and application

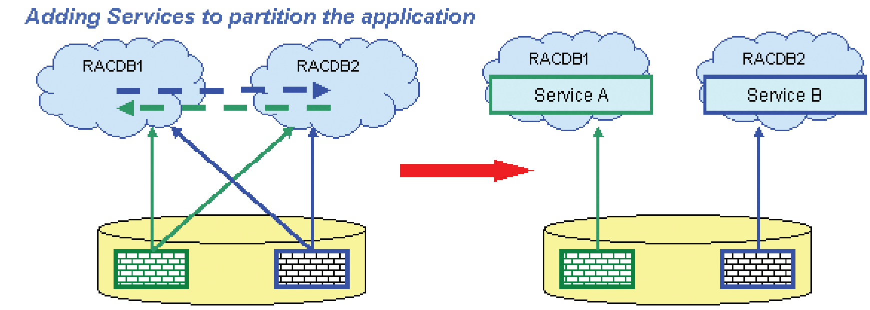
When a connection is made to the database, the service name is specified in the connect details and hence controls which instances of the cluster the session may connect to. In this way you can partition the workload to reduce the impact between parts of the application.

One of the nice things about services is that they can be dynamically altered to include further instances of the RAC cluster. Imagine that it’s Easter and in our company the customer care team get a lot more calls at this time of year – perhaps even recruit more staff to man the phones. At the IT level we can also extend the list of instances that the CALL_CENTRE service runs upon to include less heavily used instances, and so give the service more resources to run the additional workload. This can be done seamlessly and the people in the business shouldn’t even know that any changes have been made.

We may also want to segregate the application so that the ADHOC_REPORT and BATCH services run on different nodes to our CALL_CENTRE service. The aim is that the other workload then isn’t taking up resources that might be used by the CALL_CENTRE service workload.

Using services in this way is all about hiding the source of the power – i.e. the end-user or application doesn’t care or need to know what instance they are connected to. If the application needs more power then you can add in another instance and node to the cluster and extend the list of instances servicing the relevant service – the analogy often seen is that of a power station; You plug your kettle in during the ad-break and don’t care where the power comes from, or that the power station had to bring another generator online.

Figure 2



When tuning a RAC database you should approach it in the same way that you would a single instance. The only major difference is related to the way in which Oracle implements its Cache Fusion – managing the virtual cache that crosses all instances. This is implemented by using a high-speed low-latency private interconnect between the nodes in the cluster. If instance A has a copy of the block that instance B needs, it can be sent across the interconnect rather than having instance B read it from disk (there is much more to it that this but in essence Cache Fusion is all about memory reads and avoiding disk reads). Under some circumstances it could be the case that the network interconnect becomes overloaded and begins to affect performance – imagine if you have two really poor piece of sql that each perform full table scans, and that each of the SQL statements are running on both nodes; both nodes will be contending for the same blocks and the blocks will be getting passed back and forth across the interconnect as each instance comes to request the same block.

The first thing to do is to tune the SQL if possible (as you would in a single instance database). Next is where Services comes in to help things; you can reduce the data volumes being passed across the interconnect by partitioning the application with services. If you can make it so that in the prior example one of the statements only runs on one node, while the other statement only runs on the other node then as long as the statements are referencing different database objects (or blocks), none of the blocks will be getting passed across the interconnect. (See Figure 2, above.)

In real life then it may be the case that many parts of the application do not share substantial parts of the data – our customer care application looks at information such as names, addresses, telephone numbers – while the batch programs connecting to the BATCH service are looking at financial and order information. In this scenario then partitioning the application via services to run on different nodes will cut down the number of blocks being sent across the interconnect, and hence reduce the network traffic and *potentially* help sustain performance (IF the interconnect was a bottleneck).

Partitioning of the application by node/instance in this manner is one of the advantages of RAC that allows us to put more applications into a single database rather than having multiple databases (one per Application). Reducing the number of databases helps reduce the overheads and allows you to make better use of the hardware available.

Available and Preferred nodes

A feature of services is that in the event of failure, they can fail-over to run on other instances of the RAC cluster. When you create a service you can define what instances of the cluster the service will run on by default – known as the *"Preferred"* instance, and what instances will take over in the event of the Preferred instance dying – known as the *"Available"* instances.

If the Preferred instance dies then the service automatically fails over to the Available instance without any manual input from the DBA. When the failed instance is brought back online again, the service does not automatically fail back to its original position – you have to manually relocate the service. This is done so that there is not a second period of downtime while sessions are forced to disconnect and reconnect again. When forcing a relocation of a service you can choose to kill the sessions or leave them on the instance where they are.

Services should start automatically on reboot of the node – or rather they will go to the same condition as they were when the node was rebooted – hence if you manually stopped a service to do maintenance then it will stay down after the reboot.

Parallel query in services

Parallel queries are also "Service-Aware", although the way that it works in RAC differs between versions. In 10g the parallel query coordinator process is limited to nodes that are running the relevant service but the parallel query slaves themselves will run on any instance in the cluster – not just those that are running the named service. You can force the issue in 10g using the PARALLEL_INSTANCE_GROUP parameter. At 11g things are a little cleaner – you don't need to set the parameter explicitly (It is in fact deprecated) and the parallel query slaves will be restricted as you would expect to just those running the service.

> "...you can reduce the data volumes being passed across the interconnect by partitioning the application with services."

Services and the Oracle Scheduler

Services can also be tied into job classes with the Scheduler in 10g. In the older DBMS_JOB package you could specify an instance for the job, whereas at 10g you can specify a service and have the job run on any of the instances supporting the service.

To create a job class tied to a service, (see Figure 3, right).

To specify the job class when creating a job, (see Figure 4 right).

To view the results, query the INSTANCE_ID column of DBA_SCHEDULER_JOB_RUN_DETAILS as above, (See Figure 5, below).

Services and the Resource Manager

Services also fit nicely into Resource Manager; you can explicitly associate a service with a consumer group using the in-built plsql package called DBMS_RESOURCE_MANAGER. Use the SET_CONSUMER_GROUP_MAPPING procedure to map a service to a consumer group. This will define what resource group a new session is assigned to, based upon the service name that it connected with. Consumer groups use a number of attributes to decide the mapping (service name being only one of 10 attributes) and you need to decide the importance/precedence of the 10 attributes using the SET_CONSUMER_GROUP_MAPPING_PRI procedure. Check out "Specifying Session-to-Consumer Group Mapping Rules" in chapter 25 of the Database Administrators Guide for more info on this. The bottom line is that you can combine services and consumer groups to limit the resources that a group of users may consume at run time.

Figure 3

```
exec DBMS_SCHEDULER.CREATE_JOB_CLASS ( -
  JOB_CLASS_NAME => 'DEMO_CLASS', -
  RESOURCE_CONSUMER_GROUP => null, -
  SERVICE => 'call_centre');
```

Figure 4

```
exec DBMS_SCHEDULER.CREATE_JOB( -
 JOB_NAME => 'call_centre_job', -
 JOB_TYPE => 'stored_procedure',-
 JOB_ACTION => 'system.run_in_call_centre_service', -
 NUMBER_OF_ARGUMENTS => 0,-
 START_DATE => SYSDATE,-
 REPEAT_INTERVAL => 'freq=secondly; bysecond=0',  -
 END_DATE => null, -
 JOB_CLASS => 'DEMO_CLASS',-
 ENABLED => TRUE,-
 AUTO_DROP => false, -
 COMMENTS => 'my services job');
```

Creating, Managing & Monitoring Services

There are a number of ways to create and manage services but it is slightly version dependent. Methods for creating new services include using the DBCA, srvctl in a RAC environment, Enterprise Manager Grid Control or programmatically through PLQSL using the DBMS_SERVICE built-in package. You should **not** manually edit the SERVICE_NAMES database parameter in a RAC database. Although this is valid in single-instance Oracle, in RAC the parameter is set automatically and is managed by Clusterware. If you set it manually it will get overwritten. In fact at the point that you start a service using srvctl you can see this occurring as you will see a statement like the following appear in the alert-log! (See Figure 6, below.)

There are some in-built limits to the number of services that you can have; at 10.1 you can have a maximum of 64 services registered (Note:363777.1) and at 10.2 the limit is raised to 115. The total string length of all the services must not exceed 4K (i.e. maximum length of the SERVICE_NAMES parameter). At 10g you can use dbca for creating and managing services but at 11g the option to do this has been removed.

> "...you can combine services and consumer groups to limit the resources that a group of users may consume at run time."

Figure 5

```
column log_date format a40

select LOG_DATE, INSTANCE_ID
  from DBA_SCHEDULER_JOB_RUN_DETAILS
 where JOB_NAME = 'CALL_CENTRE_JOB'
 order by 1
/

LOG_DATE                                  INSTANCE_ID
----------------------------------------  ----------------
24-OCT-08 21.28.00.106110 +01:00                        1
24-OCT-08 21.29.12.511596 +01:00                        2
24-OCT-08 21.30.00.112168 +01:00                        2
24-OCT-08 21.31.00.105626 +01:00                        1
```

Figure 6

```
Thu Nov 13 21:55:24 2008
ALTER SYSTEM SET service_names='racdb1','call_centre' SCOPE=MEMORY
SID='racdb1';
```

As an example of creating and managing services using srvctl, let's walk through adding a service, starting and stopping it; In the following example I am using a 2 node 10g RAC cluster running on SLES10.

We will start off by adding a new service. To do this we will use the srvctl command to add a service, check the configuration and get the status. In the following, the *-d* switch shows database, *-s* is service name *-r* is the Preferred instance name and *-a* defines the Available instance. (See Figure 7, below.)

Let's take a look at what this adds to the OCR (Oracle Cluster Registry), using the crs_stat command from the Clusterware Oracle Home. (See Figure 8, below.)

Note in the above that there are two resources added in to the OCR – a ".srv" and a ".cs" resource. Each service will have only a single ".cs" resource but will have a ".srv" resource for each of the Preferred instances in the service.

The full Clusterware resource names are of the format; ora.<database_name>.<service_name>.<instance_name>.srv and ora.<database_name>.<service_name>.cs and you will be able to see this by calling the crs_stat command with no arguments.

Let's look at the listeners; At the point that the service is started, it is registered with all the listeners managed by the cluster. (See Figure 9, below.)

Ok so we have a running service on racdb2. Let's say we want to manually move the service to another instance; use the relocate command as below; (see Figure 10, below).

Figure 7

```
./srvctl add service -d racdb -s call_centre -r racdb2 -a racdb1

./srvctl config service -d racdb -s call_centre
call_centre PREF: racdb2 AVAIL: racdb1

./srvctl start service -d racdb -s call_centre
./srvctl status service -d racdb -s call_centre
Service call_centre is running on instance(s) racdb2
```

Figure 8

```
/opt/crs/bin/crs_stat -t –v

Name          Type         R/RA   F/FT   Target    State     Host
------------------------------------------------------------------------------------------------------
...
ora....nter.cs  application  0/0    0/1    ONLINE    ONLINE    racnode_b
ora....db2.srv  application  0/0    0/0    ONLINE    ONLINE    racnode_b
```

Figure 9

```
/opt/crs/bin/crs_stat -t –v

Name          Type         R/RA   F/FT   Target    State     Host
---------------------------------------------------------------------------------------------------------
...
ora....nter.cs  application  0/0    0/1    ONLINE    ONLINE    racnode_b
ora....db2.srv application  0/0    0/0    ONLINE    ONLINE    racnode_b
```

Figure 10

Relocate the service

```
./srvctl relocate service -d racdb -s call_centre -i racdb2 -t racdb1

./srvctl status service -d racdb -s call_centre
```

Service call_centre is running on instance(s) racdb1*

*Note the Preferred and Available instances haven't changed but the service is now running on the Available instance

```
./srvctl stop service -d racdb -s call_centre
./srvctl start service -d racdb -s call_centre
./srvctl status service -d racdb -s call_centre
```

Service call_centre is running on instance(s) racdb2*

*Restart the service and it is running on the Preferred node again.

> "...in RAC the service information is stored both in the Oracle Cluster Registry (OCR) and in the database's data dictionary."

Ok so it's a busy weekend and we want to have two instances powering the call_centre service. Below are the commands to modify the service. Note the *-i* switch rather than *-r* to specify the Preferred nodes in this command; (see Figure 11, below).

Finally we want to remove the service so we stop it and then delete it; (see Figure 12, below).

> "There is no reason why you cannot use Services in single instance Oracle."

Using DBMS_SERVICE to manage Services

In a RAC database, be careful when using the DBMS_SERVICE supplied package – in RAC the service information is stored both in the Oracle Cluster Registry (OCR) and in the database's data dictionary. DBMS_SERVICE has procedures to add, start, stop, modify and delete services; The problem is that CREATE_SERVICE doesn't create the services in Clusterware, START_SERVICE doesn't start the RAC service or register the service with the listeners, and neither does STOP_SERVICE work as you would expect. DELETE_SERVICE doesn't delete the service from the OCR but will remove entries from the DBA_SERVICES view. In a single instance database the above isn't an issue but with RAC where the information is stored in more than one place I would advise sticking to just using srvctl to manage your services.

However, there are at least two occasions where using dbms_service is of use; using srvctl to remove a service doesn't always clean it out of the data dictionary. This can lead to potential issues if the maximum number of services is exceeded – in this case DBMS_SERVICE.DELETE_SERVICE can remove old entries from the data dictionary allowing new services to be created. The other thing that DBMS_SERVICES is good for is setting parameters for TAF (Transparent Application Failover) and for the LBA (Load Balancing Advisor) for a service as you cannot completely specify these using srvctl alone; I will cover this in the next article.

Views to look at Services

There are a number of views that you can use to look at information about services including DBA_SERVICES, GV$SERVICE, GV$ACTIVE_SERVICES, GV$SERVICEMETRIC, GV$SERVICE_STATS and GV$SERVICE_EVENT (plus the V$ equivalents). Note that these views will tell you almost everything you need to know except for related info stored only in the OCR such as which instance is Preferred and which is Available for a specific service. The GV$SESSION view can be used to tell you which service on which instance a specific session is connected to.

And Finally….

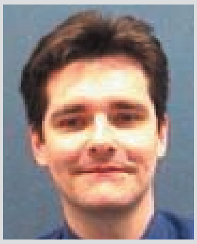
Hopefully this article will have given you an introduction to using Services in RAC. There is no reason why you cannot use Services in single instance Oracle either; it can be a useful way to instrument the system so you know that a certain percentage of your workload is coming in through say the "ORACLE_FORMS" service as opposed to the "WEB_SERVICE", and many of the tracing features in 10g onwards allow you to perform tracing against a named service (see the DBMS_MONITOR package for more info on that).

In the next article I want to talk about Connection and Server load balancing, and also about Transparent Application Failover (TAF) as services is tied in with all of these features.

References

- Note: 220970.1 Subject RAC FAQ
- Pro Oracle Database 10g RAC on Linux, Julian Dyke & Steve Shaw (very good book incidentally!)
- Note: 259301.1 CRS and 10g Real Application Clusters
- Bug No. 4622156 UNABLE TO MODIFY SERVICE VIA SRVCTL
- Oracle® Database Reference 10g Release 2 (10.2), Part Number B14237-03
- Oracle® Database Administrator's Guide, 11g Release 1 (11.1), Part Number B28310-03
- Note: 363777.1 Subject: How to Completely Remove a Service so that its Service_id Can Be Reused

About the Author



Bob Mycroft works as an independent Oracle Consultant for Fortissimo Solutions. He has over thirteen years experience with Oracle as both DBA and a Developer.

Bob can be contacted at bob_mycroft@yahoo.co.uk

Figure 11

```
./srvctl modify service -d racdb -s call_centre -i racdb1,racdb2 –n

./srvctl config service -d racdb -s call_centre
call_centre PREF: racdb1 racdb2 AVAIL:

./srvctl status service -d racdb -s call_centre
Service call_centre is running on instance(s) racdb2
```

You can see above that the configuration has changed but the service isn't running. To start the service on the relevant node do;

```
./srvctl start service -d racdb -s call_centre -i racdb1

./srvctl status service -d racdb -s call_centre
Service call_centre is running on instance(s) racdb1, racdb2
```

Figure 12

```
./srvctl stop service -d racdb -s call_centre

./srvctl remove service -d racdb -s call_centre
Remove service call_centre from the database racdb? (y/(n)) y
```

The Oracle Developers 12 Step Guide to Good Relations

By Rod West, Cabot Consulting

Bad habits are hard to change and bad habits in programming PL/SQL are no exception. After many years of debugging and maintaining PL/SQL I have stopped asking "Why o' why do people do this?" and have written a 12 step guide to better PL/SQL code.

PL/SQL is the language of choice for Oracle Database developers and is used extensively to customise the E-Business Suite and ApEx applications. Obviously there isn't space here for a definitive PL/SQL guide. With such a large and feature rich language one can only scratch the surface of all the topics covered. But it requires more than good programming skills to write good PL/SQL, as without a good understanding of the Oracle database the reasons for poor performance will remain elusive.

This is only a guide; it cannot give solutions for all situations. But hopefully by looking at just twelve areas where frequently I see mistakes, it will highlight the most common causes of poor performance, failures and confusion.

1. Watch out for incompatible types

What's the Problem?

Comparing text columns with number or date variables, or assigning text to a number or date variable.

Can You Explain?

You can find that some strange and unexpected things happen when variables of different types are compared. Take for example, the query below where empno is a VARCHAR2 column and v_empno is a NUMBER variable,

```
SELECT empname FROM EMP
WHERE dept = 'A'
AND EMPNO = v_empno
```

To be able to compare, both sides must be the same type. Therefore an implicit TO_NUMBER is introduced and the query rewritten.

```
SELECT empname FROM EMP
WHERE dept = 'A'
AND TO_NUMBER(empno) = v_empno
```

That extra TO_NUMBER will have some unwanted side effects. For a start, it will slow the query down because it prevents the database from using an index on the empno column. Worse still, the database can raise an *ORA-01722: invalid number* exception, sometimes even when all the empno are valid numbers for Dept 'A'. If the database full scans the EMP table then all the empno will be converted to numbers before there is any checking the value of the DEPT column.

Similarly, if text is assigned to a number an implicit TO_CHAR is used to convert the text resulting in unexpected *ORA-06502: PL/SQL: numeric or value error* exceptions. The results of the TO_CHAR are dependent on the database NLS parameters. These parameters can be changed or be different in other databases, introducing more unexpected behaviour. When the query is large and complex and there are many variables it is very hard to track down the implicit type conversion that is causing the problem.

What's the Solution?

- The easy answer is to use the correct type to hold all variables, so that no conversion is necessary
- If this is not possible, always use an explicit TO_NUMBER, TO_DATE or TO_CHAR to convert and where necessary handle the exception
- If your query has to convert a column and there might be a full scan on the table, then use a condition to ensure that only valid numbers are converted

```
CASE WHEN dept='A'
   THEN TO_NUMBER(empno)
 END = v_empno
```

2. Avoid explicit curses

What's the Problem?

Explicit cursors declared at the start of the programme.

Can You Explain?

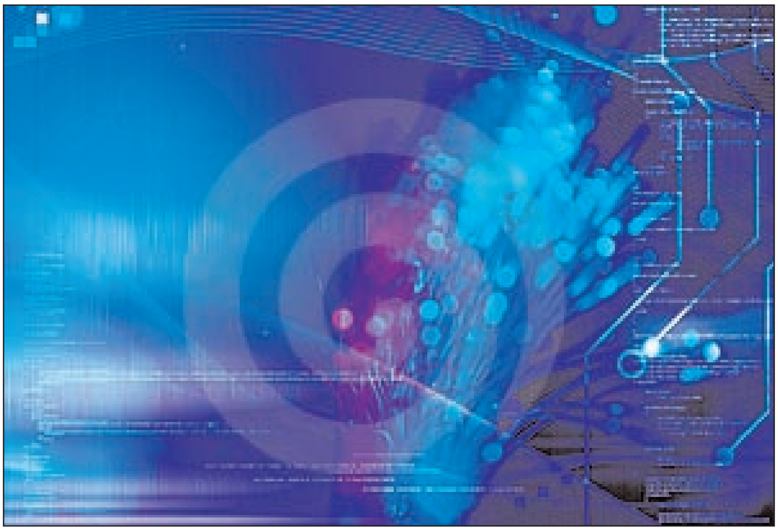
I don't like explicit cursors because I think they make the PL/SQL code hard to understand. You need to keep looking back at the cursor declaration to understand the programme. With an implicit cursor, where you just add the SQL statement into the code, you end up with less code. Less code means the PL/SQL compiler can achieve better optimisation which is why implicit cursors always perform better than explicit cursors.

Another undesirable feature of explicit cursors is that when a simple open and fetch is used to retrieve a row you will get an arbitrary result if more that one row is returned and the query is not sorted. As a consequence a programme can behave very differently when run twice with the same data; something that is very hard to troubleshoot. An implicit cursor will raise a *ORA-01422: exact fetch returns more than requested number of rows* exception in the same situation.

What's the Solution?

- Always use implicit cursors unless you need to do something clever, for example, use the REF CURSOR type
- If you do need to ensure that only one row is returned then use a group function or ROWNUM <= 1 in the query

> "When the query is large and complex and there are many variables it is very hard to track down the implicit type conversion that is causing the problem."



3. Make an exception even when others don't

What's the Problem?

Using WHEN OTHERS in the exception handling section of the programme,

```
EXCEPTION
WHEN OTHERS THEN
   DBMS_OUTPUT.put_line (SQLERRM);
END;
```

Can You Explain?

Remember that all exceptions are caught by WHEN OTHERS including exceptions that are completely unexpected; exceptions you would never have thought of. If these are not handled correctly then the programme will complete successfully but all the information about the error will be lost. Relying on the DBMS_OUTPUT package to report any errors is not sufficient because this package just stores the error in an internal array. The calling program, such as SQL*Plus, is expected to display the message and the calling program may not always be able to display the error in all circumstances.

What's the Solution?

- Handle all the expected errors explicitly and return to the main flow of the programme as soon as the error has been dealt with
- Use the exception handling built into the calling programme to handle all unexpected exceptions
- If you must use WHEN OTHERS then also include a RAISE to re-raise the exception

> "Ideally each function and procedure should do just one task; a package should then hold functions and procedures that share a common purpose."

4. Don't go off the record

What's the Problem?

Records are not used for storing variables.

Can You Explain?

A programme is much more difficult to understand and maintain if there are lots of global variables or there are lots of parameters in the procedures and functions. Global variables are in scope for the entire package and it is difficult to keep track of their use. Long parameter lists can easily be entered in the wrong order.

Records can help you by storing related variables together. A record containing a set of variables can then be passed into procedures and functions as a single parameter.

What's the Solution?

- Hold variables either in a record type defined in the package or use %ROWTYPE to define a record based on a table
- When passing a record into a procedure define the parameter as IN OUT NOCOPY

5. Understand their point of view

What's the Problem?

Views are not used to hold large or complex queries often found in PL/SQL.

Can You Explain?

Views are generally a "good thing"; they hide complexity and isolate programmes from changes to underlying data.

I find that implementing the queries used to extract data is one of the more difficult parts of implementing a PL/SQL package. Storing the query in a view means that the query can be tested, tuned and tweaked separately from the package. The package is then more compact and easier to maintain.

What's the Solution?

- Create a view to hold any large or complex queries
- When you cannot bind to a column in a view because for instance you need the condition WHERE empno = v_empno but empno is not a column in the view then create a database context for the package and bind to the view using contexts. For example, you can use the DBMS_SESSION package to set the context

```
dbms_session.set_context(
'MYNAMESPACE', 'EMPNO',
 v_empno);
```

Then bind the variable in the view using SYS_CONTEXT.

6. Don't let your body get too bulky

What's the Problem?

Packages are too big and have not been split into modular units.

Can You Explain?

I've spent a fair bit of time scanning through pages and pages of code in a package body looking for the relevant bit of code. If packages were well structured then this would be a whole lot easier. Ideally each function and procedure should do just one task; a package should then hold functions and procedures that share a common purpose.

Large packages are more difficult to maintain and can run slower because all the package must be loaded into memory before any procedure can be run.

What's the Solution?

- Always create procedures and functions within packages
- Use views and records to reduce the size of the package body
- Break packages into modular units if then get too big

7. Be a man, don't commit unless you have to

What's the Problem?

Committing too frequently, especially when the commit inside a loop changes rows used by the SELECT, *causing ORA-01555: snapshot too old* exceptions,

```
FOR c IN (SELECT ...)
LOOP
...
  COMMIT;
END LOOP;
```

Can You Explain?

I often see COMMIT statements scattered throughout PL/SQL code. This is almost always a bad idea. An Oracle COMMIT statement ends the transaction, makes all the changes permanent, frees locks and allows rollback space to be reused. However, any queries still in progress need to maintain a consistent view of the data and will need the undo information stored in those rollback segments to return data to a point in time prior to the commit. The *ORA-01555: snapshot too old* exception occurs when a new transaction reuses some undo space, freed up by a commit, but actually still needed by a query that is in progress.

Oracle transactions have a low overhead and so there is no reason to end the transaction unless you need to. Better to keep the transaction in progress until all the database changes are made. This will avoid the *ORA-01555: snapshot too old* exception but importantly it will also allow you to rollback all the changes if any error is encountered.

What's the Solution?

- In many situations, for example, in an Applications 11i concurrent request, no commits are necessary because the calling programme commits all the changes
- Always think in transactions and don't commit until the transaction is complete

> "...you don't know if the differences are due to the changes in the programme or because the database has returned rows in a different order."

8. Remember you have rights too

What's the Problem?

A package is created with definer's rights but references objects granted through roles.

Can You Explain?

Using definer's and invoker's rights, specified using the AUTHID clause in the CREATE PACKAGE command, can be a little confusing. By default a package is created with definer's rights meaning that when the package runs it uses the privileges granted to the owner of the package. But not if the owner has been granted any privileges through roles; these privileges are ignored. As a consequence, a user may have privileges to run an SQL statement in SQL*Plus but when they create a package containing this SQL the package will fail with *ORA-00942: table or view does not exist*.

On the other hand when a package is created with invoker's rights (specified using AUTHID CURRENT_USER) then the package runs using all the privileges of the current user. However, when a package is compiled (for both definer's and invoker's rights packages) the owner must have access to all the tables and views referenced.

Just to add a little to the confusion, definer's rights applies to views too, so if a view calls a function with invoker's rights then the function is run with the privileges of the owner of the view not the current user.

What's the Solution?

- Most situations require definer's rights packages and then always grant any privileges required directly to the package owner

9. Always get things sorted

What's the Problem?

Queries are not sorting data using an ORDER BY.

Can You Explain?

When you run the same programme against the same set of data one would hope that you will get the same results. But this won't be the case if data is not sorted, as then the database may return rows from a SELECT in any order. Working with a programme that behaves in an indeterminate way is always difficult and particularly so when testing changes. If you run a programme, change it and compare outputs you don't know if the differences are due to the changes in the programme or because the database has returned rows in a different order.

What's the Solution?

- Where a query returns more than one record always ensure the data extracted is sorted into a well defined order
- The only way to sort data in a query is to use an ORDER BY clause that contains sufficient columns to uniquely identify each record. For example, if you want employee records sorted by start date then ORDER BY start_date is not good enough, you need ORDER BY start_date, empno to guarantee that rows are always returned in the same order

10. Execute Immediate only as a last resort

What's the Problem?

Unnecessary use of dynamic SQL,

```
EXECUTE IMMEDIATE
'DELETE FROM EMP
WHERE EMPNO='||v_empno;
```

Can You Explain?

Native dynamic SQL is when SQL statements in PL/SQL are sent to the database with the EXECUTE IMMEDIATE command. This is a very useful technique that allows you to write generic programmes where the SQL is not known until run time. However, I am always reluctant to use dynamic SQL, using it only where there is no alternative.

I find using dynamic SQL is more complex and error prone. For a start, all the checking of the SQL, normally done when the package is compiled, must be carried out at run time. Therefore syntax errors and privilege errors won't get picked up until the SQL statement is executed. Also because the SQL is parsed each time it runs, dynamic SQL will run slower than static SQL. It will run a whole lot slower if the variables are included in the SQL text rather than using bind variables,

```
EXECUTE IMMEDIATE
'DELETE FROM EMP
WHERE EMPNO=:1'
USING v_empno;
```

What's the Solution?

- Always consider alternatives, such as multiple or more complex static SQL statements
- The DBMS_SQL package may be a better alternative if the SQL is executed many times
- Always bind the variables in dynamic SQL
- Dynamic SQL is fragile so code defensively with more exception handling

11. Never use a blunt instrument

What's the Problem?

Code is released to production without any proper instrumentation.

Can You Explain?

When code is running in production diagnostic tools such as tracing and profiling that were available in development usually cannot be used. Your production PL/SQL needs code built in to enable you to monitor performance and troubleshoot. Without it you will have no idea what your programme is doing, how well it is performing, or why it not doing what you expected.

What's the Solution?

- As a minimum use the DBMS_APPLICATION_INFO package to set the module and client_info for the session. This can be used to give an indication of what the programme is doing in the V$SESSION view
- For long running programme use the SET_SESSION_LONGOPS in the DBMS_APPLICATION_INFO package to show progress
- Include a trace mechanism where you can switch on writing trace messages into a diagnostic file

12. Enjoy the sequel

What's the Problem?

Code is written in PL/SQL when the same code could be implemented in SQL.

Can You Explain?

I think plain old SQL is often underrated and underused. There are many cool features in SQL; MERGE, analytic functions and subquery factoring to name a few. The versatility of SQL enables almost any data extraction and manipulation problem to be solved using a single SQL statement.

There is an overhead running an SQL statement from PL/SQL so you want to keep the number of SQL statements to a minimum. It is usually quicker to combine multiple SQL SELECT statements into a single SELECT, combine multiple INSERT statements into a single INSERT INTO SELECT FROM, and so on. A single SQL UPDATE statement will be much, much more efficient than using PL/SQL to loop through records in the table updating the records one at a time.

With SQL you always need to think in sets and identify the set of records that need to be extracted or modified. PL/SQL still has its uses; data formatting and complex procedural code is usually best carried out in the PL/SQL code. The aim must always be to minimise the number of SQL statements reducing the total amount of code, resulting in more efficient PL/SQL that is easier to test and maintain.

What's the Solution?

- If it can be done in SQL then do it in SQL. Sometimes your PL/SQL code is just not required at all

> "Your production PL/SQL needs code built in to enable you to monitor performance and troubleshoot."

About the Author

Rod West is a freelance contractor with over 25 years experience in the software industry who now specialises in Discoverer, Oracle Applications Express and the e-Business Suite. He first did his Oracle training on Version 3 way back in 1985 and since then has been continually extending his knowledge and learning about Oracle. He passes on his experience by contributing to OTN and the Learn Discoverer blog.

Rod can be contacted at rodwest@cabotconsulting.co.uk

The Tools of Fusion: Oracle JDeveloper and Oracle ADF

Building Business Services with JDeveloper and Oracle ADF

by Grant Ronald, Oracle

In the opening chapter of this column, I gave a broad-brush stroke overview of Oracle JDeveloper and Oracle ADF: the tools of Fusion. So now you know what they are, what can you do with them? Well, my guess is you're going to want to do exactly what Oracle is doing: build an application. And you know what? Most of these applications will have a similar goal: get data from the database onto a user's screen; let them do a bunch of processing on it and then save it back to the database. In this article I'm going to show you how Oracle's Fusion developers build the heart of these Fusion Applications: the business services.

At the heart of Fusion

Oracle ADF offers a number of different technology choices for building business services, for example, EJB, POJO (plain old Java objects) and ADF Business Components (ADF BC). Regardless of the technology many of the goals of a business service framework are the same. For example:

- O/R Mapping – just a fancy way of saying you need some way of referring to your relational tables from the object world of your Java application (think: getting a square "emp" peg in a round "emp" hole)

- Caching – when you pull data from the database, you need to hold onto it while you process it before committing it back

- Business logic – given that you are pulling this data in to do something with it, you'll probably want to add a bunch of validation and business rules

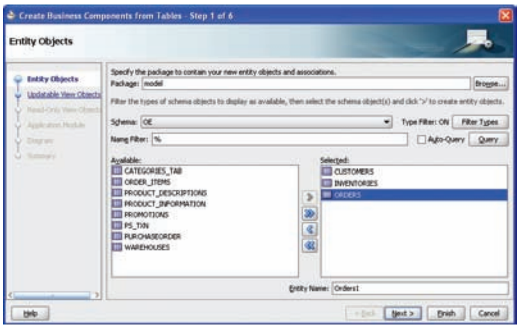
- Value add extras – let's face it, there are lots of "goodies" any developer would like to have in their applications like lists of values and calculated totals

For Oracle's Fusion Applications, the choice is ADF Business Components. There are many reasons why, one of the main ones being the productive and declarative nature of the development. Through Oracle JDeveloper, the full power of ADF Business Components is harnessed to allow complex business services to be built in only a few keystrokes.

So let's take a look at that development experience.

The ADF Business Components from Tables dialog (figure 1) allows you to choose which database tables you want to use to build your application. Choose your tables, and a couple of clicks later you have fully-functioning business service based on them. So, what makes this magic happen? Well, when you go through the dialog in Oracle JDeveloper, information about the chosen tables (the table name, the columns, primary keys, constraints etc.) is read and saved into metadata files with this metadata then being used to drive the framework. The framework already knows how to do stuff like locks, creation and deletion of records, but it's the metadata that lets it know which table, what attribute, what actions are allowed and whether a piece of data should be NOT NULL or is a primary key. So, driven by metadata, not code. Less code, less testing, easier to maintain and less things to go wrong!

Figure 1



Create Business Components from Tables - Step 1 of 6

Entity Objects

Specify the package to contain your new entity objects and associations.

Package: model

Filter the types of schema objects to display as available, then select the schema object(s) and click '>' to create entity objects.

Schema: OE

Type Filter: ON

Name Filter: %

□ Auto-Query

Available: CATEGORIES_TAB, ORDER_ITEMS, PRODUCT_DESCRIPTIONS, PRODUCT_INFORMATION, PROMOTIONS, PS_TXN, PURCHASEORDER, WAREHOUSES

Selected: CUSTOMERS, INVENTORIES, ORDERS

Entity Name: Orders1

Now, let's have a look at the various building blocks of ADF Business Components. There are essentially three main elements created when you go through the above dialog – all of which can also be created individually.

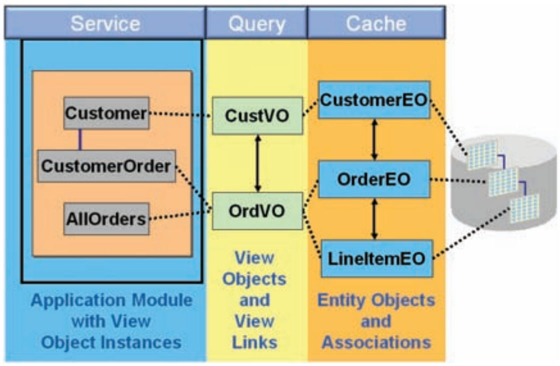
First of all there is the *entity object*. All this is doing is being a "bucket" for a row of data coming back from your database. Being the container in which the data in held and manipulated, it is also the place where you can add your data validation logic. So, as data goes in and out of the entity object, you enforce your data and business validation.

The second element is the *view object*. In the same way that a database view shapes underlying database tables, the view object shapes your view of the underlying entity objects. So, for example, your view object might include an order by clause, a where clause and pull attributes from different entity objects: allowing the application developer to create an application specific view of the data. You could have a view object of all UK employees reaching retirement age and also a view of all global new hires, both being based on the same employee entity object. This means that the same rules apply to both views (for example, determining that commission is only available to sales employees) and if you hire a 58-year-old employee based in London he naturally appears in both views because you are viewing the same source entity. So, application specific views of the same data source.

Building Business Services with JDeveloper and Oracle ADF

Finally, the *application module*: the container in which you organise your view objects to construct your business service. For example, you might have two view objects representing orders for a customer (master/detail) but may also have a view of all orders. So, a total of three instances of two individual view objects. (See Figure 2.)

Figure 2


Service
Query
Cache
Customer
CustomerOrder
AllOrders
CustVO
OrdVO
CustomerEO
OrderEO
LineItemEO
Application Module with View Object Instances
View Objects and View Links
Entity Objects and Associations

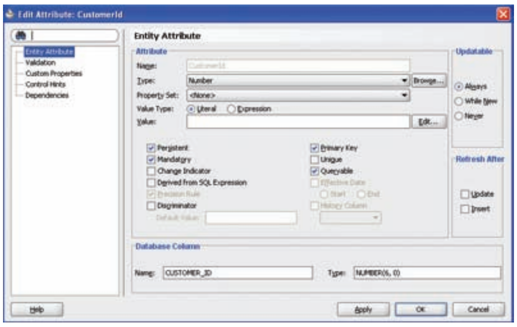
Ok, so that's the high level overview of the building blocks for these business services; the next step is to have a quick look at what this looks like to the Fusion developer.

Entity Objects

Once you have created your entity objects, there are essentially two main things you might do: fine-tune the attributes that map to the database columns, and add validation and business logic.

Figure 3 shows the Edit Attribute dialog. Most of the information on this dialog is introspected from the database table. For example, CustomerId is a number type, is mandatory and a primary key. You can of course change this as required, as well as fine-tune other aspects such as whether CustomerId is updatable. You can also start defining information as to how this attribute should be displayed when finally exposed through a UI. For example, you can specify the field label, help text associated with that field or whether a format mask should be applied. Again, all of this is codeless: it's simply you driving the framework.

Figure 3



Entity Attribute

Attribute

Name: CustomerId

Type: Number

Property Set: <None>

Value Type: ● Literal ○ Expression

Value:

☑ Persistent
☑ Mandatory
☐ Change Indicator
☐ Derived From SQL Expression
☑ Precision Rule
☐ Discriminator
Default Value:

☑ Primary Key
☐ Unique
☑ Queryable
☐ Effective Date
○ Start ○ End
☐ History Column

Updatable

● Always
○ While New
○ Never

Refresh After

☐ Update
☐ Insert

Database Column

Name: CUSTOMER_ID   Type: NUMBER(6, 0)

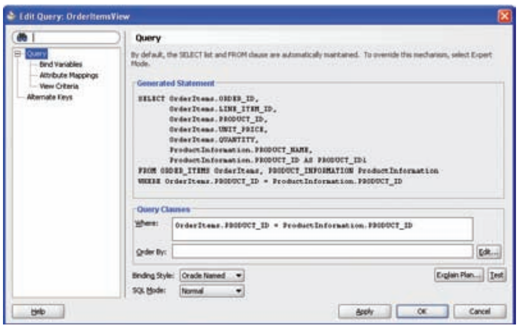
Putting validation into your business services is just as simple. If you want to set the maximum value for credit limit, check that the product id exists in the products table or ensure an email address is of the correct format, then this is the place to define those rules.

Of course, for some really complex validation cases you might want to write some code. To address this case ADF Business Components provides a method validator, that simply allows you to specify that when the framework comes to validate an attribute, it will also call your code.

View Objects

And here is some good news for all you SQL-lovers out there: the view object is just a SQL statement. When you decide that your order items view needs to pull information from the order items entity object, and that it must also use the product id to look up the product name from the products table, what you are really doing is driving the construction of a SQL statement. (See Figure 4.)

Figure 4



Query

By default, the SELECT list and FROM clause are automatically maintained. To override this mechanism, select Expert Mode.

Generated Statement

```
SELECT OrderItems.ORDER_ID,
       OrderItems.LINE_ITEM_ID,
       OrderItems.PRODUCT_ID,
       OrderItems.UNIT_PRICE,
       OrderItems.QUANTITY,
       ProductInformation.PRODUCT_NAME,
       ProductInformation.PRODUCT_ID AS PRODUCT_ID1
FROM ORDER_ITEMS OrderItems, PRODUCT_INFORMATION ProductInformation
WHERE OrderItems.PRODUCT_ID = ProductInformation.PRODUCT_ID
```

Query Clauses

Where: OrderItems.PRODUCT_ID = ProductInformation.PRODUCT_ID

Order By:

Binding Style: Oracle Named

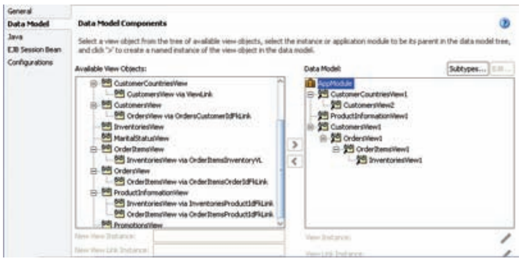
SQL Mode: Normal

You simply select the entity object on which you want to base the view object, shuttle the attributes and "Hey Presto!", a rather nifty SQL statement appears. Furthermore, just as with the entity object, if you want to earn your salary by actually writing some code, there is nothing to stop you editing that SQL. Maybe a where clause or and order by? It's up to you.

Application Module

And finally, the application module. Once you have finished building your various view objects, you can then assemble them into the "shape" your transaction really requires.

Figure 5



Select a view object from the tree of available view objects, select the instance or application module to be its parent in the data model tree, and click ">" to create a named instance of the view object in the data model.

Figure 5 shows the application module dialog with various view objects created (left hand panel). The actual data model is constructed by shuttling views from the left hand panel into the right hand panel in the hierarchy you require. So, reading from the top down, this application module contains a view of all customer countries and the customers in those countries (master/detail);, a view of all products and finally a master/detail/detail/detail of customers, orders, order items and inventories.

You can also choose to expose other functionality built into this service to anyone who consumes this service. For example, you might want to expose a method that returns the your top customer based on revenue.

Value add goodies

Of course, Oracle JDeveloper offers up a whole host of other goodies within ADF Business Components that further aid the developer in building the business services.

View objects don't have to be based just on values coming from the database. The Fusion developer can specify that a view object gets its value from a calculation, simply by setting the attributes value to an expression such as QuantityShipped * Price.

Model-driven list of values allow the developer to define that a view object attribute is backed by a list driven from a static set of data, SQL statement or view object. So, when the UI is created a combo box or list of values automatically gets hooked to that item.

And view criteria allow the developer to define useful filters that can then be exposed though a query panel to the end user.

In all these cases, the framework already knows how to do all that stuff, the developer doesn't need to code anything; instead he just tailors the framework to do it for him.

Standing on the Shoulders of Giants

The mantra behind using a framework like Oracle ADF can really be summed up by Isaac Newton's famous quote:

> **"If I have seen further than others,
> it is by standing upon
> the shoulders of giants."**

Within the fabric of Oracle ADF is woven years of experience from Oracle's development teams. Those developing on this platform benefit not only from that experience, but by using the features of the framework free themselves from the low level work and are raised to a level where the can focus on the problems of the business, not the problems of the technology.

This is the platform on which the Fusion developer stands.

Next issue, I'll be introducing a "view to a thrill". Developing the Fusion UI.

About the Author

Grant Ronald is a Group Product Manager working for Oracle's Application Development Tools group responsible for Forms and JDeveloper where he has a focus on opening up the Java platform to Oracle's current install base. Grant joined Oracle in 1997, working in Oracle support, where he headed up the Forms/Reports/Discoverer team responsible for the support of the local Oracle Support Centres throughout Europe, Middle East and Africa. Prior to Oracle, Grant worked in various development roles at EDS Defence. Grant has a BSc. in computing science and has been working in the IT Industry since 1989.

SOA Case Study

by Hugh Griffiths, Griffiths Waite

This case study covers the implementation of 'Laurus,' a Composite application based on BPM (Business Process Management) and SOA (Service Oriented Architecture). Laurus is a credit granting solution for consumer finance markets, and processes finance requests from initial application through credit scoring and underwriting to quoting and compliance, and finally to agreement completion and customer fulfillment. Laurus has been deployed at a national loan broker that has a network of branches throughout the UK and call centres in both the UK and India.

The company acts as a broker between customers seeking credit and their wide panel of lenders. They also act as a master broker enabling other brokers and introducers who do not have access to a panel of lenders to process finance requests. Improving their ability to integrate quickly and easily with multiple introducers and multiple lenders is an important part of their strategic goals.

With all cases, both prime or non-conforming, the work involves sourcing, arranging a decision-in-principle, tracking the case through to completion and beyond, covering the sales process and dealing with compliance requirements.

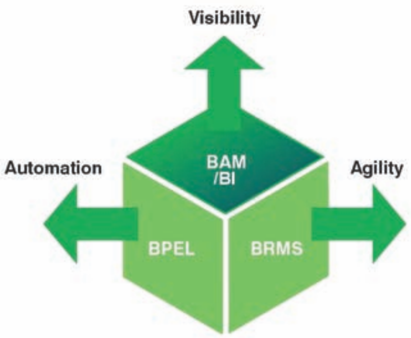
The customer's partners are demanding; introducers need to obtain instant decisions in principle, submit applications online and in bulk, and receive real time updates on the status of their cases from the allocated lenders. Lenders require the transfer of details to be seamless, and the correct checks and product selection to be completed accurately.

Technology Overview

BPM and SOA provide the ideal approach for integrating Business 2 Business processes, and services that need to support multi-channels and multiple partners.

The key products used to deliver Laurus were Oracle BPEL Process Manager, Oracle Business Intelligence and Oracle Business Rules, all working together to create and modify (using BPEL), monitor (using BI) and adjust (using Rules) complex business workflows based on a Service Oriented Architecture.

Figure 1



Oracle BPEL Process Manager

We implemented BPEL to coordinate web services and other functions. BPEL was responsible for orchestrating and managing both the low level services, and the calls to external services and partner systems. BPEL is a language for process automation and provides an open-standards framework for integrating systems and applications into complicated business processes, and provides significant benefits in terms of adaptability, ease-of-integration, portability, and interoperability.

In conjunction with Oracle BPEL Process Manager we used Oracle Business Process Analysis, (BPA) Suite as the core modelling and analysis repository for Process Improvement and BPM activities, and also to produce the blueprint designs for SOA. Oracle BPA suite provides a rich and intuitive graphical modelling environment tailored to business users, and can be used to define process maps and detailed process flows consisting of both human, automated and rule steps spanning organisational boundaries.

Oracle BPA Suite is fully integrated with Oracle BPEL Process Manager through a shared meta-model called the Process Blueprint, that promotes rapid development by generating executable code automatically from the model.

Oracle Business Rules

To make the solution more responsive to business change, we implemented key business rules such as applicant pre-qualification and product allocation in Oracle Business Rules.

Putting rules development in the hands of business users avoids the long development and testing processes typical of traditional application development, enabling business users to quickly and efficiently develop and modify rules as fast as the business requires.

Oracle Business Rules runs as a completely separate decision service, employing its own meta-model over the corporate data model. This means that it is protected from changes to underlying systems, and more importantly, the business rules are now reusable across other business systems.

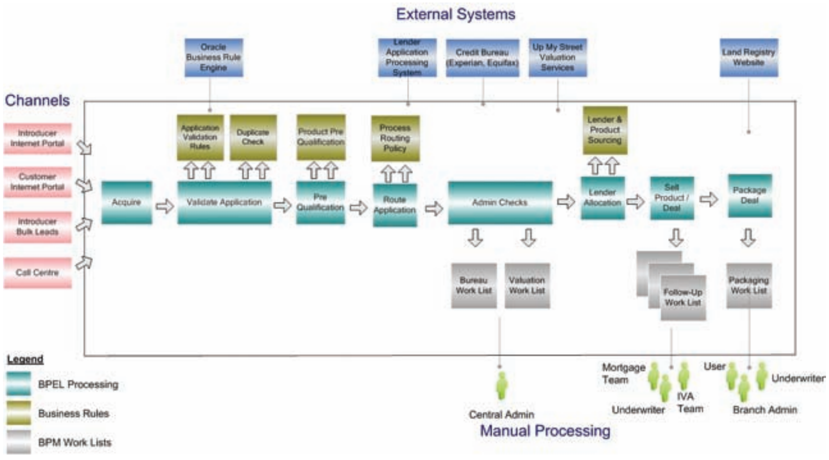
Oracle Business Intelligence

As finance requests follow BPEL, processes events are recorded in real-time. This new approach to business intelligence and data warehousing incorporates real-time feeds from application events and business processes, automated decision making, and analytics are presented not only through dashboards and reports but also integrated into applications and business processes.

This provides opportunities for continuous process improvement delivering the required level of continuity between the analysis of processes and the deployment of process improvements, enabling process enhancements to be deployed faster, more widely, with greater predictability and lower risk.

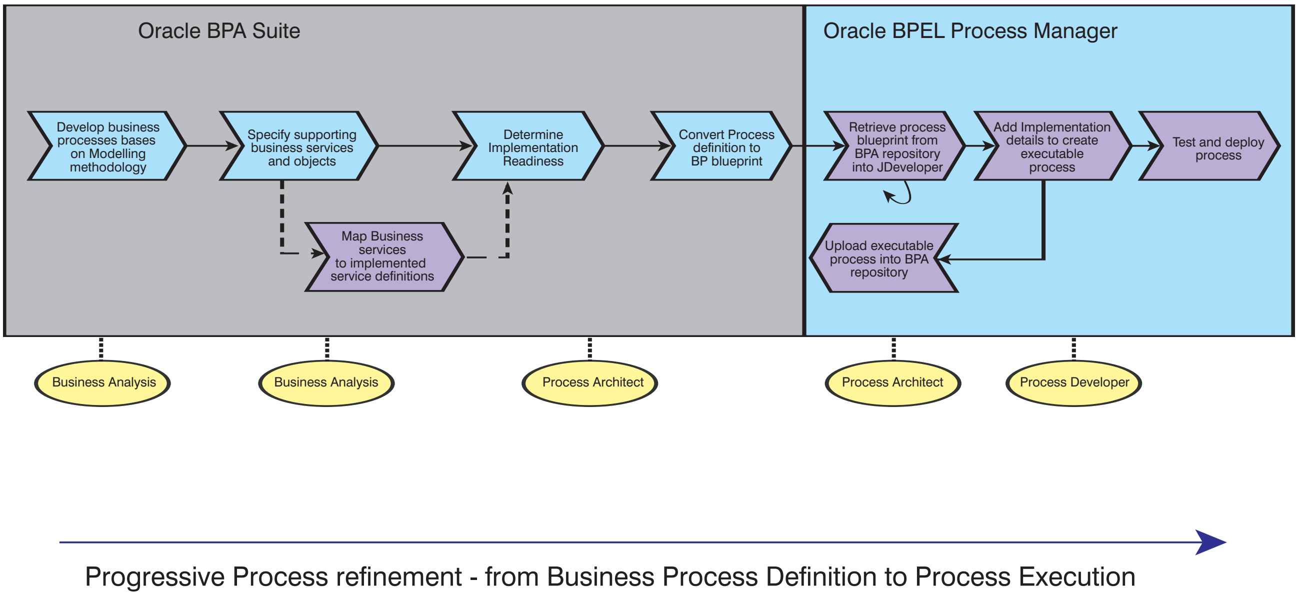
Solution Detail

Figure 2: Business Process Overview



Adopting an end-to-end "business definition to implementation" approach allows semantic validation of process models against BPMN, and then provides automatic translation into BPEL processes ensuring business processes are implemented consistently and accurately.

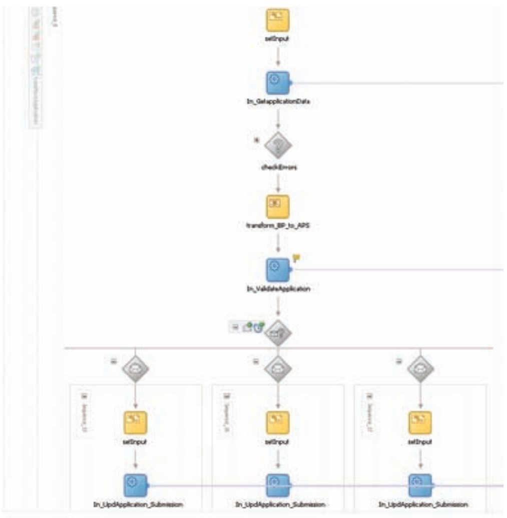
Figure 3: Business Process Definition to Process Execution


Oracle BPA Suite
Oracle BPEL Process Manager
Develop business processes bases on Modelling methodology
Specify supporting business services and objects
Determine Implementation Readiness
Convert Process definition to BP blueprint
Map Business services to implemented service definitions
Retrieve process blueprint from BPA repository into JDeveloper
Add Implementation details to create executable process
Test and deploy process
Upload executable process into BPA repository
Business Analysis
Business Analysis
Process Architect
Process Architect
Process Developer
Progressive Process refinement - from Business Process Definition to Process Execution

BPEL developers load the business process definition from Oracle BPA Suite into Oracle JDeveloper, and then add implementation details to create an executable BPEL process.

> "Oracle Business Rules runs as a completely separate decision service, employing its own meta-model over the corporate data model."

Figure 4: Oracle BPEL Business Process



By exposing the BPEL process as an external facing Web Service interface, introducers and internet portals can easily integrate with the core business process. For strategic partners, an additional Web Service definition (WSDL) and interface to a web service was provided, enabling them to directly submit bulk leads directly from their IT systems. (Figure 4.)

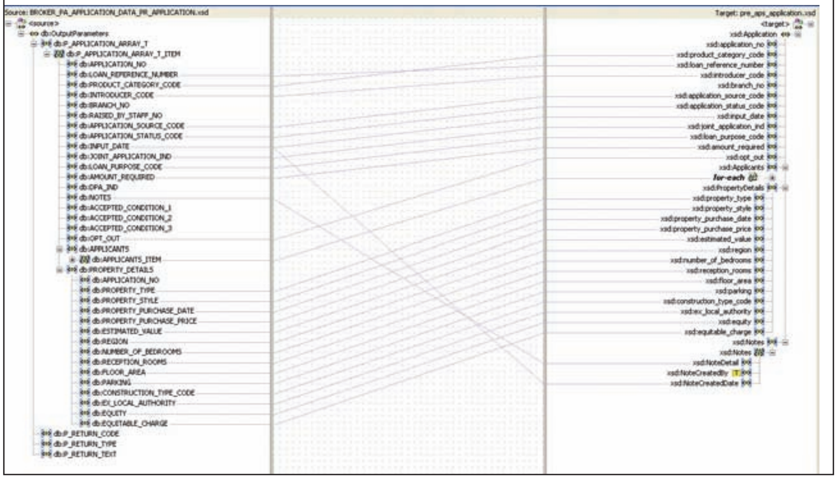
Using Oracle JDeveloper, an integrated development environment (IDE) for building service-oriented applications using the latest industry standards for Java, XML, Web services and SQL, XSLT transformations were used to transform the XML documents retrieved from strategic partners into the format used by Laurus. (Figure 5.)

Manual human intervention is a key part of the business process, and is used to facilitate the progression of an application from acquisition to completion of the loan. (Figure 6.)

Utilising Java remote APIs provided by Oracle that are packaged as part of the Oracle SOA Suite, remote Java clients were implemented to provide access into the Human Workflow tasks. The tasks APIs were implemented within Oracle Application Development Framework (ADF), and were used to extend and provide integration between the core business UI application and the BPEL process that orchestrated and controlled the flow of the application.

The key advantage of using Oracle ADF, as opposed to the Oracle BPM worklists, was the ability to produce a richer more attractive user interface and present all the relevant information to the user in a single portal. (Figure 7.)

Figure 5: Oracle BPEL Process Manager XSLT Transformation



Source: BROKER_PA_APPLICATION_DATA_PR_APPLICATION.xsd

```
<source>
  db:OutputParameters
    db:P_APPLICATION_ARRAY_T
      db:P_APPLICATION_ARRAY_T_ITEM
        db:APPLICATION_NO
        db:LOAN_REFERENCE_NUMBER
        db:PRODUCT_CATEGORY_CODE
        db:INTRODUCER_CODE
        db:BRANCH_NO
        db:RAISED_BY_STAFF_NO
        db:APPLICATION_SOURCE_CODE
        db:APPLICATION_STATUS_CODE
        db:INPUT_DATE
        db:JOINT_APPLICATION_IND
        db:LOAN_PURPOSE_CODE
        db:AMOUNT_REQUIRED
        db:DPA_IND
        db:NOTES
        db:ACCEPTED_CONDITION_1
        db:ACCEPTED_CONDITION_2
        db:ACCEPTED_CONDITION_3
        db:OPT_OUT
        db:APPLICANTS
          db:APPLICANTS_ITEM
        db:PROPERTY_DETAILS
          db:APPLICATION_NO
          db:PROPERTY_TYPE
          db:PROPERTY_STYLE
          db:PROPERTY_PURCHASE_DATE
          db:PROPERTY_PURCHASE_PRICE
          db:ESTIMATED_VALUE
          db:REGION
          db:NUMBER_OF_BEDROOMS
          db:RECEPTION_ROOMS
          db:FLOOR_AREA
          db:PARKING
          db:CONSTRUCTION_TYPE_CODE
          db:EX_LOCAL_AUTHORITY
          db:EQUITY
          db:EQUITABLE_CHARGE
    db:P_RETURN_CODE
    db:P_RETURN_TYPE
    db:P_RETURN_TEXT
```

Target: pre_aps_application.xsd

```
                              <target>
                       xsd:Application
                    xsd:application_no
            xsd:product_category_code
            xsd:loan_reference_number
                   xsd:introducer_code
                         xsd:branch_no
          xsd:application_source_code
          xsd:application_status_code
                        xsd:input_date
            xsd:joint_application_ind
                 xsd:loan_purpose_code
                   xsd:amount_required
                           xsd:opt_out
                        xsd:Applicants
                          for-each
                   xsd:PropertyDetails
                 xsd:property_type
                xsd:property_style
        xsd:property_purchase_date
       xsd:property_purchase_price
               xsd:estimated_value
                        xsd:region
            xsd:number_of_bedrooms
              xsd:reception_rooms
                    xsd:floor_area
                       xsd:parking
        xsd:construction_type_code
            xsd:ex_local_authority
                        xsd:equity
              xsd:equitable_charge
                             xsd:Notes
                           xsd:Notes
                   xsd:NoteDetail
              xsd:NoteCreatedBy
            xsd:NoteCreatedDate
```

Figure 6: Human Workflow Integrated within the BPEL Process

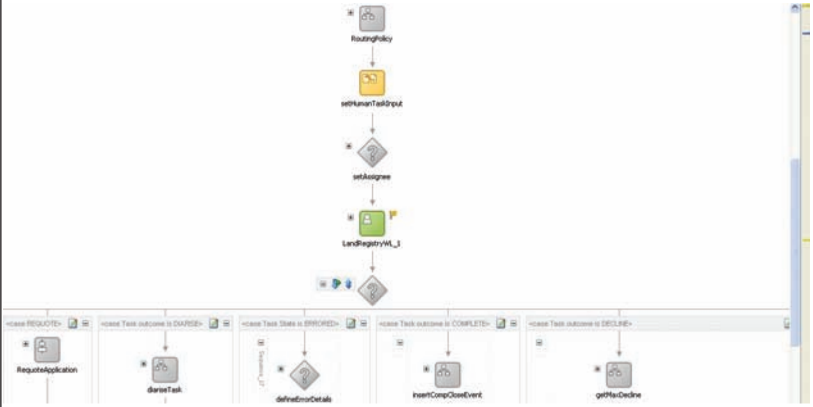
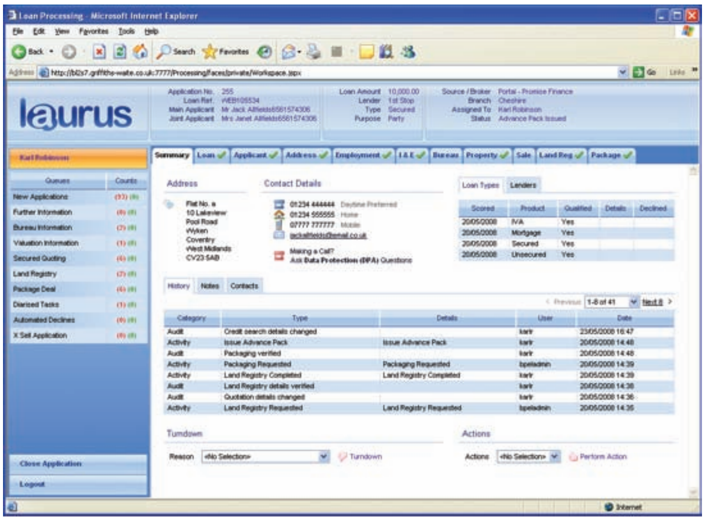


Figure 7: Oracle ADF displaying BPEL Human Workflow Tasks



http://bl2s7.griffiths-waite.co.uk:7777/Processing/faces/private/Workspace.jspx

Application No.	255	Loan Amount	10,000.00
Loan Ref.	WEB105534	Lender	1st Stop
Main Applicant	Mr Jack Allfields6561574306	Type	Secured
Joint Applicant	Mrs Janet Allfields6561574306	Purpose	Party

Source / Broker	Portal - Promise Finance
Branch	Cheshire
Assigned To	Karl Robinson
Status	Advance Pack Issued

Karl Robinson

Queues	Counts
New Applications	(93) (0)
Further Information	(0) (0)
Bureau Information	(2) (0)
Valuation Information	(1) (0)
Secured Quoting	(6) (0)
Land Registry	(2) (0)
Package Deal	(6) (0)
Diarised Tasks	(1) (0)
Automated Declines	(0) (0)
X Sell Application	(0) (0)

Address

Flat No. a
10 Lakeview
Pool Road
Wyken
Coventry
West Midlands
CV23 5AB

Contact Details

01234 444444 Daytime Preferred
01234 555555 Home
07777 777777 Mobile
jackallfields@email.co.uk

Making a Call?
Ask **Data Protection (DPA)** Questions

Scored	Product	Qualified	Details	Declined
20/05/2008	IVA	Yes		
20/05/2008	Mortgage	Yes		
20/05/2008	Secured	Yes		
20/05/2008	Unsecured	Yes		

1-8 of 41

Category	Type	Details	User	Date
Audit	Credit search details changed		karlr	23/05/2008 16:47
Activity	Issue Advance Pack	Issue Advance Pack	karlr	20/05/2008 14:48
Audit	Packaging verified		karlr	20/05/2008 14:48
Activity	Packaging Requested	Packaging Requested	bpeladmin	20/05/2008 14:39
Activity	Land Registry Completed	Land Registry Completed	karlr	20/05/2008 14:39
Audit	Land Registry details verified		karlr	20/05/2008 14:38
Audit	Quotation details changed		karlr	20/05/2008 14:36
Activity	Land Registry Requested	Land Registry Requested	bpeladmin	20/05/2008 14:35

Turndown

Reason <No Selection> Turndown

Actions

Actions <No Selection> Perform Action

> “Manual human intervention is a key part of the business process, and is used to facilitate the progression of an application from acquisition to completion of the loan.”

Key business rules from introducers, lenders, internal policy and compliance are implemented using Oracle Business Rules. Oracle Business Rules is supplied with a graphical user interface (GUI) Rule Author. For the initial creation of the rules repositories, we substituted this with a program which reads the meta model and the rules from a collection of spreadsheets and then loads these into the rule repository using Oracle's SDK. This provided the users with a more approachable user interface, and also enables us to enforce elementary validation of the rule definitions. The GUI interface was later utilised to manage these Business Policies. (Figure 8.)

By leveraging the Decision Service functionality within BPEL, a Web service that invokes specified rule repositories, we were able to incorporate the business policies within the business process, therefore providing flexibility and the ability to change decisions with the business process based upon the defined business policies. (Figure 9.)

To give immediate visibility into process performance for everyone in the organisation, it was necessary to provide application-integrated analytics to end-users that they could use as they perform their activities.

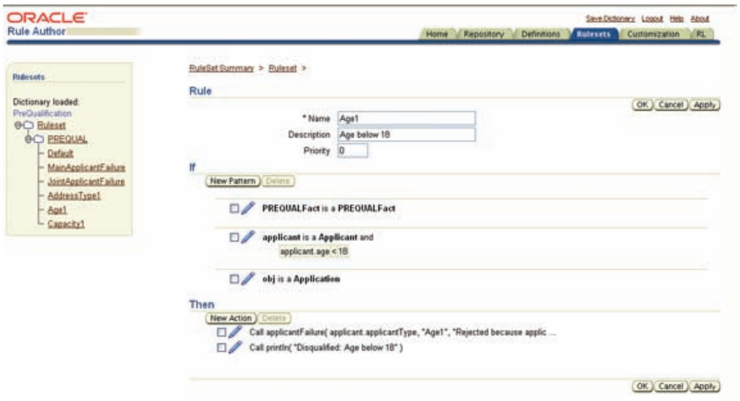
Oracle Business Intelligence enabled business users to define and monitor events, and event patterns, that occur throughout the business process. Sensors, in the form of XML representations of events, were added to the BPEL process in order to feed data to the data warehouse. Users could view dashboards in Laurus showing critical business measures, including partner service-level agreements that updated in real time and provided the ability to subsequently drill down to show the detailed information behind them. (Figure 10.)

About the Author

Hugh Griffiths is the co-founder of Griffiths Waite, an award winning Systems Integrator who specialise in the development of SOA using Oracle Fusion Middleware. He has been working with Oracle products for twenty years and provides advice and guidance to customers implementing SOA and Enterprise Architectures.

Hugh can be contacted at
hughg@griffiths-waite.co.uk

Figure 8: Oracle Business Rules



ORACLE
Rule Author

Save Dictionary Logout Help About
Home Repository Definitions Rulesets Customization RL

Rulesets

Dictionary loaded:
PreQualification
Ruleset
PREQUAL
Default
MainApplicantFailure
JointApplicantFailure
AddressType1
Age1
Capacity1

RuleSet Summary > Ruleset >

Rule

* Name: Age1
Description: Age below 18
Priority: 0

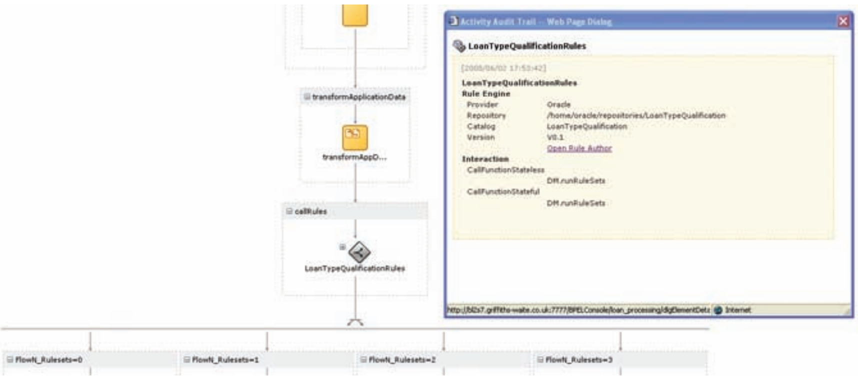
If

□ PREQUALFact is a PREQUALFact
□ applicant is a Applicant and
applicant.age < 18
□ obj is a Application

Then

□ Call applicantFailure(applicant.applicantType, "Age1", "Rejected because applic ...
□ Call println("Disqualified: Age below 18")

Figure 9: Oracle BPEL invoking Oracle Business Rules



Activity Audit Trail -- Web Page Dialog

LoanTypeQualificationRules

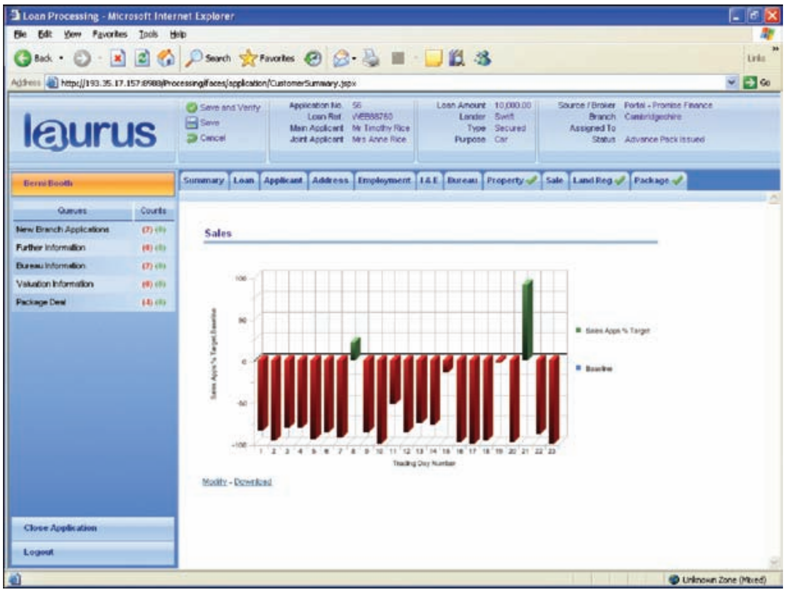
[2008/06/02 17:53:42]
LoanTypeQualificationRules
Rule Engine

Provider	Oracle
Repository	/home/oracle/repositories/LoanTypeQualification
Catalog	LoanTypeQualification
Version	V0.1
	Open Rule Author

Interaction
CallFunctionStateless
DM.runRuleSets
CallFunctionStateful
DM.runRuleSets

http://bl2s7.griffiths-waite.co.uk:7777/BPELConsole/loan_processing/dlgElementDeta Internet

Figure 10: Oracle BI Integrated into Oracle ADF



Loan Processing - Microsoft Internet Explorer

File Edit View Favorites Tools Help
Back Search Favorites Links Go

Address http://193.35.17.157:8988/Processing/faces/application/CustomerSummary.jspx

laurus
Save and Verify
Save
Cancel

Application No.	56	Loan Amount	10,000.00	Source / Broker	Portal - Promise Finance
Loan Ref	WEB88760	Lender	Swift	Branch	Cambridgeshire
Main Applicant	Mr Timothy Rice	Type	Secured	Assigned To	
Joint Applicant	Mrs Anne Rice	Purpose	Car	Status	Advance Pack Issued

Berni Booth
Summary Loan Applicant Address Employment I & E Bureau Property ✓ Sale Land Reg ✓ Package ✓

Queues	Counts
New Branch Applications	(7) (0)
Further Information	(8) (0)
Bureau Information	(7) (0)
Valuation Information	(8) (0)
Package Deal	(4) (0)

Enrich Composite Application using Oracle AIA

by Ramesh Chandra Revuru, Sierra Atlantic

A composite application is defined as an application built by combining functionality from various sources. The functionality taken from other sources could be individual functionality taken from a source or entire system or application whose output has been packaged as web services. Composite application often consists of orchestration of these functions/services drawn from different sources.

Oracle AIA stands for Oracle **A**pplication **I**ntegration **A**rchitecture. According to Oracle, AIA is the most complete integration solution for orchestrating agile, user centric business processes across your enterprise. Oracle AIA is delivered both as Foundation Pack with common objects, services, best practices and as pre-built PIP (Process Integration Packs).

We try to explain in this article how composite applications can be built quickly and easily using Oracle AIA framework. In this fast changing world, flexibility is the key to any business becoming agile in being able to offer new products or services to its customers. Companies are also grappling with challenges of being cost effective and productive while they try to be agile. Considering those inputs, we try to explore ways in which we can enrich composite applications using Oracle AIA.

Consideration

To elaborate the point, we consider the following use case. Composite Application built using Oracle Webcenter, Oracle AIA Foundation Pack & Oracle eBusiness Suite (eBS). Oracle Webcenter is used to authenticate and authorize the user. Upon authorisation, Oracle Webcenter sends the user information to Oracle eBS using AIA framework and displays all the Sales Orders that belong to the end user of the Webcenter. The flow is as follows:

1) Portlet can make use of OmniPortlet or PDK (Portal Development Kit) to invoke Web Services. OmniPortlet has the capability of using Web Services or JCA (Java Connector Architecture) as a data source. In the example below, we explore the option of using OmniPortlet to invoke the Requester ABCS (Application Business Connector Service) WSDL.

2) Portlet sends the request information in the form of an ABM (Application Business Message) to ABCS.

3) Requester ABCS then transforms the ABM to EBM (Enterprise Business Message). EBMs are based on standard EBOs (Enterprise Business Object) exposed by Oracle AIA.

4) EBM is then passed as the payload to invoke EBS (Enterprise Business Service).

5) EBS then performs the routing of message based on the logic – in this case to Oracle eBS.

6) Provider ABCS then converts the EBM to ABM and invokes the Oracle service to get Sales Orders pertaining to the end user.

7) Webcenter then displays the Sales Orders by applying the required transformation (XSLT) service to get the desired layout in the portlet.

See Figure 1, below.

Architecture

In the architecture below, we see different layers in building composite applications. Composite applications need to provide core functions like routing, error handling, security and metadata which are all part of the Oracle's AIA foundation pack. Services can be exposed and registered in a service repository called BSR (Business Service Registry). Business rules can be written using Business Rules Engine provided by Oracle AIA. Similarly, orchestration of the services/processes can be performed using Oracle BPEL PM, which again is a part of Oracle AIA infrastructure.
See Figure 2, below.

Composite application architecture relies heavily on the following two principles:

1) Reuse

2) Standardization

Figure 1: Composite Application using AIA

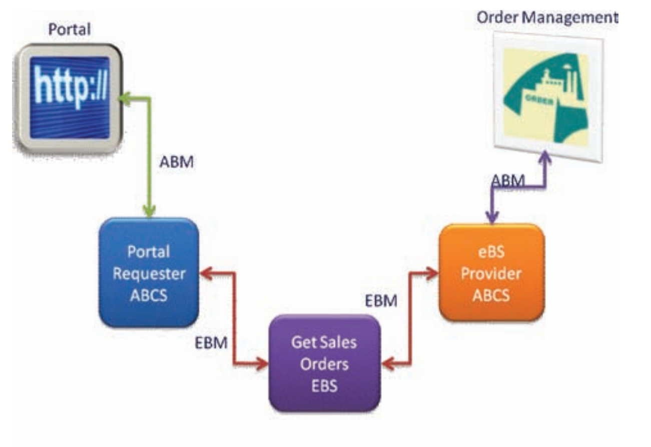
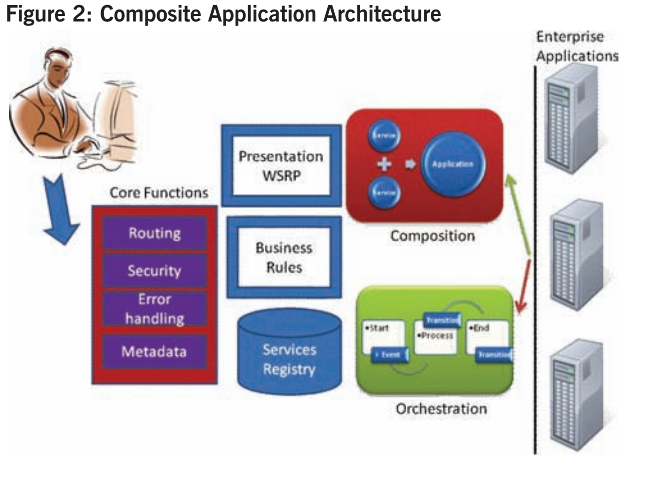


Figure 2: Composite Application Architecture



Reuse

Composite application relies a lot on reuse. Reuse could be in the form of components that are provided by vendor/application or custom built that can be reused. Reuse could also be at multiple levels in the case of a composite application. (See Table 3, right.)

Oracle AIA promotes reuse by extending prebuilt services and objects. Oracle AIA Foundation Pack 2.2 has more than 50 prebuilt business objects, more than 500 business service operations, industry specific processes catering to verticals like utilities, insurance, communications or prebuilt PIP (Process Integration Packs) between applications like Siebel CRM, Agile, Oracle eBS and SAP.

Standardization

Oracle AIA is based on industry standards. At the message level, EBOs (Enterprise Business Objects) support industry standards like OAGIS, UN/CEFACT. By using industry standard, canonical model exchange of information between application inside and outside the enterprise becomes easy.

Use of standards wherever applicable and appropriate is a good practice. Avoid creating your own custom standards. While usage of applicable standards is advised, caution has to be taken in evaluating the standards. This is relevant for those standards which are not yet mature. (See Table 4, below.)

Table 3

Integration/orchestration layer	
Name of component	**Oracle AIA/PIP Reusable Artifact**
Connector	• Technology and application adapters available - Technology – DB, File/FTP - Application – SAP, Oracle eBS
Business Objects	• Standard EBO (Enterprise Business Object) definitions available - More than 50 objects definitions available as of FP 2.2
Mapping	• As part of Pre-built PIPs (Process Integration Packs)
Error handling	• Error handling framework - Notification - Logging - Retry
Testing	• CAVS (Composite Application Validation System) framework - Integration testing - Simulation testing
Backend	
Name of component	**Oracle AIA/PIP Reusable Artifact**
Extraction Logic	• Currently only restricted to prebuilt PIPs • Composite applications relying on AIA Foundation Pack have to create on their own
CRUD Logic	• Currently only restricted to prebuilt PIPs • Composite applications relying on AIA Foundation Pack have to create on their own

Table 4

Layer	Oracle AIA Supported Standard
Data	• XML • XSLT
Messaging	• SOAP • WS-Addressing
Service description	• WSDL
Policy	• WS-Policy
Security	• WS-Security
Discovery	• UDDI • WS-Discovery
Management	• WSDM
Orchestration	• BPEL
Modeling	• BPMN

Conclusion

Oracle AIA framework provides a way of staying ahead of competition by being able to quickly develop composite applications that suit your business needs. It also provides an easy and efficient way of sharing information on the portal with users within and outside the enterprise. AIA helps organizations streamline their operations by lowering the cost of composite application development and reducing the cycle times.

About the Author

Ramesh Chandra Revuru is the global head of Integration, DW/BI & Agile at Sierra Atlantic. He has published various articles on integration, SOA and BPM and has been a speaker at Oracle User Conference, TIBCO User Conference and Oracle Open World. Before joining Sierra Atlantic, he worked with companies like Infosys, Computer Associates and Oracle."

Data Guard Broker

By Martin Bach, e-DBA Ltd

Data Guard Broker is a useful tool intended to make your life easier with Data Guard switchovers and failovers. Its configuration is not so straight forward, especially since GridControl continuously failed creating a Data Guard configuration for me. But there is a command line tool as well – dgmgrl or Data Guard Manager Line Mode coming to the rescue.

Assumptions

This document assumes you already have created a standby database, and archived redo log files are shipped between the two without problems. It also assumes your primary and standby have standby redo logs configured and you are using spfiles rather than pfiles.

Please note: this is to validate the general ability to run Data Guard only, Data Guard Broker will not use any of the parameters already set from the spfile such as log_archive_dest_n once set up, all these are taken from your (Data Guard Broker) configuration!

Implementation

For Data Guard to work, you need to set the parameter dg_broker_start to true on the primary and standby database. Wait a few seconds for the new background processes to start up.

Create the configuration

The first step is to create a configuration in Data Guard Broker. For this, you need to connect to dgmgrl as sys on the primary database, (See Figure 1):

Figure 1

```
oracle >dgmgrl /
DGMGRL for Linux: Version 10.2.0.4.0 - Production

Copyright (c) 2000, 2005, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
Connected.

DGMGRL> CREATE CONFIGURATION 'mbhtest' AS
>  PRIMARY DATABASE IS 'ora10gr2'
>  CONNECT IDENTIFIER IS ora10gr2;
```

This command creates the configuration and uses the tnsname “ora10gr2” to connect to the database. The ora10gr2 database will be registered as the primary database of that configuration. This obviously requires your tnsnames.ora to be set up properly.

Query the broker in order to see if that worked, (see Figure 2).

Note that the configuration is not yet enabled. In case of any errors, you'd see these instead of a SUCCESS message. Fast start failover is a 10g Release 2 feature allowing the fully automated failover based on certain criteria without any human intervention. It requires a so-called observer process on a third server continuously monitoring the database health.

Now let's add in the standby database, in our case it's a physical standby, (see Figure 3).

Figure 2

```
DGMGRL> show configuration

Configuration
  Name:              mbhtest
  Enabled:           NO
  Protection Mode:      MaxPerformance
  Fast-Start Failover: DISABLED
  Databases:
    ora10gr2 - Primary database

Current status for "mbhtest":
SUCCESS
```

Figure 3

```
DGMGRL> ADD DATABASE 'ora10dg' AS
>  CONNECT IDENTIFIER IS ora10dg
>  MAINTAINED AS PHYSICAL;
```

This should not take long. Let's verify the configuration, (see Figure 4).

Figure 4

```
DGMGRL> show configuration

Configuration
  Name:              mbhtest
  Enabled:           YES
  Protection Mode:      MaxPerformance
  Fast-Start Failover: DISABLED
  Databases:
    ora10gr2 - Primary database
    ora10dg  - Physical standby database

Current status for "mbhtest":
SUCCESS
```

Again, while Data Guard Broker changes the configuration you might see error messages instead of the SUCCESS. You will notice changes to the spfile in the alert.logs of both databases reflecting the new log_archive_dest_n parameters. Give it a minute to complete its work.

> “Fast start failover is a 10g Release 2 feature allowing the fully automated failover based on certain criteria without any human intervention.”

Another file worth looking at is drc<$ORACLE_SID>.log in the user_dump_dest directory – it displays a lot of information about what the broker is doing right now.

Once that has been done, enable the configuration, (see Figure 5):

Figure 5

```
DGMGRL> enable configuration
```

Check the documentation (Data Guard Broker chapter 7) in case of errors for possible resolutions. Usually your tnsnames.ora file has problems resulting on ORA-12xxx errors.

Database properties

The settings you'd have to set manually in log_archive_dest_n are controlled through database properties in Data Guard Broker. Database properties are visible by issuing a show database verbose *dbname*, the most important ones are:

- LogXptMode – specify the log transport mode (sync/async/arch)
- delayMins – specify a time delay for the application of logs
- logShipping – enable/disable log shipping to the database
- StandbyFileManagement – Auto or manual

Protection modes

Data Guard Broker uses the maximum performance protection mode by default. To upgrade the protection mode please add standby redo logs to your databases first. For maximum protection and availability you also have to use the synchronous redo transport. The following is an example of changing the protection mode from maximum performance to maximum protection, (see Figure 6).

Figure 6

```
DGMGRL> EDIT DATABASE ora10dg SET PROPERTY
'LogXptMode'='SYNC';

DGMGRL> EDIT CONFIGURATION SET PROTECTION MODE AS
MAXAVAILABILITY;
```

Networking

If not yet done so, statically add services to the listener.ora files on all hosts in your configuration. Replace primary_sid and standby_sid with the service names for your primary and standby database. Data Guard Broker tries to restart the databases during switchover and failover – obviously once a database is shut down it is no longer registered with the listener and can't be automatically started. From experience the following services should be registered, (see Figure 7).

Figure 7

```
<primary_sid>_XPT
<primary_sid>_DGMGRL
<primary_sid>
<standby_sid>_XPT
<standby_sid>_DGMGRL
<standby_sid>
```

If you have multiple standby databases, make sure to add them all in, the above example assumes a 2 database setup. If you fail to add these parameters the switchover is most likely to succeed anyway, but you will have to manually start the databases in their correct role.

Switchover

Remember that a switchover is the graceful role transition, not to be confused with a failover. Connect to DGMGRL and check the configuration, (see Figure 8).

Figure 8

```
DGMGRL> show configuration

Configuration
  Name:              mbhtest
  Enabled:            YES
  Protection Mode:     MaxPerformance
  Fast-Start Failover: DISABLED
  Databases:
    ora10gr2 - Primary database
    ora10dg  - Physical standby database

Current status for "mbhtest":
SUCCESS
```

> "...once a database is shut down it is no longer registered with the listener and can't be automatically started."

As you can see ora10dg is the primary but it should run in standby role. Let's do the switchover. It pays off to have two more putty sessions open to view the contents of the Alert logs flying by. Data Guard Broker will kill any running session on the primary, so advise all users beforehand, (see Figure 9).

> "It pays off to have two more putty sessions open to view the contents of the Alert logs flying by."

Figure 9

```
DGMGRL> switchover to ora10dg
Performing switchover NOW, please wait...
Operation requires shutdown of instance "ora10gr2" on database "ora10gr2"
Shutting down instance "ora10gr2"...

Database dismounted.
ORACLE instance shut down.
Operation requires shutdown of instance "ora10dg" on database "ora10dg"
Shutting down instance "ora10dg"...
ORA-01109: database not open

Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "ora10gr2" on database "ora10gr2"
Starting instance "ora10gr2"...
ORACLE instance started.
Database mounted.
Operation requires startup of instance "ora10dg" on database "ora10dg"
Starting instance "ora10dg"...
ORACLE instance started.
Database mounted.
Switchover succeeded, new primary is "ora10dg"
DGMGRL>
```

We can double check this by querying the configuration, (see Figure 10).

Figure 10

```
DGMGRL> show configuration

Configuration
  Name:              mbhtest
  Enabled:           YES
  Protection Mode:     MaxPerformance
  Fast-Start Failover: DISABLED
  Databases:
    ora10gr2 - Physical standby database
    ora10dg  - Primary database

Current status for "mbhtest":
SUCCESS
```

As you can see the new primary database is ora10dg whereas the former primary is now the standby database.

Failover

Failover will be required when your primary database is damaged beyond repair. That's the moment of truth!

The example assumes you have flashback logging enabled on both databases, if you haven't done so you need to rebuild the standby as in the old 9i days.

How to enable flashback logging

You need to use a flash_recovery_area to store the flashback log files, for example in /u01/app/oracle/flash_recovery_area. Enable flashback logging on primary and all standby databases as follows, (see Figure 11).

Figure 11

```
$ sqlplus / as sysdba
SQL> shutdown immediate
(database shuts down)
SQL> startup mount
(database mounts)
SQL> alter database set db_recovery_file_dest_size=30G;

Database altered

SQL> alter database set db_recovery_file_dest = '/u01/flash_recovery_area'

Database altered

SQL> alter database flashback on;

Database altered

SQL> select name,database_role,force_logging,flashback_on from v$database

NAME       DATABASE_ROLE        FOR    FLASHBACK_ON
---------- -------------------- ------ ------------------
ORA10GR2   PHYSICAL STANDBY     YES    YES
```

Figure 12

```
DGMGRL> show configuration

Configuration
  Name:            mbhtest
  Enabled:          YES
  Protection Mode:     MaxProtection
  Fast-Start Failover: DISABLED
  Databases:
    ora10gr2 - Physical standby database
    ora10dg  - Primary database
```

Use the same strategy to enable flashback logging on the primary system.

Perform failover

Connect to the physical database to become the new primary. (see Figure 12).

Note that we are still in the state as by the end of the previous example, i.e. ora10dg is running in the primary role, whereas ora10dgr2 is the standby database, (see Figure 13).

You now have a new primary database! Note that the physical standby is disabled, but still part of the configuration.

Figure 13

```
Current status for "mbhtest":
SUCCESS

DGMGRL> failover to ora10gr2
Performing failover NOW, please wait...
Failover succeeded, new primary is "ora10gr2"
DGMGRL> show configuration

Configuration
  Name:            mbhtest
  Enabled:          YES
  Protection Mode:     MaxPerformance
  Fast-Start Failover: DISABLED
  Databases:
    ora10gr2 - Primary database
    ora10dg  - Physical standby database (disabled)

Current status for "mbhtest":
SUCCESS
```

Users should be able to resume work on the new primary database but the former primary database is essentially ruined.

We now need to do some extra work on the old primary to transform it into a standby system and restart redo apply. Begin by starting the old primary into mount mode. Connect into DGMGRL as sys on the new primary.

Instead of a string of manual steps, reinstating the database is a one liner in dgmgrl, (see Figure 14).

I really like the feature, especially if you consider what is going on in the background. Verify the success, (see Figure 15).

If you had a higher protection mode then upgrade it again, refer to the earlier part of this article on how to do this.

Figure 14

```
DGMGRL> reinstate database ora10dg;
Reinstating database "ora10dg", please wait...
Operation requires shutdown of instance "ora10dg" on database "ora10dg"
Shutting down instance "ora10dg"...
ORA-01109: database not open

Database dismounted.
ORACLE instance shut down.
Operation requires startup of instance "ora10dg" on database "ora10dg"
Starting instance "ora10dg"...
ORACLE instance started.
Database mounted.
Continuing to reinstate database "ora10dg" ...
Reinstatement of database "ora10dg" succeeded
```

Figure 15

```
DGMGRL> show configuration

Configuration
  Name:            mbhtest
  Enabled:          YES
  Protection Mode:     MaxPerformance
  Fast-Start Failover: DISABLED
  Databases:
    ora10gr2 - Primary database
    ora10dg  - Physical standby database

Current status for "mbhtest":
SUCCESS
```

Martin Bach has six years of experience as a DBA and is Oracle Certified Master for Database 10g Release 2. Martin's experience covers many market sectors, gained in a number of European countries. His main focus is on mission critical systems, high availability and disaster recovery.

Martin can be contacted at: martin.bach@e-dba.com

Pass it On: Four Little-Known Secrets to Help Increase Your Support Efficiency

By Kate Cumner, Oracle Global Customer Support

If you're looking for a way to save time and increase your efficiency with Oracle Support, there's never been a better time. Take advantage of these four valuable resources, designed to help you maintain your system's peak performance and resolve issues with greater proficiency:

- **Oracle configuration manager** is included with Oracle Premier Support. OCM is a proactive, automated support capability that makes it easier to track, manage and support Oracle configurations while reducing the risk of unplanned system downtime

  OCM provides a complete, dynamic view of current configurations – including application, middleware and database versions, plus operating system and hardware platform details. This information is displayed in a new, customisable, easy-to-use dashboard – you define the filters and views to track the configuration information that matters most to your business.

  To get started with OCM, log on to *My Oracle Support* (formerly *MetaLink*); select the 'Collector' Tab. You will find all of the documentation and training to get started with OCM.

- **Demo It To Oracle (DITO)** provides you with the ability to record all screen and audio activity on your computer into standard AVI video files. DITO enables you to demonstrate your issue for Oracle Support just like you would if you were using an Oracle Web Conference; to your advantage, you can demonstrate your issue offline, at your convenience, without the need to arrange for the presence of a support engineer. Recordings can be made available to Oracle Support by uploading them to your SR in *My Oracle Support*

  Providing Oracle Support a recording of your issue can be of tremendous help to the Support Engineer working on your issue, dramatically reducing the average time to resolve the issue at hand. To receive more information on DITO, log on to *My Oracle Support* and search for note 11.1 or search the keyword "DITO". Alternatively, follow this link (that requires *My Oracle Support* login) –

  https://metalink.oracle.com/metalink/plsql/f?p=130:14:4529051465680921656::::p14_database_id,p14_docid,p14_show_header,p14_show_help,p14_black_frame,p14_font:NOT,11.1,1,1,1,helvetica

- **Speed Training** is a new series of three- to five-minute recorded training sessions that provide beneficial tips and information to help with specific situations you may encounter with Oracle Global Customer Support. The first recording in the Speed Training series is "How to Escalate a Service Request within Oracle Support", and can be accessed by logging on to *My Oracle Support* (formerly Oracle's *MetaLink*); selecting the 'Dashboard' Tab, followed by 'News, Events and Training' in the 'Getting Started' region. Please be sure to check this page often, as new recordings will be provided as they become available. Alternatively, follow this link (which requires *My Oracle Support* login) –

  https://metalink.oracle.com/metalink/plsql/f?p=130:14:4529051465680921656::::p14_database_id,p14_docid,p14_show_header,p14_show_help,p14_black_frame,p14_font:NOT,%20555057.1,1,0,1,helvetica

- **Guided Learning Paths** are longer (20-60 minute) training sessions, organised to easily connect you to beneficial information about online support tools and processes. Oracle Guided Learning can be attended through a live Web cast or online recordings. A variety of topics are available to help you get the most from your support investment, including:

A) Effective use of online support portals: *My Oracle Support* and Customer Connection

B) Using Online support resources to reduce your time to resolution

C) Support diagnostic training including: E-Business Suite and Database diagnostics, Oracle configuration manager, Maintenance Wizard and more.

To get started on your own Guided Learning Path, log onto *My Oracle Support*; select the 'Dashboard' Tab, followed by 'News, Events & Training'. Guided Learning Paths can be found Under 'Programs and Training'. Alternatively, follow this link (which requires *My Oracle Support* login) –

https://metalink.oracle.com/metalink/plsql/f?p=130:14:3359573445576131389::::p14_database_id,p14_docid,p14_show_header,p14_show_help,p14_black_frame,p14_font:NOT,%20473575.1,1,0,1,helvetica

UKOUG Conference Series 2009

The annual ‘one stop shop’ UKOUG Conference & Exhibition has been replaced with a series of Conferences under the UKOUG Conference Series umbrella with the strapline – **One Conference, many events**.

UKOUG Conference Series is a series of community specific events for industry professionals who want to know about new initiatives, solutions, products and services in relation to their Oracle investments whilst networking and sharing knowledge and experience with peers. UKOUG Conference Series is where you can learn new ways to improve productivity, profitability, efficiency and growth for your organisation. Professionals from all industries and all levels of the organisation will benefit from a host of topics covered at each event.

Top reasons to attend UKOUG Conference Series events:

- UKOUG Conference Series provides you with comprehensive educational sessions, networking, knowledge sharing and top Oracle partner solutions that can't be found anywhere else under one roof
- Attend interactive masterclasses and panel sessions. Find practical tips and techniques to boost the productivity of both you and your contacts
- Oracle Product Roadmaps. See the latest enhancements and plans for future developments to Oracle solutions
- Listen to Oracle customers talk about their problem-solving experiences and take advantage of the valuable knowledge they will share
- Discover a world of solutions in our exhibition areas. Meet Oracle experts and top Oracle partners and solution providers under one roof and get live demonstrations on new and existing solutions
- Networking. Share new ideas and best practices with your peers at various networking functions and learn how they are using Oracle solutions today to help solve the challenges their companies are facing
- Talk with Oracle experts to get advice and solutions for your immediate challenges
- Vision and strategy. Hear from Oracle’s leadership team about the future of Oracle Technology

UKOUG Conference Series is a unique opportunity for professionals who implement, administer, support or use Oracle solutions to meet with industry experts, discuss best practices with peers, provide feedback to Oracle product developers, or simply evaluate the latest solutions from Oracle partners. Each Conference attendee – no matter what their level in their organisation – will walk away from these high-value Conferences better informed, better skilled, better connected, and better prepared to make a difference.

UKOUG Conference Series 2009 calendar

22nd April	UKOUG Conferences Series Ireland 2009	1 day
19th – 20th May	UKOUG Conference Series PeopleSoft 2009	2 days
23rd June	UKOUG Conference Series Scotland 2009	1 day
24th – 25th June	UKOUG Conference Series Hyperion 2009	2 days
30th June	UKOUG Conference Series Siebel 2009	1 day
11th – 12th November	UKOUG Conferences Series JD Edwards 2009	2 days
30th Nov – 2nd December	UKOUG Conference Series Technology & E-Business Suite 2009	3 days

UKOUG membership entitlement

Your UKOUG membership entitles you and your colleagues to a number of free Conference day places – see table below for details. This entitlement remains the same.

These free Conference places can be used in any combination at any of the UKOUG Conference Series 2009 events. Please note that to attend the events in Ireland and Scotland you will need to use a free Conference day place.

Those of you with a Bronze membership will need to either upgrade your membership or pay for a Conference pass for each event your wish to attend. To find out which option will be more cost effective for you, please contact the membership team at: info@ukoug.org

Membership entitlements

Platinum Unlimited membership	5 days
Partner membership	4 days
Gold Corporate membership	4 days
Silver Associate membership	2 days
Bronze Affiliate membership	0 days

Not a member?

It’s easy to join. Call the membership department on +44 (0)20 8545 9670 to discuss your membership requirements.

Meet the new UKOUG Directors

The UKOUG Board of Directors are responsible for ensuring that the group is performing and working in the right direction to fulfil the needs of our members. UKOUG would like to pass on a warm welcome to the newly appointed members of the Board. Here's a little more about them.

Ari Aaltonen – Head of Oracle Solutions Practice :: Cognizant

I have more than 10 years experience developing, implementing, supporting, troubleshooting and using Oracle products including Oracle Ebiz, PeopleSoft, JD Edwards, Siebel and Oracle Technology and Middleware. Currently I am heading Oracle Solutions Practise at Cognizant, my practice implements Oracle Applications, Oracle Technology, JDE Applications and PeopleSoft application solutions.

I have worked for Oracle for nine years at Oracle, supporting Oracle Customers, worked as a SIG Buddy (User Group Ambassador) Public Sector Local Government SIG and supported Public Sector Central Government SIG. After leaving Oracle two years ago I have remained an active participant to SIGs and other UKOUG events contributing to them by presenting new ways of supporting and implementing Oracle products. I am also working with Oracle Partner Network in Oracle products advisory board providing input to Oracle in future product engagements and support related issues.

My main emphasis will be improving Oracle customers experience using Oracle products. This can be achieved by passionately pushing Oracle to improve its communication between Oracle and its users and driving better support to Oracle customers. UKOUG should maintain and elevate the communication and lobbying between Oracle and its product users, covering all levels of customer organisation from product user to IT managers and CIOs.

Lisa Dobson

I have been actively involved with UKOUG for many years, initially as a member of the community, before becoming a Volunteer in 2005 with the creation of the Oracle on Windows SIG, of which I am co-chair. I am also a regular speaker at many of the UKOUG events, including the Annual Conference, OUG Scotland Conference, and OUG Ireland Conference. Being a 'Northern Lass' myself, I have campaigned endlessly to raise the presence of the UK Oracle User Group within the North of England and have been the driving force behind the Northern Server Technology Day. The success of this event over the last three years has been a big achievement and the delegate feedback has been outstanding.

I am also known within the community for the work I have done in ensuring that beginners to Oracle Technology are well represented. It began with the creation of the 'Oracle Newbies Blog' in 2005, and continues with the new 'Back to Basics' event and the commitment to ensuring the provision of beginner level content on the Annual Conference Agenda. Delegate feedback has consistently congratulated the UKOUG on striving to meet the needs of all members of the Oracle Community.

I will bring boundless energy and enthusiasm to making sure UKOUG continues to work to meet the needs of its members. We need to continue to grow and evaluate how we can provide you with the best possible service, whatever your skill level or location, and regardless of what has traditionally been done in the past.

Julian Dyke – Database Consultant :: JulianDyke.com Limited

I am an independent consultant specialising in performance tuning specialising in Oracle Real Application Clusters environments. I have regular customers across a variety of industry sectors including manufacturing, finance, public sector, broadcasting, media, gambling, health, logistics, insurance and retailing. I have worked with Oracle databases for almost 20 years as a developer, database administrator and consultant. I have worked with every release since Oracle 6.

Over the past eight years I have presented regularly at SIGs, seminars and conferences across the world on Oracle internal and performance related topics including an Advanced RAC event in the Oracle University Celebrity Seminar series I have written a book with Steve Shaw of Intel called Pro Oracle Database 10g RAC on Linux which was published in 2006. I am also a member of the Oak Table network.

I was instrumental in founding the UKOUG RAC & HA SIG and have been the elected Chair of the SIG for the past five years. Although I come from a Server Technology background I recognise that the UKOUG represents a wide constituency of members and undertake to ensure the interests of all members are represented in a fair and balanced way. We need to recognize that, in addition to the membership fees, there are indirect costs of attending UKOUG events including time and expenses. We therefore need to maximise the value of our events to members.

Sue Yates – Independent Consultant :: Personal membership

I first learned how to spell Oracle in 1990 when I joined a central Oracle support group at British Airways. While there I pioneered the use of Oracle Case and Designer at the company and was involved in beta programmes for Oracle UK. I have continued to champion the use of Oracle Designer and modelling tools in general through my career as a business consultant, first at CACI Limited and most recently at EDS Consulting Services.

I have been an active participant in UKOUG events since the early 1990s, spent eight years as Co-Chair of the Modelling, Analysis and Design SIG and have regularly contributed to paper judging and agenda selection days for conference. My most recent 'day job' as a consultant for EDS Consulting Services was a hands-on, client facing role, often acting on behalf of users of a variety of modelling and design tools, including the Oracle suite of tools. Combining this significant user experience with my knowledge of the UKOUG from the perspective of both member and volunteer gives me a great insight to not only what the user group can offer but also what members would like and expect.

I strongly believe that membership of a user group has to be an enjoyable experience with events geared to members needs and providing inspiration to participate and learn. UKOUG provides members with unparalleled networking, learning and sharing opportunities – considered leadership and innovative thought are needed to ensure that UKOUG continues to flourish.

The long-standing UKOUG Directors are Ronan Miles, Debra Lilley, Tracey Bleakley, Carl Dudley, Jonathan Lewis, Anne Power, Peter Dobson and David Rowntree. All Directors can be contacted at: director@ukoug.org

Debra's diary

Gosh, when I look back at my last entry, it was written in San Francisco after Oracle Open World and today as I write the next instalment I am on a flight back to San Francisco for The International Oracle Usersgroups Community Summit.

Thinking about Open World, I was so shocked at how big the Elevator Experience became and when Oracle published their Top Five Things from Oracle Open World, I was No. 3, it seems they had been looking for a story to communicate Web 2.0 with and blogging from an elevator gave them all that and more! UKOUG are again the ones Oracle turn to.

The Summit is an annual event when Users Group Presidents from around the world come to Oracle HQ for product briefing from top Oracle Executives. There will be about one hundred Presidents and the Executives tend to be more open with User Group Leaders. Ronan, James and I are attending. Most of the big Users Groups have Executives like our James and it is good for them to share best practice.

Ronan and I attended the EOUC Summit in November which was held here in the UK. Not all users groups from EMEA can travel to the main IOUC summit, so there is a much smaller event held in the region.

Staff at UKOUG are professionals in their area and in November for the fourth consecutive year our events team organised the Oracle Partner Network OPN day event, this time at the Sofitel in Terminal Five Heathrow. This gives UKOUG valuable revenue and the staff the opportunity to demonstrate their wider abilities.

Fiona Martin (our Partner Forum Chair), John Rodway (who leads the Acquisitions Working Party)and I spoke at an Oracle Mergers and Acquisitions Partner Event just before the conference where Oracle took some of the best practice found in our Partner Forum into EMEA. It was great to hear Oracle tell their partners just how much UKOUG has to offer.

I also talked last time about the need to restructure our Conference, and I hope you have seen our announcements about the Conference Series on page 43.

We had a Conference outside the traditional Birmingham event for JD Edwards in November at Ascot and it was a great success, (see photograph), not only did the numbers impress us but some delegates had not been to a User Group event since before the PeopleSoft acquisition of JD Edwards. This event has given us the confidence to go forward with our Conference Series, it is a new concept and we will learn on the way but together we will be able to add value to our offering and keep within the budgets we like everyone else see squeezed. Our Birmingham conference is not going away and it will still be the first week of December, 30th November – 2nd December, but it will go back to a more manageable three days (for staff and directors we were there for 6 or 7 days). That means that we will be in a constant state of conference planning all year and we have already started.

UKOUG 2008 was our 25th birthday and did we celebrate! The conference, as I said, was five days with people attending whatever was relevant to them, it was not intended for delegates to attend all five days, but many did. As ever, we had lots of speakers and delegates from overseas, and I spoke to many of them who said UKOUG is the 'must attend' conference and of more value to them than Open World. This time they had the added benefit of a weak pound.

Each year at conference we have director elections; there are twelve directors,each with a two year term, so normally there are six vacancies. This year five of the incumbent directors stood along with a similar number of new candidates and I would like to welcome Julian Dyke, Lisa Dobson and Sue Yates (all previous SIG Chairs) and Ari Aaltonen, who was a SIG buddy when he was at Oracle, to the board. This is the first time in the eight years I have been on the board that we have had such a change in directors but change is good and it shows that members are taking more of an interest in the directors that lead the community.

A few years back, I achieved Chartered Director with the IOD and one thing I do each year for new directors is hold a day training on the role and obligations of a director, this was my biggest class to date. I am sure you will join me in congratulating them and in thanking David Kurtz, Peter Phillips and Roy Bailey for their contributions to the board in the past and their continued support of UKOUG.

If you follow this diary or my blog: http://www.debrasoracle.blogspot.com you will know I lead the Product Development Committee at the IOUC, and do so to ensure UKOUG are heard at a Global level and that we have direct communication from development. I was very proud therefore to be awarded the Oracle Magazine Editor's Award for 'User Group Evangelist of the Year' 2008. Whilst this is a great honour, it is interesting to note that I am simply following a great tradition, within the current board of directors there are four previous winners of Editors Choice Awards, Jonathan Lewis (Author of the Year), Carl Dudley (DBA of the Year), Ronan Miles (IT Manager of the Year) and our new editor Mark Rittman (Oracle ACE of the Year), as their editor said,

> "It speaks very well for the membership, I think, that the skill sets are so diverse."

By the time you read this, Collaborate 09 will be upon us: this is the largest User group conference in the US. I, along with many UKOUG volunteers, will be speaking and flying the flag for UKOUG. There will be another Fusion Inner Circle where I will get the latest information and hopefully we will soon be able to give you more details.

Debra Lilley is a Principal Business Consultant with Fujitsu Services. She is both an Oracle Certified Professional (Applications) and Oracle Master (IT Professional). Debra has been a UKOUG director since 2004 and is currently Deputy Chairman. She is also responsible for the Product Development Committees at both EMEA and International Oracle User Community.

Blogspot

What is a Customisation?

Having recently started a new position as Project Manager on a well established implementation, I decided that I needed to get up to speed with the project history, future requirements, and the solutions being delivered. After reading many lengthy process documents, it became clear that this was by no means a straightforward project and that many of the client's legacy processes needing to be replicated in Oracle have required additional development to achieve the project's success criteria.

When I requested a comprehensive list of customisations across all modules, the response I received from both the client and other consultants was, "we only have a few customisations – everything else is an add-on." My immediate thought was: what is an add-on if it's not a customisation?

When I challenged their interpretation of a customisation and listed requirements such as the ability to upload ADI journals across sets of books, ability to automatically reconcile subledger activity to GL transactions via custom tables, and the ability for specific users to post 'retro' journals when a period is closed, I was told that they are not customisations because we are not changing standard code. Without having seen the technical specs for these "add-ons" it's difficult for me to argue whether a code change was required or not (although I suspect it was in some cases). However, the misconception that something is only a customisation if a code change is required needs to be rectified.

Being a functional specialist with some technical understanding, I decided to research this topic further in order to prove that a customisation is not purely defined as something that requires a change to standard code. According to many Oracle professionals specialising in customisations, they are commonly defined as follows:

Extensions: Components developed using a variety of tools such as Oracle Forms, Oracle Reports, and Oracle Workflow, which provide new functionality without changing the core application structure. For example, creating a new report or modifying workflow to meet specific requirements can be achieved using these tools to create data extensions. However, because the new report or tailored workflow is not standard/pre-packaged, this development is still classified as a customisation.

Modifications: Changes made to standard code or database objects in the E-Business Suite. This can include adding or modifying tables, columns, stored procedures and triggers which are part of the E-Business Suite application schema.

In the majority of cases, Oracle customers generally develop 'extensions' rather than 'modifications' as this is the safest way to provide a solution without changing the core data structure (but may involve creating new objects rather than modifying standard objects). However, extreme cases may require a modification which, in addition to introducing potential risks to data integrity, is something which consequently presents many challenges during an upgrade. How a customer chooses to customise / modify the applications also affects product licensing and support.

So in conclusion, I prefer to define a customisation as a function (i.e. form), subfunction, process (including interfaces), report, or workflow that:

- Is not standard functionality.
- Cannot be provided by configuration alone.
- Requires development using tools provided, or recommended, by Oracle.

If anyone disagrees, I am very interested to hear your views.

David Kelly has over 10 years of Oracle E-Business implementation experience spanning financial services, manufacturing and retail industries across Europe, Middle East and Africa (EMEA), North America, and Asia Pacific.

Content supplied by:

Global Oracle Contractors Network

ORACLE
contractors.com

UKOUG calendar of events 2009

January

20th **UNIX SIG Meeting**, London

22nd **Coherence SIG Meeting**, London

February

4th **Oracle Financials SIG Meeting**, London

5th **Public Sector HCM Customer Forum**, Midlands

10th **Business Intelligence & Reporting Tools SIG Meeting**, London

11th **RAC & HA SIG Meeting**, Slough

12th **JD Edwards EnterpriseOne SIG Meeting**, Slough

26th **HCM SIG Meeting**, Slough

March

3rd **Local Government Applications SIG Meeting**, London

3rd **Oracle Government User Group SIG Meeting**, London

3rd **Supply Chain & Manufacturing SIG Meeting**, Midlands

4th **App Server & Middleware SIG Meeting & Apps DBA for OEBS SIG Meeting**, Slough

10th **Hyperion Essbase Meeting**, London

10th **Hyperion Planning Meeting**, London

11th **Local Government CRM Customer Forum**, London

17th **DBMS SIG Meeting**, Slough

17th **JD Edwards World SIG Meeting**, London

24th **Criminal Justice SIG**, London/Reading

26th **Development Engineering, Modelling, Analysis & Design & Oracle and .NET Combined SIG Meeting**, Slough

31st **Oracle Spatial SIG Meeting**, Reading/London

April

2nd **Management & Infrastructure SIG Meeting**, London

22nd **UKOUG Conference Series 2009 Ireland**, Dublin

23rd **Education & Research SIG Meeting**, London

23rd **Scottish DBA SIG Meeting**, Edinburgh

28th **Northern Server Technology Day**, York

28th **UKOUG Partner Forum**, London

29th **Oracle Projects SIG Meeting**, London

May

12th **Stellent SIG Meeting**, London

12th **Oracle on Windows SIG Meeting**, Reading

19th – 20th **UKOUG Conference Series PeopleSoft 2009**, London

20th **UNIX SIG Meeting**, Midlands

21st **Oracle Financials SIG Meeting**, Manchester

21st **Public Sector HCM Customer Forum**, Midlands

June

4th **SIG Volunteers Meeting**, Birmingham

10th **JD Edwards Combined SIG Meeting**, Slough

11th **HCM SIG Meeting**, London

16th **RAC & HA SIG Meeting**, London

24th – 25th **UKOUG Conference Series Hyperion & BI 2009**, London

18th **Business Intelligence & Reporting Tools SIG Meeting**, London

23rd **UKOUG Conference Series Scotland 2009**, Glasgow

24th **App Server & Middleware SIG Meeting**, London

30th **UKOUG Conference Series Siebel 2009**, Ascot

July

2nd **DBMS SIG Meeting**, London

14th **Development Engineering SIG Meeting**, Midlands

15th **UKOUG Partner Forum**, London

August

No events in August

September

8th **UNIX SIG Meeting**, Slough

9th **Irish HCM SIG Meeting**, Dublin

10th **Oracle Financials SIG Meeting**, London

17th **Business Intelligence & Reporting Tools SIG Meeting**, London

22nd **App Server & Middleware SIG Meeting**, Midlands

22nd **Oracle Spatial SIG Meeting**, Midlands

24th **Public Sector HCM Customer Forum**, Midlands

24th **Oracle on Windows SIG**, Reading

29th **Oracle Projects SIG Meeting**, Midlands

29th **RAC & HA SIG Meeting**, London

October

TBC **Public Sector Combined event**

1st **Local Government CRM Customer Forum**, Midlands

1st **Management & Infrastructure SIG Meeting**, London

6th **Supply Chain & Manufacturing SIG Meeting**, Midlands

6th **Apps DBA for OEBS SIG Meeting**, London

7th **Oracle and .NET SIG Meeting**, London

8th **Local Government Applications SIG Meeting**, London

13th **Criminal Justice SIG Meeting**, London

13th **Education & Research SIG Meeting**, Midlands

14th **Stellent SIG Meeting**, London

15th **Development Engineering SIG Meeting**, London

15th **DBMS SIG**, Midlands

21st **Scottish BI SIG Meeting**, Edinburgh

22nd **HCM SIG Meeting**, Midlands

November

3rd **Oracle Government User Group SIG Meeting**, London

5th **Siebel SIG Meeting,** Reading

11th **Scottish Developer SIG Meeting,** Edinburgh

11th-12th **UKOUG Conference Series JD Edwards 2009**, Twickenham

30th **UKOUG Conference Series Technology & E-Business Suite 2009,** Birmingham

December

1st – 2nd **UKOUG Conference Series Technology & E-Business Suite 2009**, Birmingham

2nd **UKOUG Partner Forum**, Birmingham

All event dates are subject to change