



By Grant Ronald

ORACLE® TOOLS DIRECTION

Oracle Forms

MODERNIZING ORACLE FORMS: FROM CLIENT/SERVER TO SOA

As I sit writing this article on one of the final days of winter, it strikes me as ironic that on the final day of support for client/server Forms 6i, I am writing about the future direction for Oracle Forms. Given I'll be addressing the future of Oracle Forms many of you might expect this to be a very short article. Client/server desupport essentially sounded the death knell for Forms didn't it? The world is all about Java now...or .Net? If you've got a Forms application you need to get rid of it now, right?

Well, maybe, maybe not. But whether you think this will be a short article or not, I've got about 1300 words to play with. So pull up a chair, park your preconceptions to one side and let's hear what Oracle has to say about the roadmap for Oracle Forms.

The History of Oracle Forms: Volume 1

If you have made it this far, and given you are reading this article in the ODTUG *Technical Journal*, there is a good chance you've been around Oracle Forms for quite a while, so lets take a nostalgic look back at Forms.

Oracle Forms started way back with release 5 of the Oracle database (although it existed in embryonic form in earlier versions) and allowed you to build data input applications for the cutting edge computing resources of the day: green-screen mainframes. But the landscape changed, Windows came along and all of a sudden users had a mouse, graphics, and a color other than green; and so Forms evolved. With a bit of hacking and some perseverance, you could move your Forms applications off of your mainframe and exploit the processing power of the desktop.

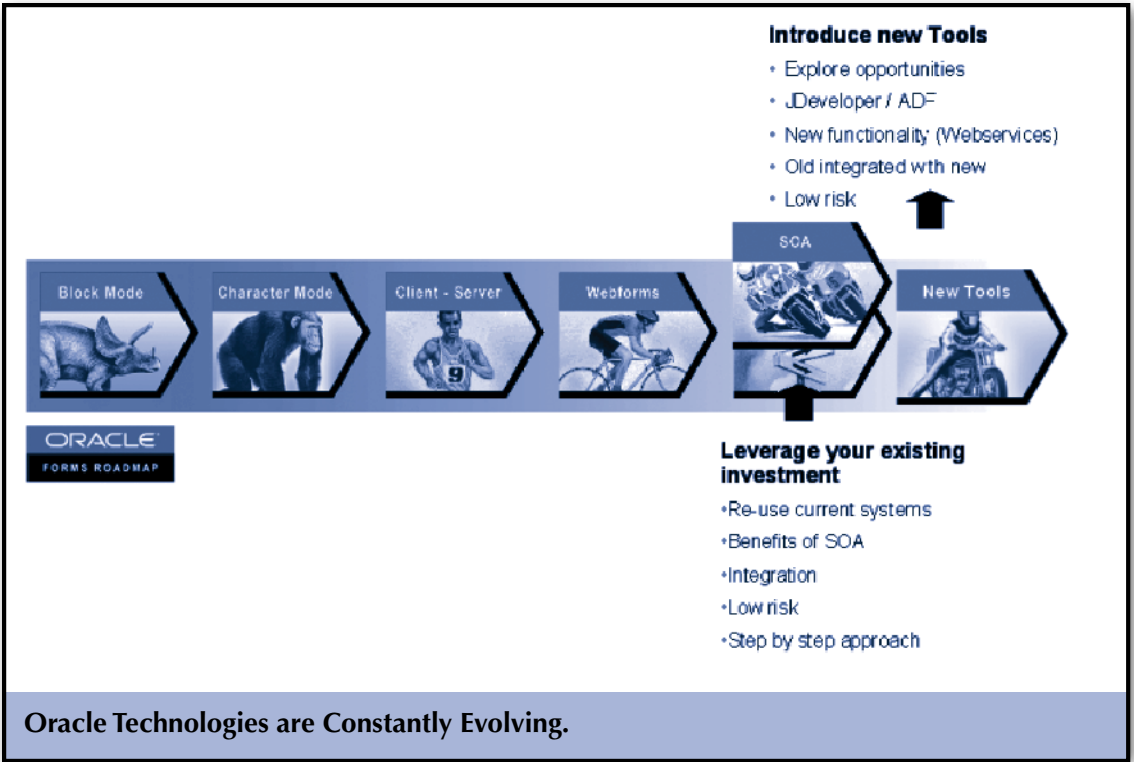
But as Benjamin Disraeli once said: "Change is inevitable. Change is constant," and the Internet certainly changed things. Oracle Forms evolved, and for the

past ten years Oracle Forms has allowed to you build and run professional data entry applications across networks.

Ok, so now is it time for Forms to rest on its well-earned laurels? Well actually, no. Today there are new challenges: Standards, Java, and SOA (Service Oriented Architecture) to name only a few. While you might want to take your first steps into these new worlds, what you don't want to do is destabilize the years of investment in your existing applications. Hence, what we see now is not so much an evolution (and hence change the way these applications work), instead more of an adaption. Forms is adapting to the SOA world. Without changing the fundamental way a Forms application functions, it is allowing integration with a whole new world of services and technologies.

But here is the big question you should be asking. "Why should I be changing anything?" After all, these applications are underpinning the workings of your banks, governments, universities, transport, and commercial systems.

Well, in the same way that technology has changed, it's also true to say that the requirements for business have changed. Oracle Forms was, and still remains, a formidable tool for building your backoffice data entry applications. But those were the days when your backoffice staff was based in your HQ or call center. Geographic boundaries fade as businesses globalize and merge and you need to centralize your applications while allowing ease of access to these applications. And Oracle Forms meets that challenge head on,



allowing your Oracle Forms based professional applications to be accessed via a browser across a network. That's the Forms "sweet spot".

But other changes are less well addressed by Oracle Forms. Sure, you can still call up your bank and get a statement of your balance but the Internet has changed that. "Self-service application" is the ubiquitous name given to more ad-hoc casual-use application. Of course, in theory you can use Oracle Forms but there are a whole range of technologies that address that need without the expense of downloading an applet and installing a plug-in. And what about being able to run on wireless or handheld devices?

There has also been a change in thinking as well. In the past, IT served the business by building monolithic applications for managing your data. A typical Forms application could be hundreds of forms and libraries implementing many business processes. But the business is now recognizing the benefits of moving to a service approach. You don't just build silo applications, instead you build services that map onto your business processes and hook these services into your applications. So, "Create Standing Order" it not owned by a single application but instead is consumable by any application that requires an implementation of that business process. This gives a better alignment between business and IT as well as ease of integration, agility, and of course, reuse.

A Meeting of Two Worlds

And this is where the legacy and the new worlds meet. With years of investment in your Forms applications it's a brave executive who makes the decision to rewrite or pull the plug on your core business system just because the technology is not "cool enough". You need a strategy that will allow you to:

- Protect the investment you already have
- Extend those applications as the business dictates
- Evolve your applications and skills

It's almost a case of technology imitating life. We grow, evolve, and adapt to new challenges all the time but still try to retain a level of stability. And the roadmap for Oracle Forms firmly reflects that approach:

- A commitment to the future of Oracle Forms
- The ability to extend applications through services and new technologies
- Evolve to meet your strategic goals

And this is an approach that is being reflected in the customer base as well. TietoEnator's Mortgage and Savings Suite is the United Kingdom's most widely used mortgage software, with 25% of mortgage accounts managed using the system. Paul Scott, System Architect at TietoEnator, comments:

"We want to evolve and modernize our applications but we have to remember that these applications are core to our

business. Our goal of extending and evolving the application must be balanced by the aim of also protecting, not only our own investment, but also that of our customers. Oracle's roadmap gives us the potential to adopt a low risk approach to achieving that."

The step-by-step approach to modernization is also reflected at international IT services company Atos Origin. Malcolm Smith, Portfolio Manager at Atos Origin: *"By using SOA, companies can evolve their applications at their own speed: Think Big, Start Small. This gives you a step-by-step approach: adopting SOA, discovering new technologies and tools and finally reuse of existing Forms logic. These are the three most important components of what is known within Atos Origin as the Forms Roadmap."*

And ODTUG Editor's Choice Award winner and system architect at Eurotransplant, Wilfred van der Deijl, is already on the road to SOA: *"We want to deploy an evolutionary model, not a revolutionary one. In the past, adopting a new technology frequently meant we had to abandon the old technology. With SOA and Oracle BPEL Process Manager, we are proving that we can service-enable our existing systems and reuse them in the new architecture."*

The Technical Details

So, what is in your armory to help you, as a developer, integrate your new services and your legacy Forms applications? Well this is where you might take your first steps into the Java world with Oracle JDeveloper, which allows you create a simple Java class for calling a Web service. That class can then be "wrapped" in PL/SQL by the Java Importer in Oracle Forms. Voila! Forms calling a service. And you don't have to stop there. A BPEL process, an orchestration of multiple services is exposed to Forms in exactly the same way.

But let's look at some more complex use cases: what if you have a long running service (for example, a credit check) that requires the service to asynchronously return a result to Oracle Forms? Or you need to integrate your Forms application with an ADF Faces application? In our next release, Oracle Forms 11g, we are planning two major new features:

- External events – an enhanced event model that will allow Oracle Forms to react to asynchronous service calls.
- JavaScript integration – allowing Forms to call client-side JavaScript and for that JavaScript to call back into Oracle Forms!

Your Way Forward

So, while now is a time when many of you are considering modernizing your IT applications, probably very few of you will be afforded the luxury of green field development. Your existing application investment has to be factored in. For some, changes in business may mean rip and replace is an option. But for most, the cost and risk associated to the business is too prohibitive. In which case, you need a modernization option that allows you to mix these applications

with your new development in a way that allows simplified integration, reuse of your skills and investment while taking advantage of new technologies and opportunities. This is the Forms roadmap. ☼☼

FIND OUT MORE

Oracle Forms and SOA Executive Summary:

http://www.oracle.com/technology/products/forms/pdf/10gR2/Forms_SOA_Summary.pdf

Oracle Forms and SOA: The Hows and Whys

http://www.oracle.com/technology/products/forms/pdf/10gr2/formssoa_whys.pdf

Oracle Forms Modernization events:

http://www.oracle.com/technology/products/forms/forms_events.html

Tools statement of direction: <http://www.oracle.com/technology/products/forms/pdf/10gToolsSOD.pdf>

Oracle Forms calling a Web service:

http://www.oracle.com/technology/products/forms/htdocs/10gr2/howto/webservicefromforms/ws_10_1_3_from_forms.html

Oracle Forms Modernization: Client/Server to SOA

This site provides a roadmap to modernize applications based on Oracle Forms technology.

http://www.oracle.com/technology/products/forms/forms_modernization.html



About The Author

Grant Ronald is a group product manager working for the application development tools group responsible for Forms and JDeveloper where he has a focus on opening up the Java platform to Oracle's Forms 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 Centers 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.



Technical Calendar

JUNE

June 10-11

New York Oracle User Group Meeting
New York, New York

www.nyoug.org

June 12

Georgia Oracle User Group Meeting
Atlanta, Georgia

www.gouser.com

June 14

ODTUG Brigade: A Day of Community Service
New Orleans, Louisiana

www.odtugkaleidoscope.com

June 15-19

ODTUG Kaleidoscope 2008
New Orleans, Louisiana

www.odtugkaleidoscope.com

JULY

July 10

Georgia Oracle User Group Meeting
Atlanta, Georgia

www.gouser.com

July 14-18

Puget Sound Oracle User Group Oracle Days
Bellevue, Washington

www.psoug.org

For Important ODTUG News & Updates Visit

WWW.ODTUG.COM