

—Duncan Mills
Oracle Corporation

At any conference for Oracle developers I can guarantee that the question will be asked: “So what tool should I be using?” In the dim and distant past, things were a lot simpler. You could build applications the easy way with a declarative environment like Forms, or the hard way with languages like C and Cobol. Today, however, Oracle offers several tools that are usable by developers of all levels and it’s that choice that seems to foster the confusion. Things are further muddled with the Open Standards vs. Microsoft .Net vs. Scripting Languages debate. That’s something for another article and although there are merits in some of those approaches they are not strategically important to Oracle in the way that the tools I explore here are.

Let’s start off by looking at the candidates, although you’ll forgive me if I exclude the precompilers and Applications-only solutions such as PeopleTools or third-party IDEs from the considerations:

- The Java platform, more specifically JDeveloper, Oracle ADF (Application Development Framework), Oracle WebCenter, and Fusion Middleware, represents the bulk of our development focus at Oracle right now.
- Oracle Application Express, the new kid on the block.
- Oracle Forms, still going strong, and representing a huge pool of developer talent.

So which tool should you choose? Well what I won’t do is make your decision for you, all I can do is furnish you with the basic facts sprinkled with a little opinion based on what we use the tools for ourselves. The main problem in this discussion resolves down to the old adage, “Give a man a hammer and everything looks like a nail.” We can use any of these technologies to build a basic master-detail application. In fact, experts in each of the tools would probably complete such a task in about the same time. This should not really be a surprise; database applications are our bread and butter. Any tool that Oracle produces will address the basic use case of building CRUD (Create, Retrieve, Update, and Delete) screens as a matter of course. To explore the merits of the different tools we need to look beyond these trivial tasks and try to establish the specialty or “sweet spot” for each of the tools I’ve mentioned.

Let’s start with Java, ADF, WebCenter and the Fusion Middleware suite. Here we are dealing with the enterprise development space, and the sheer number of individual

The Right Tool For The Right Job?

technologies involved is daunting. Fortunately it’s not an all or nothing proposition and developers will rarely have to be experts in the full range of possible technologies. The JDeveloper IDE aims to cross the declarative development experience and productivity of the PL/SQL-based development tools with the power and flexibility of Java and Object Oriented development. Based on open standards, the ADF Framework allows developers to build complex applications in the way that best suits their development style, be it highly visual with modeling and wizards or low-level code editing.

In contrast, Oracle Application Express is firmly aimed at the Oracle community, with SQL and PL/SQL at its core and a very simple browser-based design and deployment model. Oracle Application Express excels in rapid template-based development of departmental or opportunistic point solutions. Agility is the key attribute here. For many organizations Oracle Application Express is an effective Microsoft Access replacement, solving the same problems but without the inherent data control, security, and performance issues of Access.

Finally we have Oracle Forms. For such a long time, Oracle Forms ruled the roost with a virtual monopoly in the Oracle developer space. Oracle Forms also uses PL/SQL as its core programming language. This familiar language for database users, coupled with the highly declarative nature of the Forms development environment, has ensured the continuing success of the tool within the Oracle community.

So let’s consider how these tools play out as options. The key thing to understand is that it’s never a mutually exclusive situation. Within Oracle we use all of these tools in our day-to-day work, and such heterogeneous environments are common throughout the Oracle customer community as well. At the basic level, the key strengths of each of the tools lend focus to a particular segment of an organization’s application-building needs.

Oracle Forms excels as an engine for back-office functionality and data entry, supporting functions such as type-ahead and instant validation. The ability to integrate with the end user desktop via OLE interfaces to Microsoft Office and facilities such as file upload and download to the client desktop provides the kind of integrated operating environment that is required for most enterprise office systems in one form or another.

Oracle Application Express lends itself to the rapid development and deployment of opportunistic applications typically used within departments. We also see it being used as an ad-hoc office productivity tool replacing spreadsheets or providing one-off reports. Within my team for example, we use Oracle Application Express to replace spreadsheets for tasks such as tracking abstract submissions for events. Oracle Application Express also comes with a whole set of surprisingly useful mini apps predefined. It is a good analogy to compare Oracle Application Express to the Dashboard feature on Mac or the Gadgets feature in Windows Vista. It's there, it's quick, and there are whole bunches of useful prebuilt widgets.

JDeveloper as an IDE for ADF, WebCenter, and the Fusion Middleware Suite is altogether more complex and the emphasis here is in branching out beyond the basic job of putting a UI in front of a set of database tables. Modern enterprise development is all about working in a heterogeneous environment glued together by sets of open standards. A business can be viewed as a series of services that can be reused or orchestrated in various ways to build complex applications, cutting across traditional application boundaries. Much of the focus within such applications pivots on the plumbing issues of assembling and orchestrating services and building user interfaces or data exchange APIs without real awareness of any particular database involvement. Of course the database is, more often than not, intimately involved, but that fact is not a constraint for the application developer. The focus has shifted somewhat from the view of the database as the single integration hub, to a situation where this role has shifted to the orchestration layer provided by the application server; hence the name Middleware. ADF brings with it several key technologies that are increasingly important to business applications. Particularly modern UI frameworks such as JavaServer Faces provide the capability to build highly interactive self service UIs using Ajax technologies, but without the programming complexity usually implied by such richness.

Which Way to Lean?

As I implied before, the lines between these tools are very blurred and we can stretch and pound them all to fill pretty much any application niche. The trick is to evaluate all of the factors that will help you to make an informed business decision.

THE WHY, WHAT, WHO AND WHERE OF CHOOSING A TOOL

Rather glibly we can sum up the important factors in terms of these simple questions:

Why?

First and foremost what is the exercise all about? Why are you looking at making a decision about which tool to

use? It certainly makes sense to be aware of your options, but one of the biggest mistakes a development organization can make is to jump into a new technology for the sake of fashion. If a system needs to be replaced or enhanced (or indeed something new is required) for good operational reasons then that's a reason to examine your options. If your applications are working and on a supported platform there is very little to be gained from rewriting just for the sake of it.

On the other hand, a move onto an open standards-based platform using ADF and Java Enterprise Edition may prove to be a very attractive proposition if you are building an application for resale. These technologies are, by definition, able to run on any Java Enterprise Edition-based application server and the ability to support a wider range of vendors and platforms is a very good reason to move into the Java space.

What?

What are you trying to build? As we've discussed, the tools have particular core competencies, and what you've read already may be enough to clearly identify the right choice. However, we have to accept that many applications are large and multifaceted. A single development tool may not be enough. For example, you might build internal dashboard screens using Oracle Application Express but need something with WebCenter customization capabilities for parts of the public interface. The one sound bit of advice on adopting any new tool, has to be of course choosing your battles. Be prepared for blood, sweat, and toil in your first encounter with any new technology. As such, select a project that won't cripple your business if it's not delivered on time. Be realistic and keep to a small scale in the first instance.

Who?

Tools don't build applications, people do. Who do you already employ and what are their skills? If you branch into a new technology do you want to go it alone, or perhaps hire some experts to mentor you through those first few projects? Your existing skill set is often the most significant factor in being successful with a toolset. If you have a room full of Forms programmers then accept that they're going to crank out Forms applications more efficiently than anything else. One thing is certain, developers who understand the full range of available technologies, Forms, PL/SQL, SOA and Java are a rare breed and a valuable asset.

Where?

Where is a highly significant factor. What's the deployment target for the application you want to build? Will it be running in-house in a controlled environment? Or does it get sold outside into an environment where you can't rely on an Oracle database or application server being used. In this latter case, ADF becomes the perfect choice allowing for the use of any database or application server.

Continued

ORACLE

Tools Direction

(continued)

Another factor for resold applications in particular is the requirement for flexibility. Do you need to support both skinning and customization for a particular customer, what happens when you release a patch, can you even support customization in the first place? This is a key focus for ADF and the whole Fusion Architecture.

Oracle Application Express also has a great strength here, although reliant on the Oracle database, that's a pretty pervasive platform, and the sheer ease of installation and deployment in that case makes it very attractive.

What to Do Next?

I've talked about the options and I firmly believe that your best choice is to use one or perhaps all of these technologies in the right places. I also believe that, although there are many choices outside of the Oracle stable for tools, these are not a productive way forward for most customers. Each of the Oracle tools has all of the basics such as CRUD covered; having taken huge amounts of investment to engender productivity and fitness for their chosen task, a brief foray into alternative tools with any real application

The Right Tool For The Right Job?

will quickly make you realize exactly how much you get today for free with any of the Oracle tooling options.

My final advice is to be informed and pragmatic. It is a mistake to base any kind of strategy on information that you can't entirely trust, so I ask that you don't entirely trust me. Try out each of the tools for yourself; one size most definitely does not fit all.

About the Author

Duncan Mills is Director of Product Management for Oracle's Application Development Tools including Oracle JDeveloper, Forms, and the ADF Framework. Duncan is currently responsible for product direction, evangelism, and courseware development around the development tools products. He has been working with Oracle in a variety of application development and DBA roles since 1988. For the past 14 years he has been working at Oracle in both Support and product development, spending the last 8 years in product management for the development tools platform. Duncan is the coauthor of Oracle JDeveloper 10g for Forms and PL/SQL Developers—A Guide to Web Development with Oracle ADF.