

Deploying PLSQL applications, building Rome in a day

Alan Arentsen
Ordina
The Netherlands

Keywords:

PL/SQL, APEX, Deployment, Installer, Upgrade

Introduction

All roads lead to Rome but some roads just are more comfortable than others. I found a really nice road to ride when it comes to deploying PL/SQL and APEX applications.

Windows users use an MSI-file to install their applications, Apple users use their own packages. But what do PL/SQL users do? Usually they export stuff from the database and fabricate some kind of installer procedure.

How nice would it be if you just click 'install' and your application will be deployed? Or is this too good to be true? I don't think so!

If you follow the right procedures during the development process, this installation process could be the smoothest way. I wrote an installation set which easily installs my application. From now on it is not more than a push on a button and the application will be installed as it should.

In this article I will explain why I did this, how I did this and what opportunities there are for expanding this solution with. I'll discuss the installation of new applications as well as running an upgrade on an existing one.

Inspirations

If you read the Oracle documentation of 'Deploying a Database Application' online, you will have a good overview on how to deploy your application and what things are important when doing so.

But, there may be some more things for you to do. The text is full of lines like this:

- 'Exporting the Database Objects'
- 'Generating a Script for Creating...'
- 'Special Considerations...'

And beside that there is an eight step plan to run your installation scripts.

It is a lot of work to do. And you have to do it every time you deliver your software. So I think if you want a smooth software delivery you will have to start working with a structured environment when you are developing.

Developing

There are as many ways to developed as there are roads to Rome. But what way should you take and which exit is yours?

A few major decisions you should make are about working file based or from the database, using a revision control system or not and do or do not automate your installation process.

It's a big deal. Some people just work in de database, because they always did and because they think is it better and easier. Well maybe it is in the projects they work. Maybe they work alone on a project. It is not a bad habit to do so. But there are some disadvantages in working in the database especially when you work on an application with a few colleagues.

At first when you work with multiple people on the same project, there comes a moment when you edit the same package and one of you lost its source code. The easiest way to work around that is by communicate with each other about who is editing this package.

The second is that when you did a change last week and after testing you decide the change was not the right thing to do after all. You cannot rollback or flashback to your change. And if you could, which one is the one you need?

That's where revision control comes in. If you put your files in revision control you will always be able to compare your checked in file with each other. So it's easy to find the right version and rollback if you want.

Another good thing about revision control is, you can always see which user made which change.

But when you are under revision control, there are two places the database objects are stored. In de database and in the revision control system. And where there two versions, one of them should be the owner of the objects.

If we should pick the database as the owner of the object, you are responsible for updating subversion.

And another problem is: what should you do with temporarily objects you use just for testing your software. If the database is the owner, these objects belong to the application and also should be deployed to the testing and acceptance environment.

On the other hand, if the revision control system is the owner of the data then you can do what you want in the database, the objects are in the revision control system. And you always we be able to reinstall your database scheme from the revision control system.

And now the last one looks clear. You should automate your installation process. But how far will you go. For smaller projects with less needs it will be sufficient to just use the standard installer you use for all you projects. If you are in a bigger project with different kind of acceptance testers you might want to go full throttle and us a build server like Hudson or Jenkins to automatically deploy your build every night. The best way is to start small and expand your process by your needs.

Contact address:

Alan Arentsen

Ordina

Ringwade 1

3438 MN, Nieuwegein

Phone: +31 (0) 6 13 01 88 32

Email alan.arentsen@ordina.nl

Internet: alanarentsen.blogspot.nl