

# MARRYING OSCM AND JAVA DEVELOPMENT: THE PRENUPTIAL AGREEMENT

*Andrew Faulkner, Datagruven Systems Inc.*

## Abstract

Oracle Software Configuration Management (OSCM) is a powerful offering of the developer suite with its origins in the management of Forms and Reports. Many development shops exist with a legacy of this work and some have taken the leap and irrevocably flipped the switch on their Designer Repositories and are now committed to SCM. There are a number of considerations as we start to manage all those artifacts that our Java development efforts are starting to spawn.

The technical and workplace issues around using the OSCM tools on Java projects are highlighted with best practices suggestions offered. The topics addressed include: JDeveloper integration, XMI transfer between IDEs, granularity, impact analysis, and integration with the heritage forms applications.

## Introduction

Long standing Oracle Forms/Designer shops are now facing the very real need to incorporate the efforts of their Java development initiatives within some enterprise Software Control Management process. Oracle offers their product, Oracle Software Control Management (OSCM), within the Oracle Repository that has been quite successful in supporting the system development lifecycle (SDLC) for the Designer/Forms approach.

The Oracle Java development environment JDeveloper integrates with the Oracle Repository and provides attractive functionality for integrating with OSCM. Before embarking on this course of action a number of fundamental decisions have to be made. The joining of OSCM to Java requires an up front understanding much like the prenuptial agreement that services an impending marriage.

The approaches discussed in this paper are from the perspective of large government institutions that require custom software solutions developed in order to achieve their program objectives.

## The Agreement

First a quick primer on prenuptial agreements (prenup). Claiming no expertise in this field, and having never had occasion to create my own, I differ to the eminent source of the *Pennsylvania Weddings* magazine. (Albert-Heise 1998). There are three basic purposes to a prenuptial agreement:

1. Identify and distinguish between pre-marital property and marital property.
2. Outline the expectations of each spouse with regards to pre-marital and marital assets and debts in the event of dissolution of the marriage or in the event of death of a spouse.
3. The obligation of each spouse to support the other in the event of dissolution of the marriage.

For our purposes these can be rephrased as:

1. Identify and distinguish what is to be maintained in OSCM from here on forward and what remains outside the sphere of OSCM.
2. The development process is a life cycle of inception, analysis, design, development, maintenance and obsolescence. There are expectations of performance by the Java development teams and the OSCM Repository management (project management) services that need to be clearly understood.
3. The project will not be in the intense project development stage forever. What needs to be in place in the long term to justify the effort and commitment required to pursue rigorous SCM protocols during the SDLC.

## **SCM Objectives**

As work progresses on a project, resources are spent and various artifacts are generated. These artifacts (entities, tables, diagrams, procedural code, analysis documents, etc.) are the intellectual property that result from the expenditure of money and people's talent and it is prudent to safeguard them.

Software Control Management's purpose is to maintain the integrity of these artifacts through the entire project lifecycle. SCM classically addresses four common kinds of problems:

1. communication problems which appear when the number of communication paths increase within the project team,
2. problems caused when a developer is not aware of the latest changes to software configuration items shared among project members,
3. problems related to the maintenance of multiple versions of a given item,
4. and problems with simultaneous updates of shared components. (Leon 2000)

Oracle SCM does satisfy these four cornerstone requirements with a granular management of repository objects and a complete implementation of versioning, branching, and release configuration.

In this paper we are primarily concerned with SCM adding value to the enterprise in two ways: the safeguarding of assets and promoting productivity/quality enhancements. These objectives are from the perspective of a large government institution.

## **Safeguard Assets**

Assets need to be securely managed for any number of reasons but paramount among them is to retain the intellectual property in the control of the enterprise so that if, for whatever reason, the developer working on it becomes unavailable they are not lost.

Secondly, the artifacts produced will typically go through iterations of development and retaining all versions of these allows the developer to step back to a previous, more successful iteration when a particular development direction being explored is found to be ineffective or they just make a bad mistake and need to recover.

Government institutions typically lack a stable of in-house developers to produce and maintain applications. They are instead in the business of contracting out the development to third parties who create the application to their requirements. Interim and terminal delivery of project artifacts provides a level of assurance that some useful work is being done on the project. In addition taking delivery of these artifacts into the institution's repository provides for some control should the relationship with the developers be severed.

## **Productivity/Quality Enhancement**

The repository of project artifacts is a tremendous wealth of metadata that describes the institution's business, development projects, and deployed applications. By having all projects in the OSCM repository both the heritage Designer/Forms applications and the Java projects, there are a number of productivity and quality enhancements that can be realized.

Large institutions are interested in the development of Enterprise models that assure common use of the best available models. These models change over time. With all applications in the Repository it is possible to conduct thorough impact analysis when changes to standards and implementations are proposed.

## **Enterprise Environment**

The enterprise environment within which the implementation takes place will have a great effect on the depth and quality of the SCM effort. To be considered are:

1. The nature of the workplace.
2. Tools environment.
3. System Development Life Cycle (SDLC).

## **Workplace**

The nature of the relationship between the application developers and the client institution will control the ease with which the developer's efforts can be integrated into the corporate repository.

### **DEVELOPERS**

The developers themselves may work externally to the institution (down the street, near-shore, off-shore) and have physical limitations to accessing the Enterprise repository. Secure networks need to be established to OSCM Repository. The requirements have to be defined for the developer describing:

- integration with the Enterprise models;
- frequency of artifact delivery;
- notification process;
- level of repository management authority;
- promotion process from dev to test to prod;
- granularity of artifact delivery (bundled package like a zip or EAR file, or individual class files, etc.)

The developers may be in-house personnel either as employees or as embedded developers working in-house. As such they can expect real time access to the OSCM repository without impediment.

### **QUALITY ASSURANCE STAFF**

The workplace will contain staff who function in the quality assurance (QA) roles. The QA staff take delivery of project artifacts as they are promoted from the Development to Acceptance Testing and onto Production. The artifacts that are bundled into a Configuration provide a discrete unit of management for promotion between environments. With the metadata of the artifacts in the repository it is possible to efficiently test the quality of the deliverable. With all products under version control there is no confusion about which DDL or document or Diagram is to be evaluated. Other areas of QA interest are:

- Conformity with Enterprise modeling standards;

- Domain usages;
- Naming conventions;
- Completeness of comments and descriptions;

## Tools Environment

Oracle's own Java development environment is JDeveloper which connects with the OSCM and synchronizes the files. The functionality for OSCM is being improved with every software release. There are many concerns about the usability of OSCM integrated with the current release of JDeveloper .

Other tools environments such as Rational Rose, TogetherSoft or JBuilder may be in use by the developers. This presents a problem since there is no easy way to version control these foreign artifacts in OSCM and reap the productivity and quality benefits desired. One avenue of approach is the use of XMI files (XML interchange files) for importing these other models into JDeveloper and then using the JDeveloper integration to access the OSCM.

## SDLC

The enterprise should have a System Delivery Life Cycle (SDLC) that will outline the delivery requirements for the project. Effort should be taken to incorporate the SDLC into the SCM process. Structuring the Folders for each project container so that there is a clear and logical place for project artifacts will be helpful to the developers. Using JDeveloper to connect to the OSCM will store the files in a Folder structure defined by the product.

Questions that need to be addressed in your SDLS related to the SCM are:

- Who does the delivery?
- What level of granularity do you store the artifacts in?
- How often to mandate updates?
- Who approves configurations for promotion?

## Java SCM Problem

For Designer/Forms development shops that matured through the 1980s and 1990s the move to Java can be quite jarring. It seems we spent decades refining and improving our tools and processes so that we could achieve Repository based GUI driven CASE development with 100% generation. Remember the slogan, "Pictures to Code and back"? I have it on an Oracle Designer T-shirt. Now the colleges are churning out Java coders who are talented in writing code for web-based applications. But lost along the way is our Repository based GUI driven CASE development that generates the code. People are hand coding like never before.

The tools environment the developers work within (JDeveloper, JBuilder, TogetherSoft, Rational Rose) generates hundreds and hundreds of small files, as compared to the dozens and dozens we are used to in Forms development. Understanding and controlling all these artifacts as they get incorporated in the Enterprise repository is daunting. Remember earlier we stated our objectives for Enterprise repository SCM initiative was 1. Safeguard Assets and 2. Productivity/Quality Enhancement.

The first objective seems attainable in some form by taking delivery of the artifact bundle, even if it is in one honking .ZIP file and versioning it at this gross level into the repository. Meeting objective 2., however, requires a granularity of representation that allows various tools and parsers to work on the metadata so as to achieve quality enhancements such as cross enterprise impact analysis for prospective changes to an application.

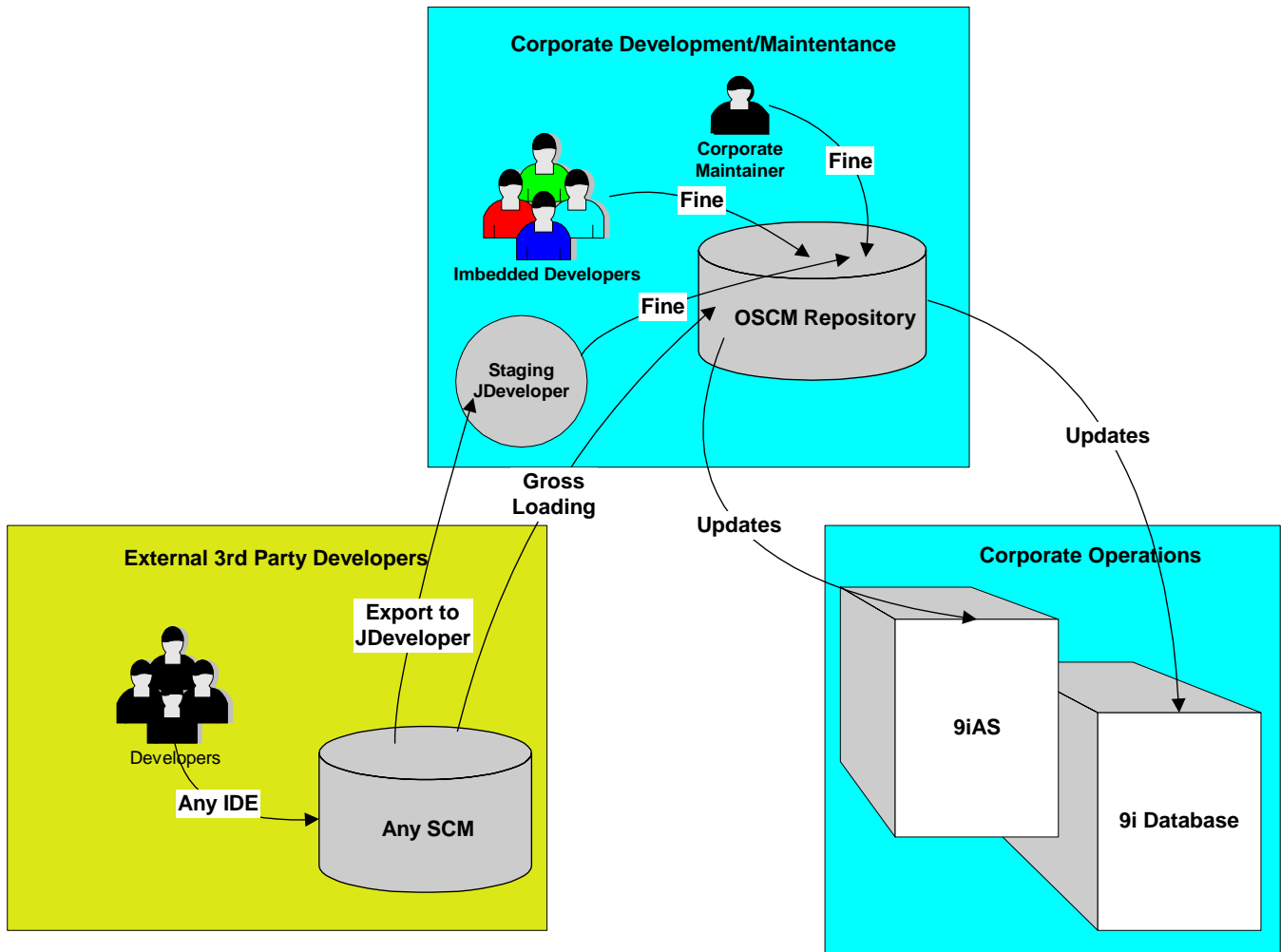


Figure 1 Metadata Movement

## State of the Art

There are two main approaches to Java SCM using Oracle SCM. The first approach is to bundle up entire deliveries into a .ZIP file and hand that off for versioning in the repository. I call this the *gross* approach. The second approach involves some strategy of file level synchronization, either through JDeveloper or by using the file synchronization capabilities of OSCM. I call this second approach the *fine* approach.

Both these approaches are depicted in Figure 1. above.

## Gross Approach

The bundled files can be Java EAR files, .TAR files or even .ZIP files. The development group delivers the bundle and it is securely loaded into the project container and versioned as a whole. Loading the target operational systems, middle tier server and database, require staged deployment by expanding the contents into the file directories.

What is lost in this approach is the connectedness to the other metadata in the repository. Parsers cannot trace the dependencies between objects. Reports cannot look for all occurrences of a “Party” entity.

## Fine Approach

The more granular approach exploits the detailed representation of the project objects in the Repository metadata. By synchronizing the objects using a File based mapping or the JDeveloper integration the Productivity/QA objectives can be met. The ability to use the parsers to investigate dependencies is enabled and Reports on the metadata can be complete.

## Work in progress

Version controlling the Java projects using OSCM is not yet a thing of beauty. There is room for improvement in a number of areas. Here’s a sample of what needs to be addressed:

- Performance can be a problem with long wait times in check-in and check-out.
- Documentation on right-sizing the tool to different project complexities is needed.
- Lack of Repository reports of interest to Java Developers.

## Summary

The Oracle Designer/SCM product is on a quarterly release schedule and is very much a work in progress. In a hybrid environment in a large institutional setting with both heritage Forms applications and the newer Java applications using the OSCM product makes some sense. It is one place to put everything. Some of the performance issues and the feature limitations have brought into question whether the pain is really worth it. The Java community has bred it’s own SCM favorites and processes for quality assurance and product configuration.

When putting OSCM together with Java development think prenuptial agreement:

- The two are getting together for a greater good.
- They both bring baggage to the relationship so clear delineation of responsibility and expectation is required.
- One doesn’t always understand the needs of the other but there is overall mutual agenda, so communication is key.

## Resources

The very best coverage of Oracle SCM issues are on the Oracle Technet site <http://otn.oracle.com> Both the Designer and the SCM forums are relevant. There are a number of serious practitioners who are always commenting, suggesting and answering the tough questions. The responses from the Oracle staff who lurk there are quick and complete. David Brown seems to post something almost every day. Even if some particular feature is agonizing or impossible at the moment they do give hints as to what is just around the corner in the next release.

## References

Albert-Heise, Deborah (1998 February). Yours, Mine and Ours: A Guide to Prenuptial Agreements for the Bride and Groom, Pennsylvania Weddings Magazine, (See also <http://www.paweddings.com/PRENUP.HTM>)

Leon, Alexis (2000). A Guide To Software Configuration Management, Artech House Computing Library, Norwood, MA

## About the Author

For 20 years Andrew Faulkner has worked as a software developer, systems analyst, DBA, data warehouse architect, Senior Data Manager, and now as a consultant based in Victoria BC. An Oracle Certified Professional with a B.A. (Psychology), an advanced diploma in Software Development and B.Tech. (Computer Systems) he has served on the executive of the Victoria Oracle Users Group for the past four years. He has presented a number of papers, in recent years, at both OOW and ODTUG on the topics of BI tools, Metadata management, and more recently work in knowledge management. He can be reached at: [afaulkner@datagruven.com](mailto:afaulkner@datagruven.com)