

# Promoting Project Integrity with Reliable Version Management

*An Oracle Data Integrator Technical Brief  
Updated December 2006*

# Promoting Project Integrity with Reliable Version Management: An Oracle Data Integrator Technical Brief

**An accurate, reliable system for version management is critical to the success of any integration project.**

## **THE CHALLENGE OF VERSION MANAGEMENT**

One of the most difficult challenges development teams face is managing a project's hundreds—sometimes thousands—of work units throughout the development process and beyond. Success or failure of the version management (or version control) process can greatly affect the integrity and success of any development project.

Oracle Data Integrator addresses the challenge of version management by providing the features necessary to easily safeguard, replicate, restore, and secure work units consistently and smoothly, in even the largest development environments. This technical brief describes the challenges of maintaining accurate version control during the development lifecycle and how development teams can use Oracle Data Integrator to ensure the integrity of their integration projects.

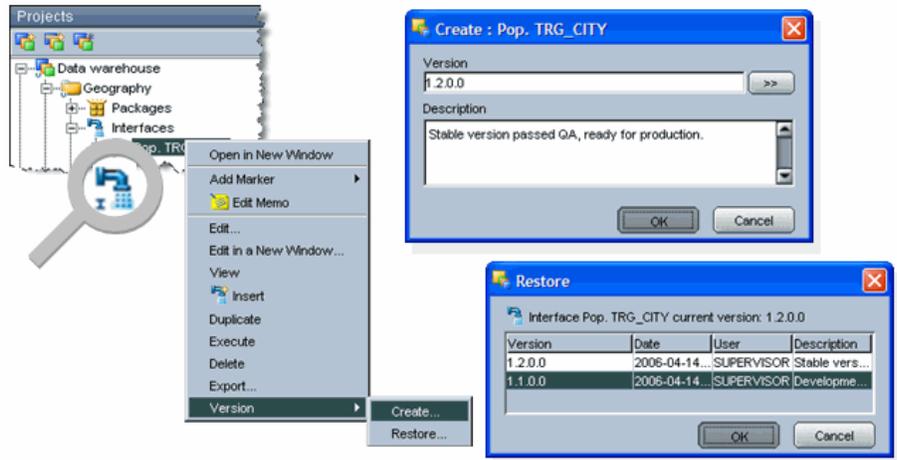
## **ABOUT VERSION MANAGEMENT**

Large-scale integration projects are increasingly being developed by numerous people in different locations who frequently change, test, compile, replicate, and restore the project's work units during the product lifecycle. Ensuring that the version of each work unit is the most current and its integrity is maintained during handling is critical to producing a superior product on time.

For example, developers must be certain that when they perform a backup of an integration project's source code before final production, that code will be in the same state when it is later restored for further development or maintenance.

### **Safeguarding Work**

Oracle Data Integrator provides the ability to safeguard and then restore versions of work units such as projects, models, interface, packages, and so on. Oracle Data Integrator does this by creating a version of an interface and then restoring it later, as shown in Figure 1 below.



**Figure 1: Oracle Data Integrator creates and then restores a version of an interface.**

With Oracle Data Integrator, each version—not each work unit—is identified by a version number and a comment (as shown in the Create window in Figure 1). Work units are identified by their icons in the tree views (as shown in the Projects window). New work units and those that have been updated from the last iteration are clearly identified as such. For example, the **I** (shown in the magnifying glass) in the Projects window indicates a new work unit.

### Replicating Work

A serious problem can occur during the development process if work units are stored in more than one place. In a development environment with multiple repositories, it is sometimes necessary to replicate integration processes between repositories. This structure carries the risk of overwriting the most recent version with an older one.

Oracle Data Integrator eliminates this risk by storing all versions in the Master Repository. Developers can restore copies of the versions from the Master Repository into any Work Repository attached to that Master Repository. This structure allows developers to safely and securely replicate work units between repositories while maintaining the current versions in one central location. In this way, moving objects from the Development Repository to the Test Repository to the Production Repository is easy and safe.

## MANAGING DEPENDENCIES

### The Dependencies Challenge

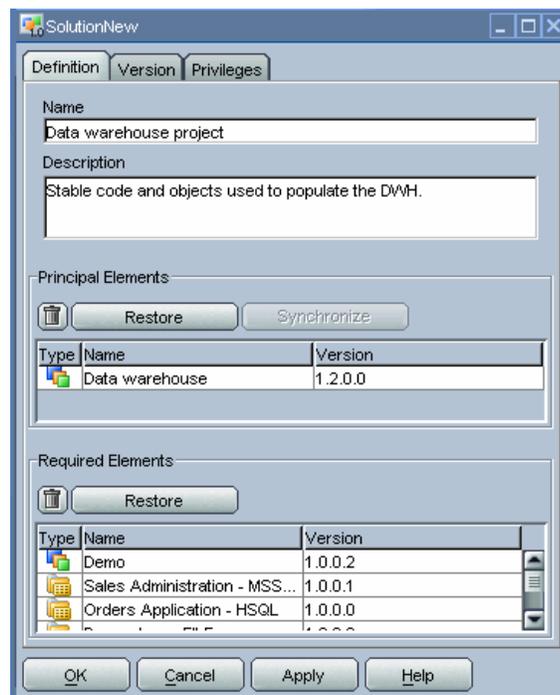
A work unit in Oracle Data Integrator rarely exists in isolation. Work units most often depend on other elements. For example, interfaces rely on the data models that describe the data that’s being integrated. Restoring an earlier version of an interface does not guarantee that the version will work if the data models were changed after the earlier version was saved. In the past, a simple interface–data

model relationship could be managed by manually archiving the model. For today's projects—which involve dozens of models and projects and hundreds of interfaces with numerous and complex references—this manual archiving process is no longer viable.

## Working with Solutions

Oracle Data Integrator's extensive Metadata Repository contains all the necessary cross-references between objects. Cross-references are automatically parsed to form groups of interdependent objects, which then become solutions. For example, before you perform a project release, you create a version of the project. To ensure that all the work units involved in the project are also versioned consistently, you create a solution and enter the project in the Solution window (as shown in Figure 2 below). The Designer module then calculates all the dependencies between work units and automatically creates a new version for projects, models, scenarios, and global objects when needed for the project release. Finally, you can create a version of your solution to safeguard it.

**All versions of work units are stored in the Master Repository. Developers can restore copies from the Master Repository into any Work Repository attached to that Master Repository. This structure allows developers to safely and securely replicate work units between repositories while maintaining the current versions in one central location.**



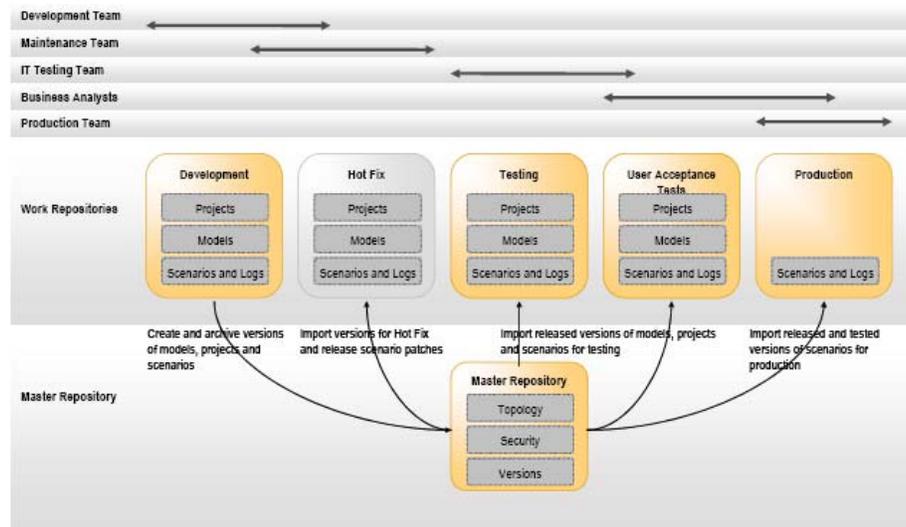
**Figure 2: Adding the Data Warehouse project to the solution automatically includes all its required elements.**

## MANAGING LARGE DEVELOPMENT ENVIRONMENTS

### Maintaining Consistency

Oracle Data Integrator allows you to smoothly manage even the largest and most complex development environments. When you work in a large environment, it is easy and safe to pass objects between repositories during the development process.

Oracle Data Integrator creates versions in one repository and then restores them somewhere else. In this way, you preserve consistent versions of the source (project) and of the generated code running in production (scenario) in the same place. Figure 3 below illustrates this process.



**Figure 3: Using Oracle Data Integrator in a comprehensive development environment.**

### Securing Projects

In a large development environment involving multiple repositories and users, it is important to set up appropriate privileges for project managers, developers, and other users. You can define access privileges for Oracle Data Integrator’s version management features using Oracle users and profiles.

To create additional protection against catastrophic loss or damage of your projects, you can export solutions to a remote environment or store them in an external version management system.

### CONCLUSION

Securely maintaining version management throughout an integration project’s development lifecycle poses challenges and risks. With large-scale projects increasingly being developed by multiple people located in different locations around the globe, these risks are compounded. By using Oracle Data Integrator version control, development teams can virtually eliminate these risks and concentrate on producing and releasing a superior integration processes on schedule.



Promoting Product Integrity with Reliable Version Management: An Oracle Data Integrator Technical Brief  
Updated December 2006

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.