White Paper

Data Integrator:
Migration Practices

BUSINESS OBJECTS

SENSE CORP

Author: Sense Corp
Contributors: Peter Siegel, Alicia Chang, George Ku
Audience: Data Integrator Administrators & Developers
Date : October 2004
Based on BusinessObjects Data Integrator version 6.5

Contents

Executive Summary

The purpose of this document is to provide guidance on migrating objects from one repository to another in a multi-user environment with local repositories. This document is intended to be a supplement, not a replacement, to existing BusinessObjects Data Integrator (BODI) technical documentation. This paper is based on BusinessObjects Data Integrator 6.5, and the general content (not necessarily each screen shot) is compatible with all currently supported versions.

Assumptions

- Environment is either 2 or 3 tier Development/Production environments

- Methodology or change management documents are a part of the migration process

- Migration from one environment to the next should occur when users are NOT logged in the target repository

- Migration of jobs, workflows, or dataflows will be performed using the BODI export feature from one repository to another repository (as opposed to exporting to a *.atl file first and subsequently importing into another repository.

- All pertinent objects, table structures, scripts, etc. will be backed up prior to Production exports
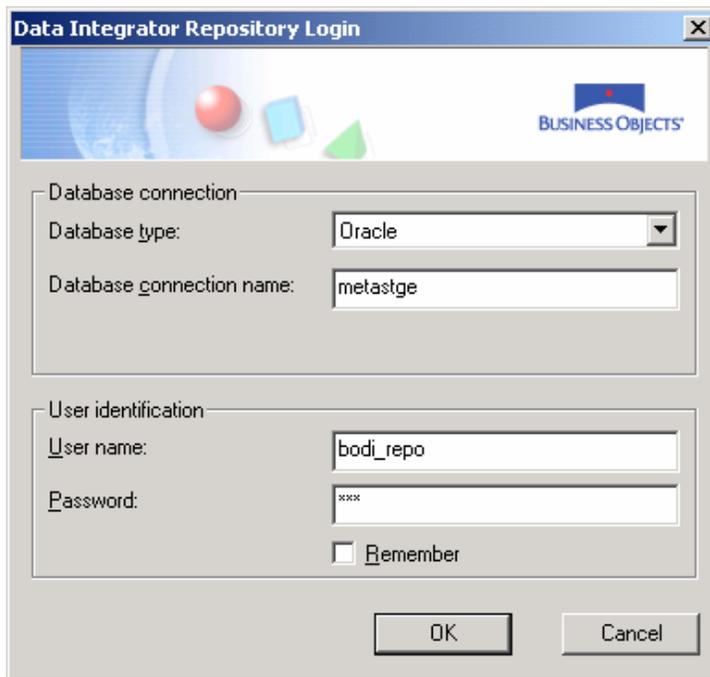
Repository/Table Backup

Before performing migration it is a best practice to backup affected repositories and tables. This is a precaution so that if an incorrect version of BODI code or table is migrated and written over a desired existing version, the previous version can be restored.
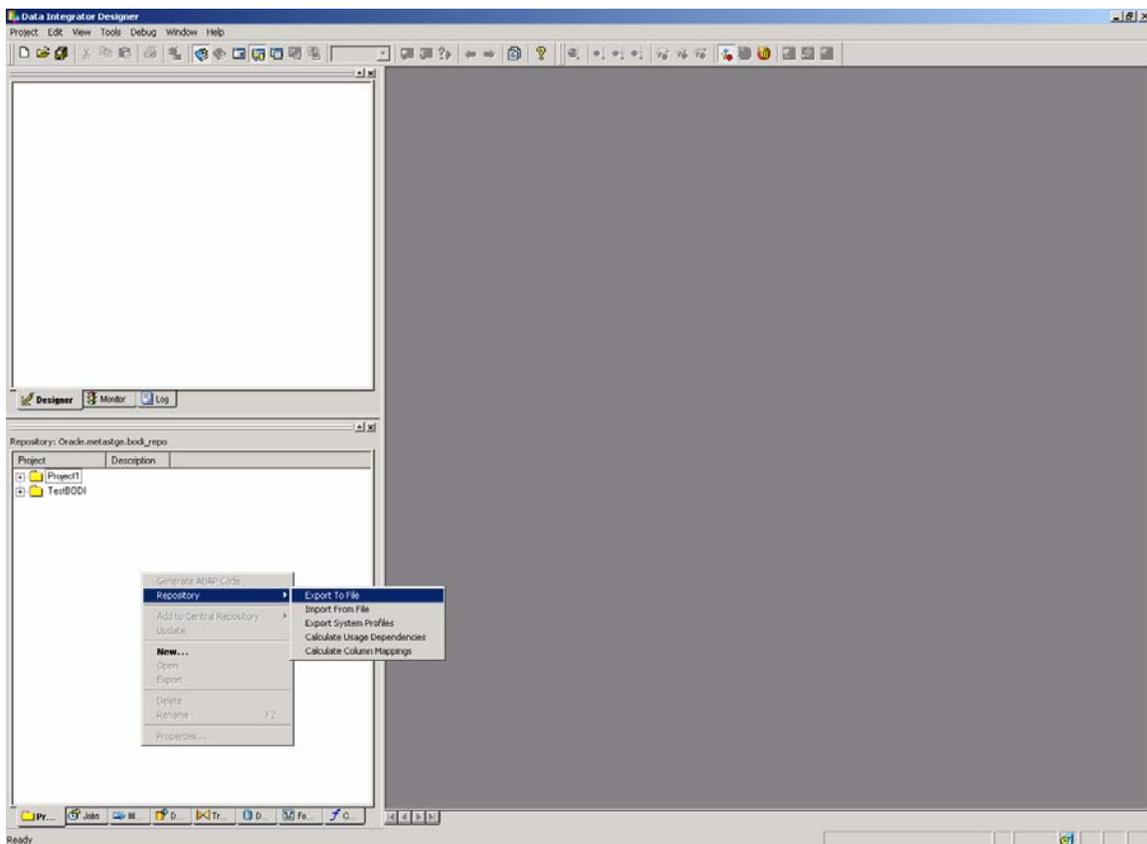
Repository Backup Steps

- Log into the repository to which you are migrating. For instance, if you are migrating to Production, then log into the Production repository.

- After logging into the repository, use the export file utility in the object library to export the metadata to an *.atl file. Save the file in a place where you can retrieve it easily (typically on a network or zip drive).

- The metadata for the objects you've developed is now saved. If you need to restore this repository, the metadata and can be imported using the import from file utility in the shortcut menu on the object library window.

- Backing up your repositories should be a standard and frequent practice, especially if you perform a lot of development. It is easier to recover a day's worth of work versus a week's worth of work.

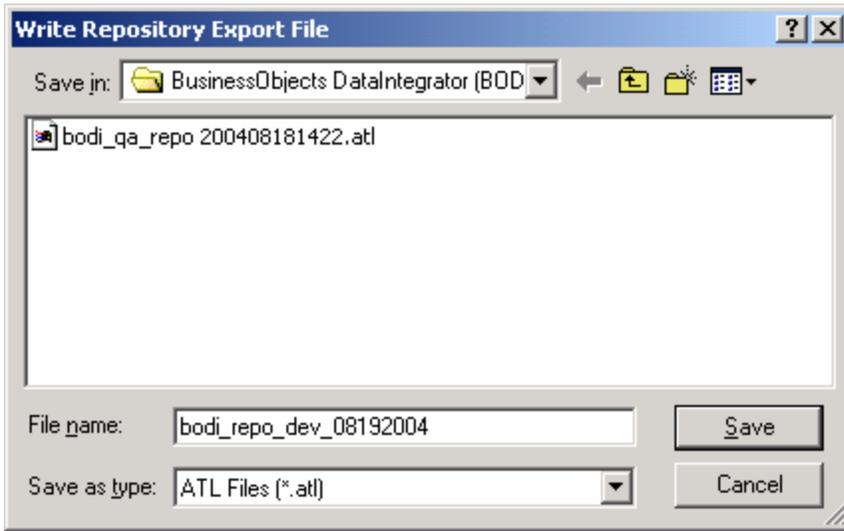The specific steps to backup your repository are as follows:

Log into the export destination repository:

Right-click in the object library in your repository to get the 'Export to file' option:
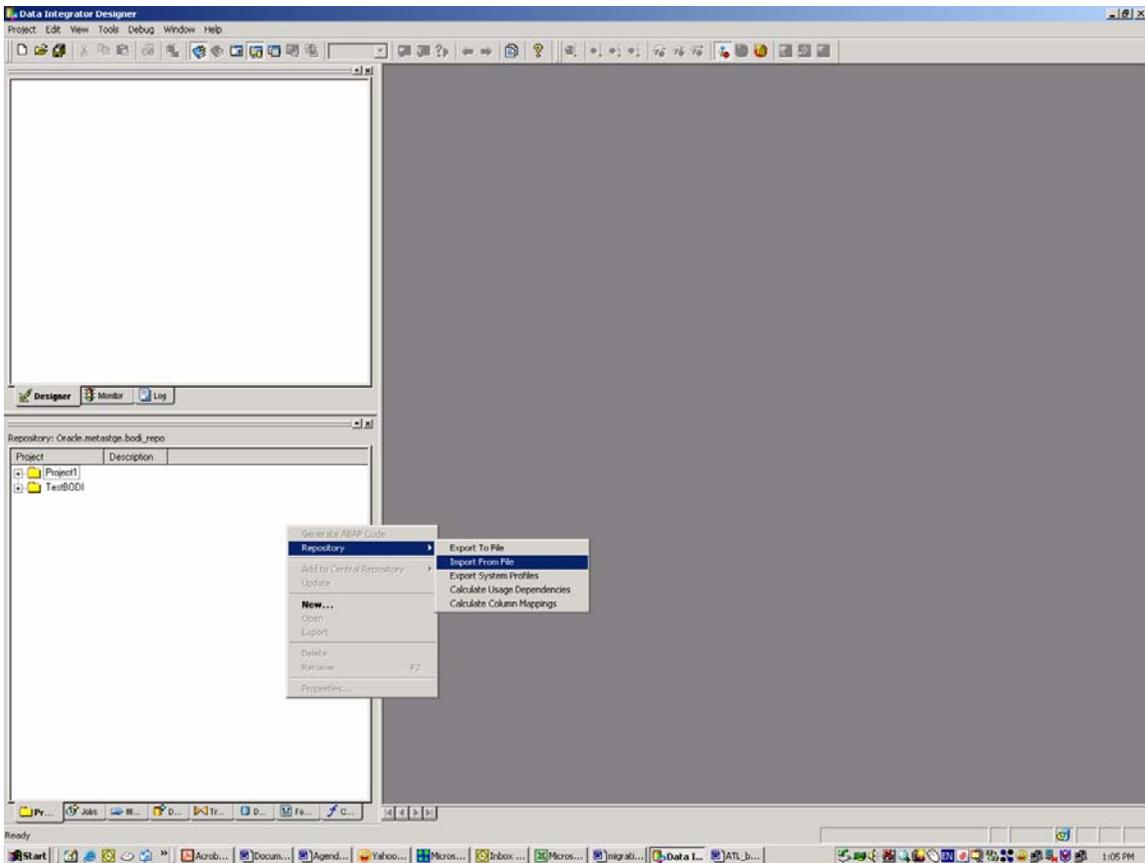
Save the file with the repository name and date in a folder that is readily accessible and choose 'Save'.
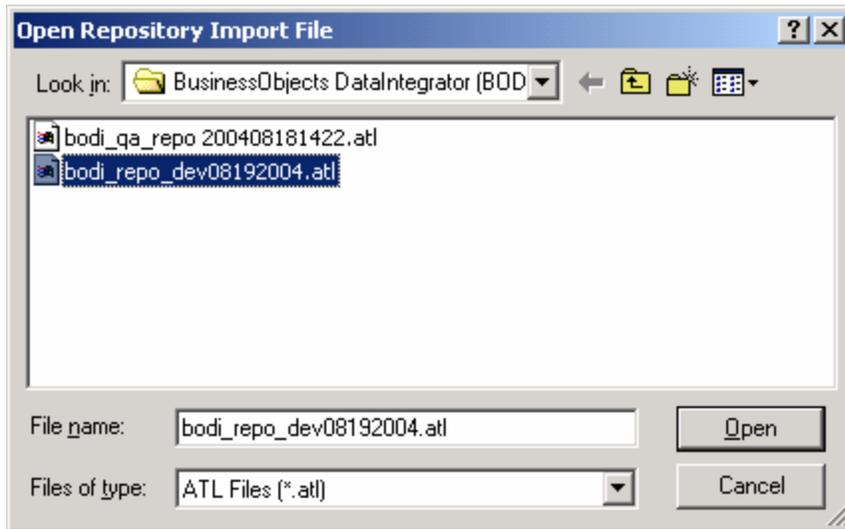

If you must restore from a backup *.atl file, the specific steps are as follows:

Open the desired repository and right-click in the object library in your repository to get the import file option.

Choose the .atl file that you want to import and then click 'Open".

You will see a warning message that you are about to overwrite the existing metadata in your repository.   If this is what you want to do, then click 'OK'. The import may take a while if there are a lot of objects to import.

Steps to create backup for your tables in an Oracle target:

- Log into Toad or SQL plus for the data schema

- Back up Oracle tables

- If the table is too large for a quick back-up, your Oracle space permissions and/or existing database table space, be sure a cold-back up of the data tables has already been done or you have the ability to roll back.

- If the table is not large and can be backed-up without much effort or space, then use the example below:

  Create table Sales_bak as
  *(Select * from Sales)*

Migration Process

## *Migration Checklist:*

- Make sure all repository and database back-ups have been performed prior to migration, especially when migrating to Production.

- Make sure all developers are disconnected from of the destination repository.

- Make sure all pertinent documentation (design, testing, and migration documents) has been completed and reviewed. This is especially important when you do your final migration to Production.

Note: A sample migration checklist is provided in Appendix A.

Migrating BODI Objects to Other Environments Steps:

Log into the repository from which you want to migrate objects.
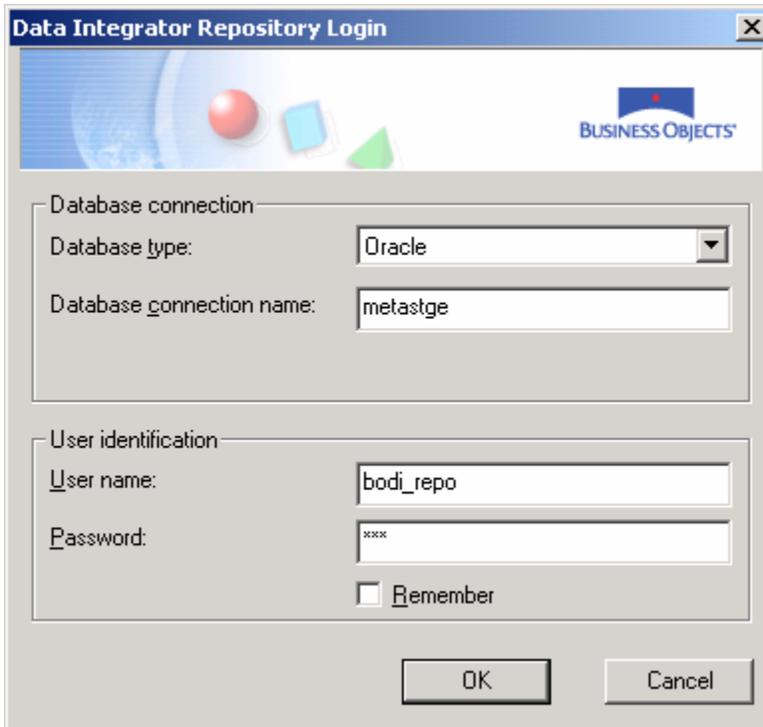
Choose the job in the object library and right click to get the shortcut menu.
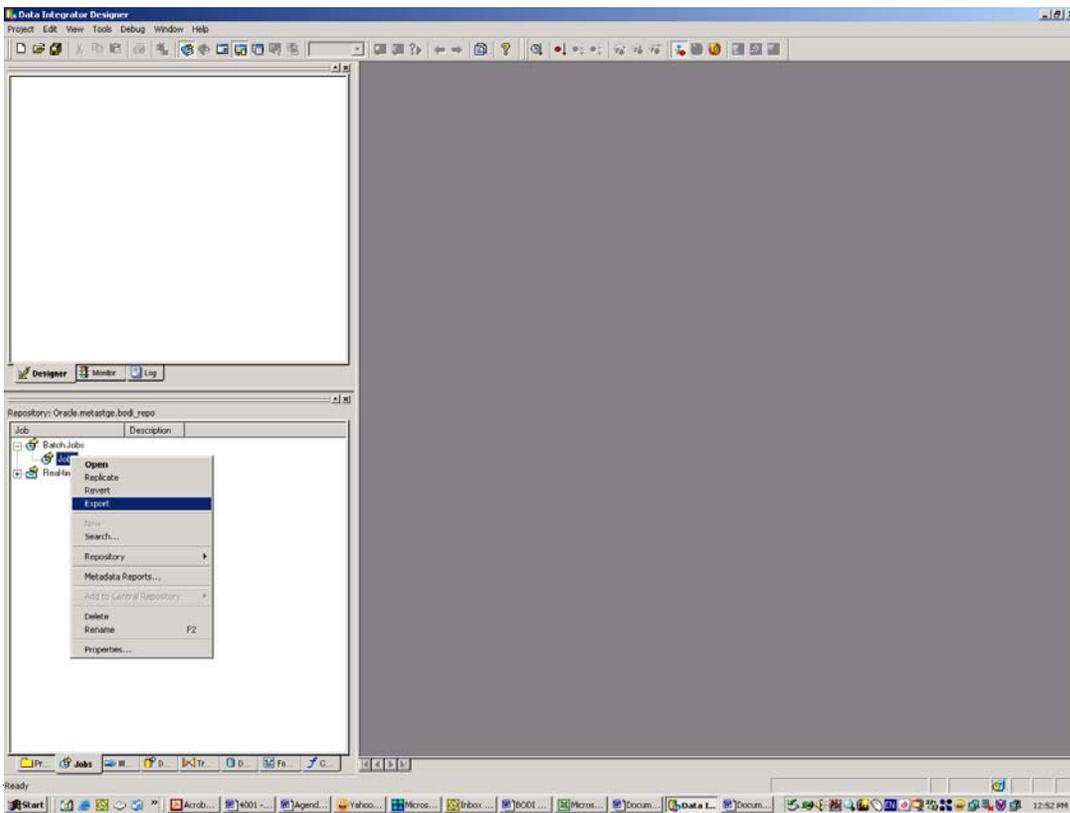
In the export window, choose the job, right click again, and choose 'Export'. The Export destination window opens.
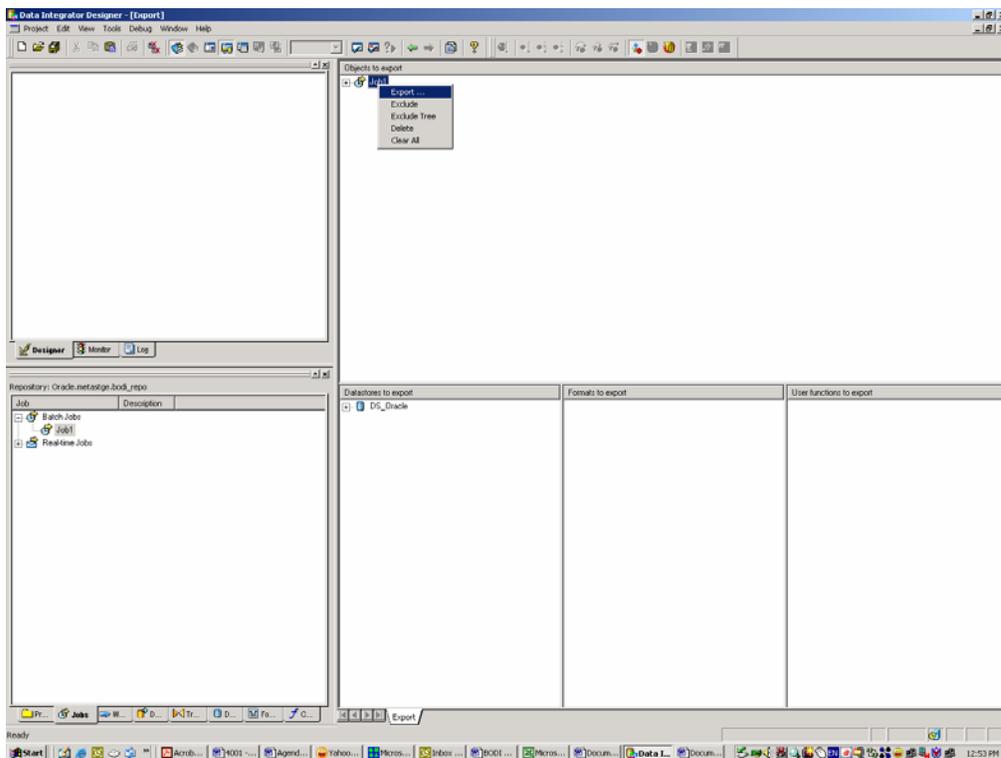
Log into the target repository and click 'Next'.
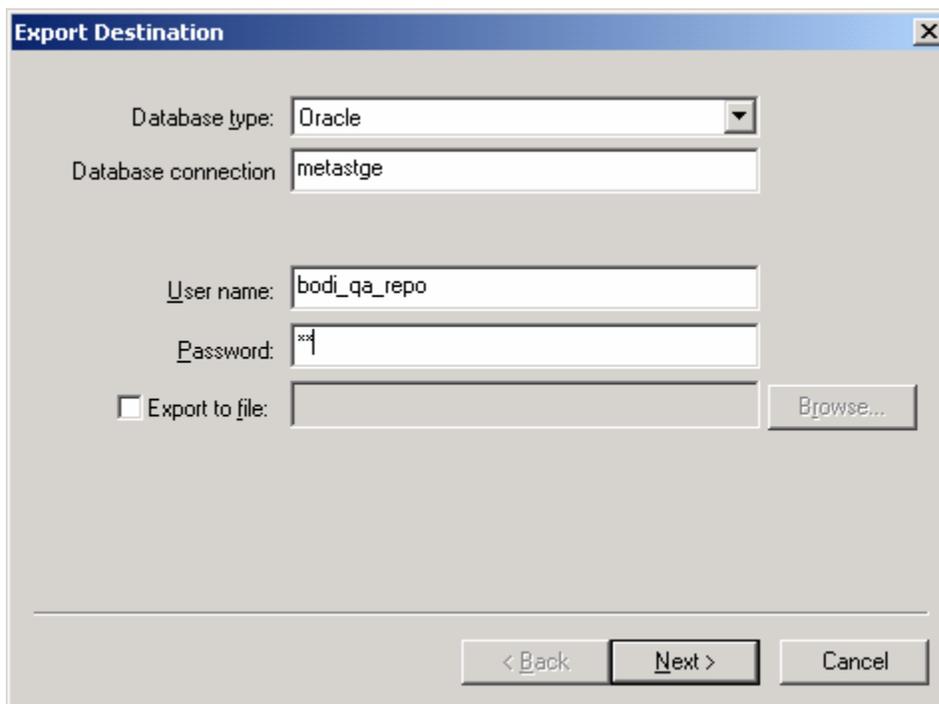
Choose the top-most node to export and click 'Next'.
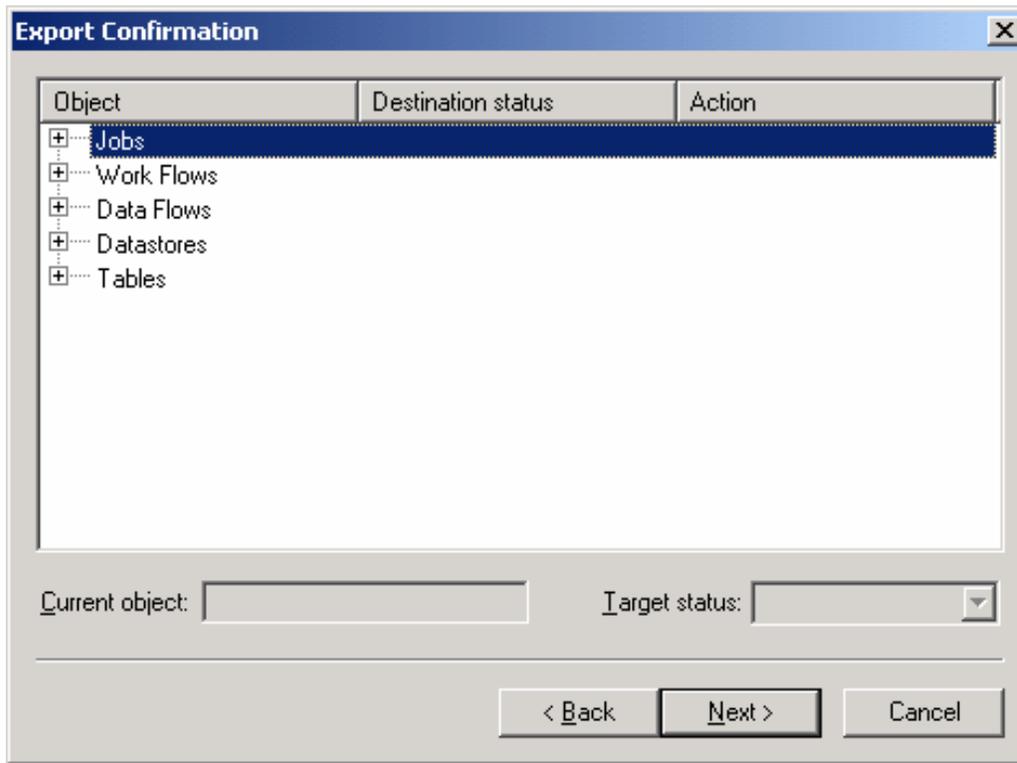
When you get to the Datastore Export Option Window, click Advanced. This gives you the option to choose the connection to your data on your database.
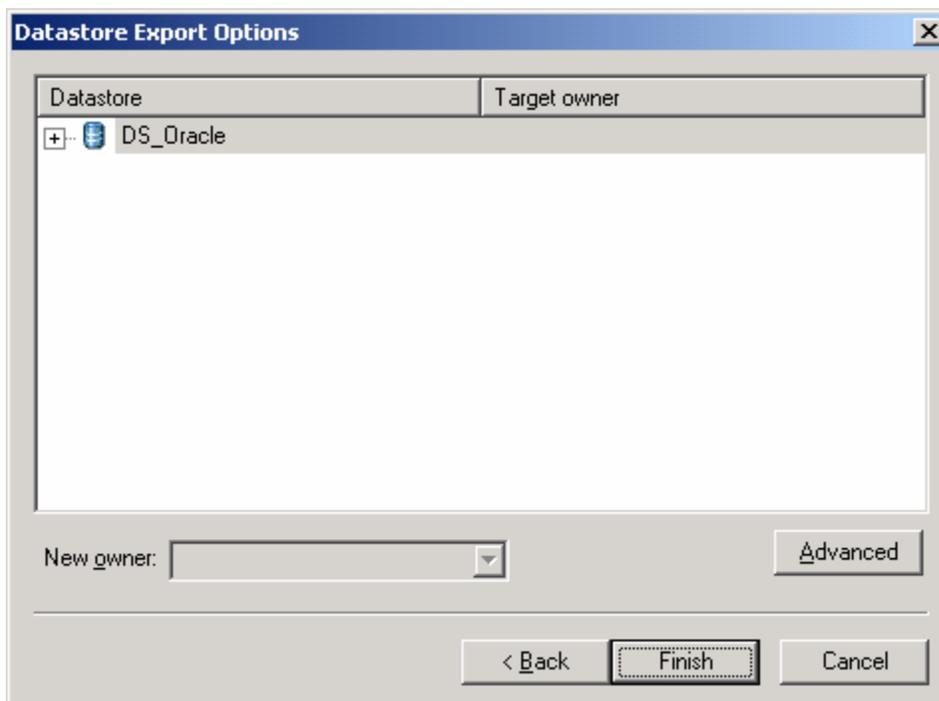
Make sure that the connection to the database is accurate. Verify this after you have migrated objects and as you are migrating in the Datastore Editor.
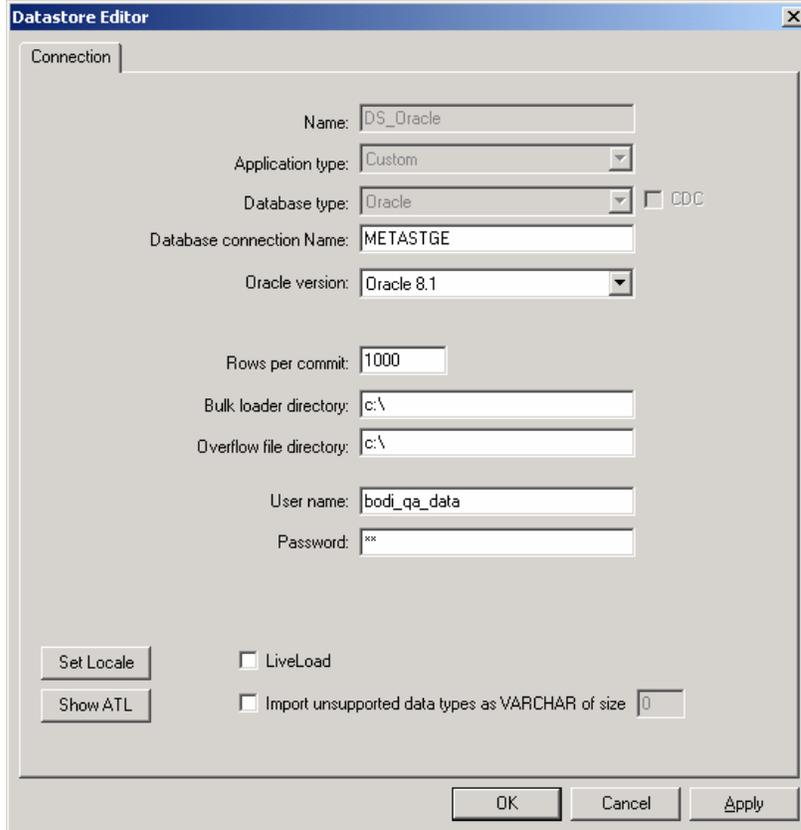
After migration, an output window appears to tell you how many objects were exported, or if there were any errors during the export process.
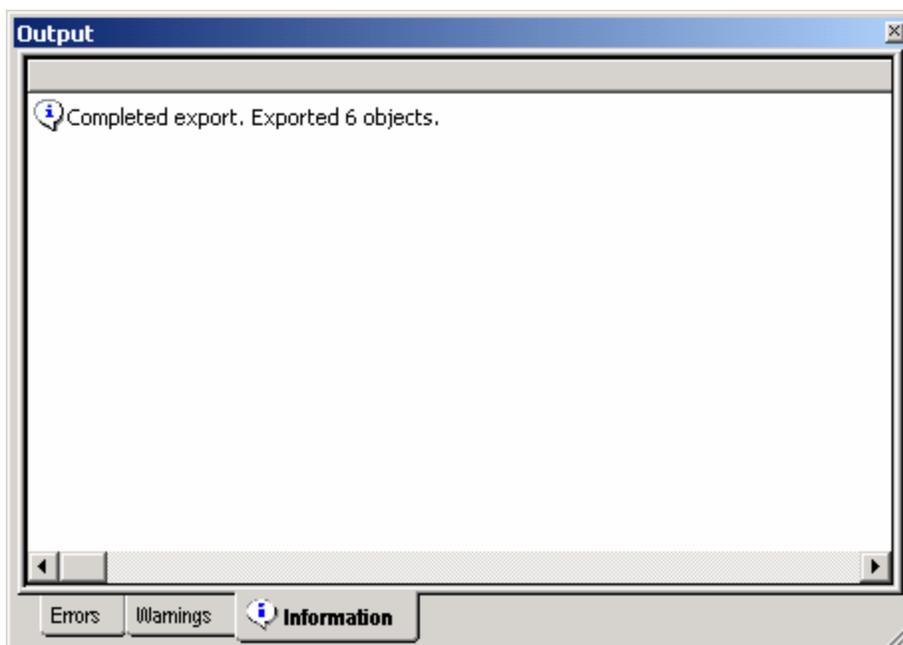
Miscellaneous

- When choosing datastore connections for the data (either source or target) in the DataStore Data Options during export, make sure you are pointing to the correct data.

- As you develop and export to Production, keep a running log of what was exported, who exported, and the date it was exported. This "change log" is useful in case any future errors or issues arise in that it allows you to track the developer and time of change.

- As a part of good methodology, make sure to include a migration document along with design and testing documents.

- To ensure fast and seamless migration, use a common naming convention across all systems and phases of your development environment.

Conclusion

Business Objects Data Integrator offers a robust development environment allowing for multiple user repositories and environments. Developers can create Data Integrator jobs in their own repositories and migrate them through different levels of implementation, such as Development, Quality Assurance, or Production.

With Data Integrator's built-in Export utility, the migration process is relatively quick and easy. Objects can be moved from repository to repository with no external manipulation. This reduces the possibility of migrating the incorrect version of a job to another environment.

This white paper is a step-by-step guide to migrating Data Integrator objects. See Business Objects Data Integrator technical documentation for more information on migration practices.

Appendix A

Production Migration Checklist

Table(s):
Workflow(s):
Dataflow(s):
ABAP(s)*:

Directions: For every workflow migrated to production, fill in each line with a check (✓) for complete or "NA" for Not Applicable.

Before Migration:

___ ABAP Migration Form complete

___ Staging Migration Form complete

___ Target Migration Form complete

___ Approved for Migration (validate with analyst and/or migration manager)

Migration:

Oracle Table Migration:

New Table ("Net Change" or "BL-T"):

___ Determine appropriate table space settings

___ Move Oracle Table DDL to PD from ST (using TOAD)

___ Validate table Primary Keys

Existing "Net Change" Table:

___ DO **NOT** DROP Production Table

___ Back-up table (2 options)

    ___ Create *_BAK table and use INSERT INTO to load production data

    ___ Use SQL RENAME command to rename PD table to *_BAK

___ Generate new table structure (if different) (2 options)

    ___ Drop Table & Regenerate DDL

        ___ Determine appropriate table space settings

        ___ Move Oracle Table DDL to PD from ST (using TOAD)

        ___ Validate table Primary Keys

        ___ Re-insert data from *_BAK table (if necessary)

    ___ Use SQL 'ALTER TABLE' command to modify existing table

___ Determine if table reload required (yes if wrong data before or new columns added)

___ Determine reload start point (beginning of time, specific years, months, etc.)

___ Modify EXECUTIONSTATUS table to reload start point

Existing "BL-T" Table:

___ Back-up table (2 options)

    ___ Create *_BAK table and use INSERT INTO to load production data

    ___ Use SQL RENAME command to rename PD table to *_BAK

___ DO **NOT** DROP Production Table until backup table is created

___ Generate new table structure (if different) (2 options)

    ___ Drop Table & Regenerate DDL

        ___ Determine appropriate table space settings

        ___ Move Oracle Table DDL to PD from ST (using TOAD)

        ___ Validate table Primary Keys

    ___ Use SQL 'ALTER TABLE' command to modify existing table

SAP

___ Validate ABAP has been migrated to PCx environment(s)

___ Check SAP program version date/time created or last changed (ABAP Workbench menu) if issues

BODI

___ Export appropriate objects from ST to PD (Workflow or Job)

Staging:

___ Change BODI Data store Settings to PD

___ Verify connection settings to correct SAP server

___ Execute new object in Production Job (2 options)

    ___ Load historical data or

    ___ Test to see that job executes correctly (cancel job at execution window)

Target:

___ Change BODI Data store Settings to PD

___ Execute new object in Production Job (2 options)

    ___ Load historical data or

___ Test to see that job executes correctly (cancel job at execution window)

New Job (if needed)

___ Determine which nightly Control- M job group to put job in (12am or 1am for PRD)

___ Coordinate new job with Production Support (they need to update documentation)

___ Generate new Batch File for new job and edit script (use existing batch file job script as example)

___ Insert new Batch File into Control-M Files folder on the correct server (i.e. PRD = icsccc07)

___ Contact Control-M to put new job in Production nightly run

About Sense Corp

Sense Corp is a full services consulting firm that helps clients improve their performance utilizing business intelligence, system integration, and data warehousing solutions. We develop long-term relationships, which bind us to the quality of our work. Our team members combine business and technical expertise as well as structured methodologies to implement common sense solutions.

Customer success dictates Sense Corp success.

For further information about Sense Corp, visit its web site at: www.sensecorp.com

About Business Objects

Business Objects is the world's leading business intelligence (BI) software company. Business intelligence enables organizations to track, understand, and manage enterprise performance. The company's solutions leverage the information that is stored in an array of corporate databases, enterprise resource planning (ERP), and customer relationship management (CRM) systems.

Popular uses of BI include enterprise reporting, management dashboards and scorecards, customer intelligence applications, financial reporting, and both customer and partner extranets. These solutions enable companies to gain visibility into their business, acquire and retain profitable customers, reduce costs, optimize the supply chain, increase productivity, and improve financial performance.

In December 2003, Business Objects completed the acquisition of Crystal Decisions, the leader in enterprise reporting. The combined product line includes software for reporting, query and analysis, performance management, analytic applications, and data integration. In addition, Business Objects offers consulting and education services to help customers effectively deploy their business intelligence projects.

Business Objects has more than 24,000 customers in over 80 countries. The company's stock is traded under the ticker symbols NASDAQ: BOBJ and Euronext Paris (ISIN: FR0004026250 - BOB). It is included in the SBF 120 and IT CAC 50 French stock market indexes. Business Objects can be reached at +1 800 877 2340 and www.businessobjects.com.