

MIGRATING APPLICATIONS AND DATA TO ORACLE9i OLAP CASE STUDIES

*Dan Vlamis, Vlamis Software Solutions, Inc.
dvlamis@vlamis.com*

INTRODUCTION

With the introduction of Oracle 9i and BI Beans, the landscape for development of BI applications has changed. This leaves clients with existing Express-based applications asking "How do we move our applications and data?" and "Do we store our OLAP data in RDBMS tables or analytic workspaces?". This paper will address the issues involved in converting and migrating these types of applications over to the new technology.

TYPES OF EXPRESS APPLICATIONS

Express-based applications generally fall into one of the six following categories:

- Character-mode Pure Express Applications
- Visual Basic or C applications with Express "back end"
- Express Web Agent Applications (coded in HTML or Express programs that generate HTML)
- Oracle Express Analyzer (typically relatively simple applications for ad-hoc analysis of data)
- Oracle Express Objects (GUI environment for creating complex applications)
- Oracle Applications – Oracle Financial Analyzer (OFA), Oracle Sales Analyzer (OSA), Oracle Demand Planning (ODP)

While details of the structure of the data may differ depending on the nature of the Express application, the general nature of moving the data is generally the same.

MIGRATION OVERVIEW

In order to migrate an Express-based application to Oracle9i OLAP, several areas must be addressed:

- Migrating the data – This is actually one of the easier tasks, but can be time consuming
- Migrating the metadata – This involves what are dimensions, hierarchies, variables, etc.
- Migrating the application logic – Express programs can be migrated, but logic elsewhere can present problems
- Migrating the front end – Generally, this has to be re-coded, but is actually not as time-consuming as you would think
- Migrating user reports and settings – These typically have to be re-implemented

While details of the structure of the data may differ depending on the nature of the Express application, the general nature of moving the data is generally the same.

MIGRATING THE DATA – STAR SCHEMA OR ANALYTIC WORKSPACE?

In Oracle 9i OLAP, you can now enjoy the benefits of multidimensional access to your data and still store it in the Oracle RDBMS. Companies that have Express databases need to evaluate whether they should keep their data in Express, use analytic workspaces, or migrate the data to the Oracle RDBMS. Of course, if your data is truly stored in an RDBMS and you access the data in Express via the Relational Access Manager (RAM), you don't really need to migrate the data, as it already is

in the RDBMS. In this case, you simply need to re-map the relational structures to 9i OLAP structures using the Cube Builder component of Oracle Enterprise Manager.

MIGRATING TO AN ANALYTIC WORKSPACE

Migrating data to analytic workspaces of Oracle 9i OLAP is very simple. EIF files can be written by Express and read by Oracle9i OLAP. All of this is clearly documented in the Oracle9i OLAP User's Guide, but all you generally do is export the data to an EIF file from Express and create an analytic workspace in Oracle9i OLAP and import the EIF file into the analytic workspace. The Oracle9i OLAP User's Guide is available on technet.oracle.com.

MIGRATING TO RDBMS TABLES

Migrating Express data to RDBMS tables is trickier because new objects must be created. As of this writing, there are no automated tools for creating a RDBMS schema from Express databases. Relational objects must be created for each Express object using a tool such as Oracle Warehouse Builder. Once these objects are created, the data can be migrated.

MIGRATING DATA VIA OLAP DML

A straight-forward way for migrating Express data is by executing OLAP DML commands. In order to do this, you simply execute a series of SQL INSERT statements, one for each cell of data. Use the SQL PREPARE and SQL EXECUTE statements to make this more efficient. It can be as simple as the following program:

```
DEFINE INSERTPRICE PROGRAM
PROGRAM

limit TIME to all
limit PROD to all
sql prepare s1 from insert into pricetbl values (:time, :prod, :price)

for TIME PROD
  if price ne na
    then do
      sql execute s1
      if sqlcode ne 0
        then continue
    doend
ENDs
```

MIGRATING DATA VIA FLAT FILES

Of course, a final method for migrating Express data is the tried and true method of writing data out to flat files and using the SQL Loader to read it back into the RDBMS. As arcane and old-fashioned as this may sound, often this method can be the most productive and efficient since: a) it naturally provides a way to assure quality since the intermediary file can be viewed, b) the SQL Loader is very efficient at loading large amounts of data and can be run in parallel (as can the Express sessions that write the data), and c) tools to accomplish this are readily available. In order to do this, you simply open a file using the OUTFILE OLAP DML command, and write out data using the REPORT or ROW commands. If using the REPORT command, be sure to use the NOHEAD option to remove headings that might produce unwanted header records. A simple program to report out a two-dimensional PRICE variable would be the following:

```
DEFINE WRITEPRICE PROGRAM
PROGRAM
```

```

outfile tempfile.txt
limit TIME to all
limit PROD to all
report nohead width 20 down PROD across TIME: PRICE
outfile eof
END

```

Be sure to take advantage of any conjoint dimensions that may exist to compress out any NA's in the data and consider even changing composites into conjoints (using the CHGDFN command) and using the key function to identify the key dimensions of the conjoint. If SALES is dimensioned as <TIME <PRODGEOG PROD GEOG>>, you can write it out with the sample program:

```

DEFINE WRITESALES PROGRAM
PROGRAM

outfile tempfile.txt
limit TIME to all
limit PROD to all
limit GEOG to all
chgdfn PRODGEOG dimension
report nohead width 20 down key(PRODGEOG PROD) -
          width 20 key(PRODGEOG GEOG) -
          across TIME: SALES

outfile eof
END

```

AW OR STAR SCHEMA?

So which storage method is best? Unfortunately, little data exists at this writing to provide a real answer. The easiest migration for Express data is to move the data to an analytic workspace via an EIF file. With SQL extensions in Oracle 9i OLAP release 2, the data in analytic workspaces is accessible via SQL, so perhaps the data can stay in an analytic workspace. Until better performance data is available, it is difficult to provide an answer on whether to move data to the RDBMS in a star schema or whether to keep data stored multi-dimensionally. Performance characteristics will vary by release, but in one study, we found that for a six-dimensional schema with very sparse data, performance was considerably better using an analytic workspace over a star schema. In general, we believe query performance is better with analytic workspaces than with relational storage for schemas with more than four dimensions.

MIGRATING THE METADATA

Migrating the metadata involves registering with 9iOLAP the various dimensions, hierarchies, measures, etc. that comprise a database. Unfortunately, this area does not have a ready-made solution. For example, you may be able to create an analytic workspace that contains all of your Express data, but in order to use that analytic workspace with BI Beans, all of the dimensions, hierarchies, levels, parentage, variables, etc. must be defined in the OLAP Catalog. This can be performed with a series of calls to CWM2 calls. Oracle is working on a converter that automates this procedure, but as of this writing, no details exist on exactly how much this will cover. In addition, newer releases of Oracle Warehouse Builder will help automate this task.

MIGRATING THE APPLICATION LOGIC

Application logic that is housed in Express code may be able to be simply moved to Oracle9i OLAP and run as-is. The Express SPL language is left whole in what is now referred to as the "OLAP DML". This means that Express programs that contain MODELS and FORMULAs with logic that manipulates the data can be left intact. Unfortunately, often, application logic often creates new data. While commands such as DEFINE and MAINTAIN will still run without any modification in Oracle9i OLAP, there may be other tasks that have to be performed such as registering the new data in the OLAP catalog. These can be performed with the CWM2 series of commands, but will require a significant amount of work.

MIGRATING THE FRONT END

Express-based applications generally fall into one of the six following categories:

- Character-mode Pure Express Applications
- Visual Basic or C applications with Express "back end"
- Express Web Agent Applications (coded in HTML or Express programs that generate HTML)
- Oracle Express Analyzer (typically relatively simple applications for ad-hoc analysis of data)
- Oracle Express Objects (GUI environment for creating complex applications)
- Oracle Applications – Oracle Financial Analyzer (OFA), Oracle Sales Analyzer (OSA), Oracle Demand Planning (ODP)

CHARACTER-MODE PURE EXPRESS APPLICATIONS

Very few character-mode pure Express Applications are left. This technology became somewhat obsolete in the late 1980's when graphical user interfaces became more common. The main elements of driving a character-mode user interface, the table and form commands and the picklist qualifier of the get function were removed in version 6.0 of Express. Migrating a character-mode pure Express application consists of isolating the user interface sections of the code from the rest of the application. That U/I code can be replicated in Java code or OLAP Web Agent. The remaining Express "back-end" code may be simply ported to Oracle9i OLAP DML by exporting the code to an EIF file. Any code that communicates with the operating system must be examined as much of this has changed in Oracle9i OLAP. In one particular case study, Vlamis Software is planning to replicate much of the functionality of the old Express TABLE command in a Java bean, allowing existing Express code to run. This allows an application based on the TABLE command to operate in the new Oracle9i OLAP environment.

VISUAL BASIC OR C APPLICATIONS WITH EXPRESS "BACK END"

These applications have a front end component that communicates to Express via SNAPI (Structured N-dimensional Application Programming Interface) calls. The SNAPI interface is not available in Oracle9i OLAP and has generally been replaced with the Java OLAP API. These applications can have the Visual Basic or C "front end" code be re-coded in Java and pure back-end code can be simply ported to Oracle9i OLAP DML by exporting the code to an EIF file. In practice, however, often the line separating back end and front end code is relatively fuzzy in these applications, however, making the determination of what code can be retained difficult.

EXPRESS WEB AGENT APPLICATIONS

There is already a viable option for migrating Express Web Agent apps: Oracle has released Oracle9i OLAP Web Agent, which allows existing Web Agent applications to run under Oracle 9i OLAP. This option only uses the analytic workspace and does not use the OLAP Catalog for storage of the OLAP metadata. This essentially allows for the import of Express data and Web Agent programs into the analytic workspace and runs the applications using an updated version of the Express 6.3.4 Web Agent application code. If you want to use BI Beans or SQL access alongside such legacy Express Web Agent code, then you will need to create the necessary OLAP Catalog metadata and SQL views.

ORACLE EXPRESS ANALYZER

These really aren't applications; rather, they are a series of ad-hoc reports. These can be readily replaced with a series of presentations created in BI Beans. There is currently no automatic way to migrate the reports themselves—they have to be recreated in BI Beans. Ad-hoc query tools such as VSS Business Analyzer are good replacements for these applications.

ORACLE EXPRESS OBJECTS

Applications coded in Oracle Express Objects can be re-coded in Java using BI Beans. If the OEO GUI environment was extensively used, this will likely require a thick Java application, since the JSP environment does not offer as much control over the GUI environment as a thick Java application. An article to appear on OTN with a working title of "Developing the Next Generation Business Intelligence Applications with BI Beans and Oracle 9i OLAP" by Chris Claterbos describes the migration of an Oracle Express Objects application.

ORACLE APPLICATIONS – ORACLE FINANCIAL ANALYZER (OFA), ORACLE SALES ANALYZER (OSA), ORACLE DEMAND PLANNING (ODP)

These applications are being replaced by Oracle's Enterprise Planning and Budgeting (EPB) application. Since this product is not slated for release until later in 2003, details are sketchy, although in later releases, Oracle plans to have a migration path available for customers with OFA, OSA, or ODP. It is clear, however, that EPB will be more tightly integrated with the rest of the Oracle E-Business Suite, easing data loading issues for Oracle E-Business Suite customers, but likely complicating the environment for OFA, OSA, or ODP customers that load data from other sources.

MIGRATING USER REPORTS AND SETTINGS

In most OLAP applications, users have the ability to modify reports and graphs and save them back to a central repository. This is implemented using the BI Beans repository with BI Beans applications. There does not currently exist an automatic way to migrate these reports, graphs, and selections into the BI Beans repository. In time, perhaps tools will exist to automate this process, but for now, reports must be recreated using BI Beans or in a JSP and saved in the repository.

CASE STUDIES

Few migrations have really been performed as of this writing to draw conclusions from, but various projects are underway. In one project, an pure character-mode Express application is being ported to Oracle9i OLAP. In this implementation, the company had a large investment in the Express SPL code and wanted to retain that investment. Their application generated TABLE commands on the fly to present multidimensional data. We chose to code a replacement for the Express TABLE command in Java to preserve the investment in the Express code. Once we have that TABLE command, we only need to recode the menu system itself, which is very simple in JDeveloper. As the project is still underway, few conclusions can be drawn yet.

In another case study, a web agent application was examined to determine the modifications necessary to migrate it to Oracle9i OLAP. As discussed previously, Express Web Agent applications are good candidates for migrating to Oracle9i OLAP Web Agent, since the API is simply ported to Oracle9i OLAP. In this case, there are complications from: 1) operating system calls (actually, easily overcome), and 2) HTML calls involving extra packages that have to be investigated further. In any analysis of a Express Web Agent application, look for extra packages supplied by outside vendors and experiment to see whether these will work with Oracle9i OLAP or how readily these packages can be used or replicated in the Oracle environment.

CONCLUSION

While few case studies exist in migrating Express applications to Oracle9i OLAP, clearly, there is a path for applications. Since the Express Web Agent API remains intact in the Oracle9i OLAP Web Agent API, these applications are the easiest to migrate. Toolkits for migrating data and metadata will ease the burden. Oracle Warehouse Builder, Oracle Enterprise Manager, and Oracle Warehouse Builder can be used in the meantime to help migrate the data. Often, with these applications it is time to make changes that require reimplementing anyway.