Excel and Oracle

Paul Willyams

Senior Manager

Deloitte Touche Tohmatsu

This paper describes a funky Excel Add-In which uses Oracle Objects for OLE. I developed the add-in progressively over several Oracle projects. Initially I used the tool to support implementation work, such as ad-hoc queries and conversion uploads. Some conversions we have undertaken include:

- Historical general ledger balances through GL_INTERFACE
- Item master records through MTL_SYSTEM_ITEMS_INTERFACE

The add-in has developed to the extent that two clients now use the code for production systems –

- Download of GL codes and employee hierarchies for myPCard.com
- Upload of journal transactions from myPCard.com to General Ledger
- Upload of asset meter readings into Enterprise Asset Maintenance
- Dynamic extract of Balanced Scorecard data and narrative assessments for Board Reporting[1]

The add-in currently provides the following functionality:

- Run a short query entered in cell A1 on any sheet, with the output going directly to the sheet, such as select vendor_name, segment1as "Number" from po_vendors
- Enter a long query into a comment in cell A1 on any sheet. You can substitute parameters such as &period_name. Any content in cell A1 is appended, such as order by vendor_name
- The option to output vertically, especially for queries returning more than 256 columns
- Warnings if a large number of rows has been returned, allowing you to continue or abort
- Enter a query into a formula, such as =query("select count(*) from gl_interface where status=" & B2)
- Create an array formula, which can return several columns and rows, such as =query("select status, count(*) from gl_interface group by status").
- Use predefined formula, such as =GetFVSID(set_of_books_id, segment_name) to return the flex value set id for an account flexfield segment.
- Define and use your own formula, which call the generic query function
- Automatically create a data entry sheet for any Oracle table (usually an Open Interface table). Surplus columns can be deleted without adverse effect
- Upload from the data entry sheet to Oracle. Rows with errors are flagged and can be loaded again once corrected.

[1] See paper 9 – Using Oracle Balanced Scorecard in Anger

The speed of the tool is impressive. I have loaded journals with more than 10,000 rows in a few minutes. A Balanced Scorecard Report with about 70 dynamic array formula will calculate in about five seconds. A query returning two columns and 8,000 rows completes in about 2 seconds.

The user logs in as an Oracle user, not an Applications user. Although you can login as read-only, I prefer to add Oracle security to that. Therefore for production use, I get a specific Oracle user set up with appropriate access rights. The connection is kept open so the user doesn’t have to log in for each query, and is closed automatically when they exit Excel.

The underlying middleware is Oracle Objects for OLE (OO4O). This software is similar to an ODBC driver, but provides much faster performance and wider access to Oracle features, including seamless access to PL/SQL. As an OLE application, it is easy to work with in Visual Basic for Applications. See technet for details and downloads. OO4O is already installed with some Oracle Applications software.

The main reason for using Excel is to leverage a widely used and powerful interface. We could have met some of the extract and reporting needs using reporting tools such as Discoverer and perhaps exporting the results to Excel. We could have converted our data by writing to a flat file then using SQL*Loader or Data Loader[2]. Each of these tools could be appropriate for a given situation. But many times I find that we can accomplish the task directly and without technical assistance using the add-in.

For me, the add-in has become an indispensable part of my Oracle work and has enabled our clients to create some easily used, cost-effective solutions.

At this stage the Excel add-in is not a product you can buy (nor beg), but you can contact me (pwillyams@deloitte.co.nz) if you have a specific need to discuss. Any suggestions are always welcome.

[2] See paper 5 – Automating data loading into Oracle Applications for End Users