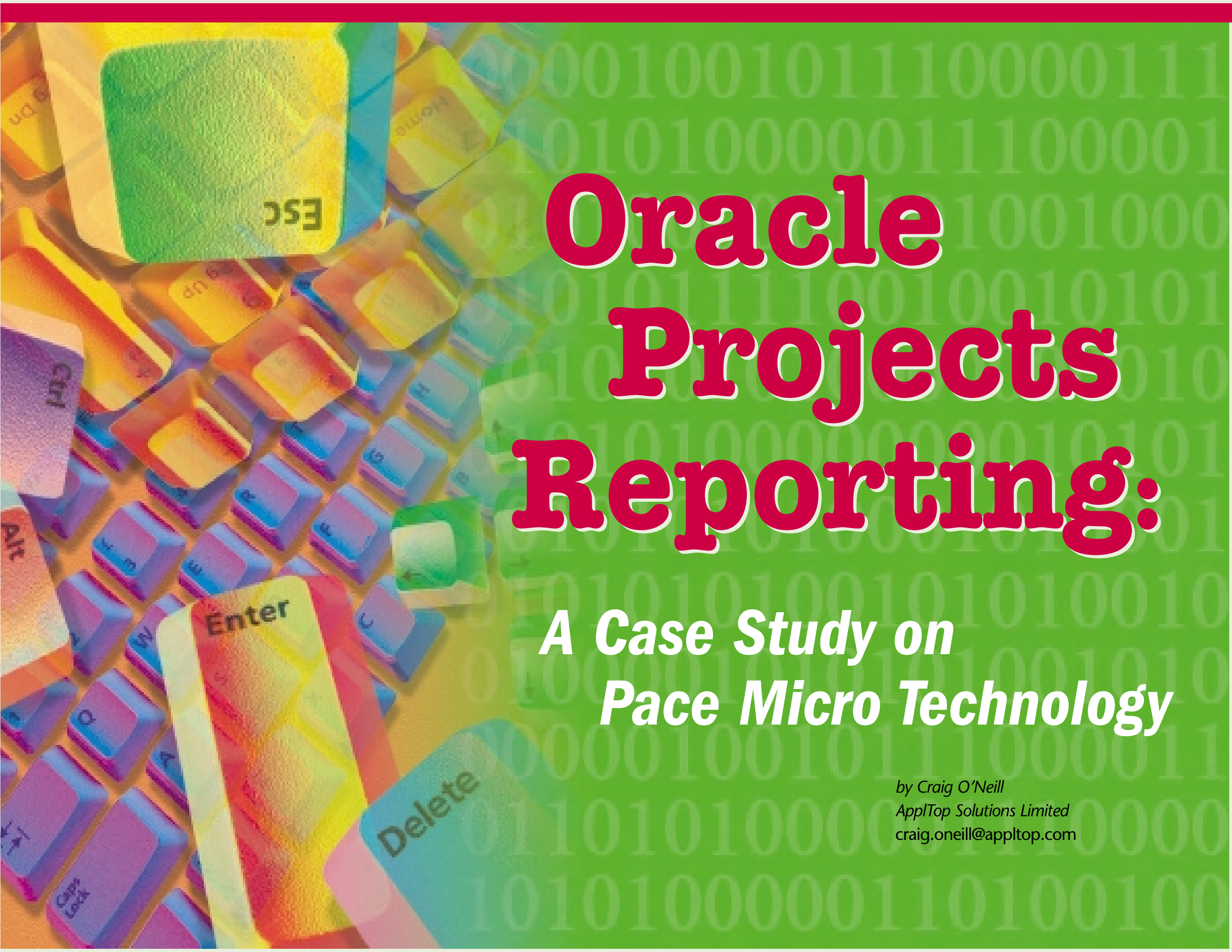


Oracle Projects Reporting:

A Case Study on Pace Micro Technology

by Craig O'Neill
ApplTop Solutions Limited
craig.oneill@appltop.com

> "As with all implementations of large-scale, "off-the-shelf" suites of enterprise applications, there are always situations where the applications do not quite meet the needs of the business."

Pace Micro Technology, a designer and manufacturer of digital set-top boxes, went live with the Oracle E-Business Suite 11*i* in March 2003. The overall implementation was a success and Pace is, for the most part, enjoying the benefits that 11*i* brings to the business.

However, as with all implementations of large-scale, "off-the-shelf" suites of enterprise applications, there are always situations where the applications do not quite meet the needs of the business, and some additional work is required.

Typically, these additional requirements are satisfied with complementary products, either from Oracle or from other third-party vendors, as well as a little customization here and there, along with a sizeable amount of custom reports. In this respect, the implementation at Pace was no different.

Product Details

ASL*Form is the big brother to another product called ASL*Rapport. ASL*Rapport is a scaled down version of ASL*Form that concentrates on reporting functionality. ASL*Form takes this to the next level and allows fully transactional screens to be created. The example solution discussed in this article could be produced using either ASL*Rapport or ASL*Form.

Pace has a small team within the IT department with the appropriate technical skills to be able to work with business process owners who can work together to fill any gaps in functionality within 11*i*.

One particular requirement that Pace had that could not be easily satisfied using their existing toolkit was a complex reporting extension to the Oracle Projects module.

Several Issues

Pace makes full use of the Oracle Projects module; in particular, Pace's project engineers use the standard Project Status Enquiry suite of screens within Oracle Projects to view critical project information.

Since 11*i* was introduced, Pace has endured several problems with using the standard supplied solution:

- Much of the information displayed is not native to Oracle Projects and therefore has to be imported from other modules; this in itself is cumbersome, error prone and can take a long time;
- The required data drill downs are not provided;
- Much of the information presented is not quite in the desired format; and
- The standard solution is generally slow to use.

Possible Solutions

Pace could have provided a solution for the above issues in a number of ways:

- Customize the existing forms;
- Rewrite the existing forms from scratch;
- Provide custom reports that the user could print to gain access to their information; and
- Use some out-of-Oracle solution.

Each of the above solutions suffers with one or more of the following problems:

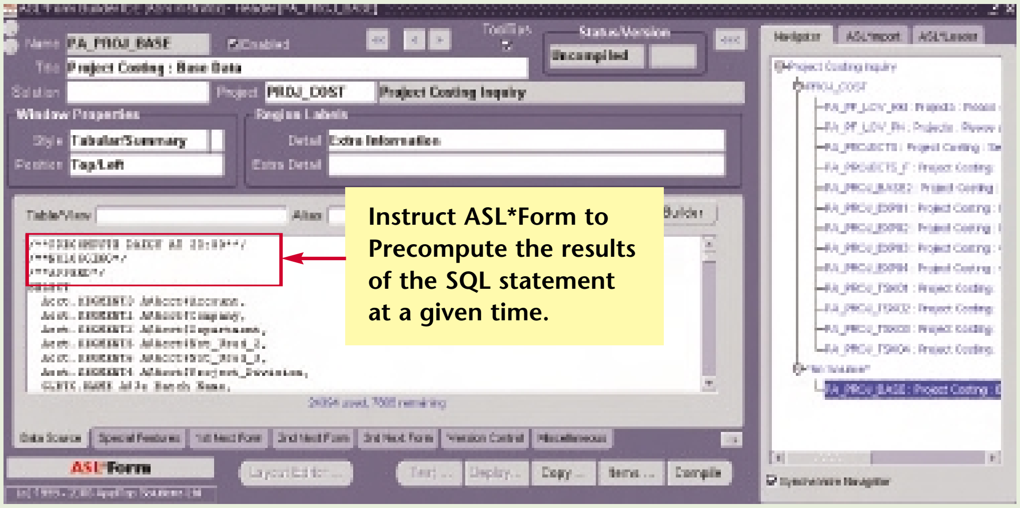
- Very expensive to produce;
- Very time consuming to produce;
- Troublesome and costly to support;
- Risky in terms of system integrity and risk of rework; and
- Would not provide the user with the best possible solution.

Due to the problems with all of the available options, Pace initially decided to make do with what was provided as standard.

The Chosen Solution

Pace recently added two products from ApplTop Solutions to their Oracle toolkit – ASL*Form and ASL*Accelerate. It is with ASL*Form that a solution was ultimately found. ASL*Form is the big brother to a product called ASL*Rapport. Although the chosen solution was created using ASL*Form, it could just as easily have been created with ASL*Rapport.

The requirements for the project were as follows:



Name: PA_PROJ_BASE ☑ Enabled

Title: Project Costing : Base Data

Project: PROJ_COST Project Costing Inquiry

Status/Version: Uncompiled

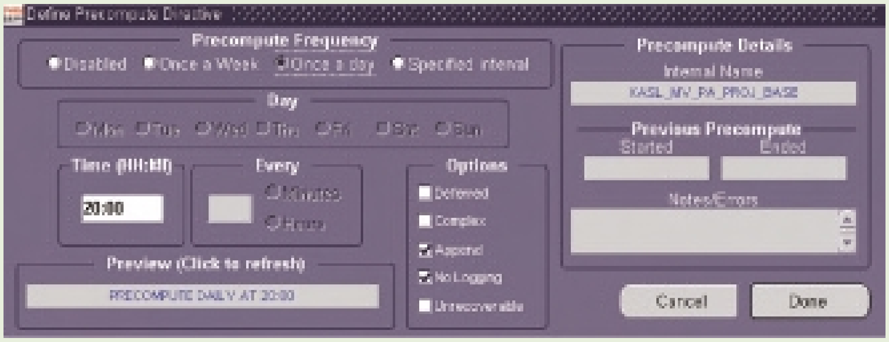
Window Properties — Style: Tabular/Summary; Position: Top/Left

Region Labels — Detail: Extra Information

```
/**PRECOMPUTE DAILY AT 20:00**/
/**NOLOGGING*/
/**APPEND*/
SELECT
```

Instruct ASL*Form to Precompute the results of the SQL statement at a given time.

Figure 1A: Solution Builder IDE



Define Precompute Directive

Precompute Frequency: ○ Disabled ○ Once a Week ● Once a day ○ Specified interval

Time (HH:MI): 20:00

Options: ☐ Deferred ☐ Complex ☑ Append ☑ No Logging ☐ Unrecoverable

Preview (Click to refresh): PRECOMPUTE DAILY AT 20:00

Precompute Details — Internal Name: KASL_MV_PA_PROJ_BASE

Figure 1B: Precompute Dialog

- Additional drill down and formatting issues should be satisfied;
- The data import process should be avoided if possible;
- The solution should be very responsive;
- The solution should be available directly on the end-users menu just like the standard solution;
- The solution should have the native Oracle 11*i* look and feel;
- The solution should be very cost effective;
- Development times should be short; and
- The solution should be able to be supported by Pace's own internal team.

After discussing the requirements with ApplTop, it was determined that ASL*Form could easily meet all of the above requirements. Work began by producing the appropriate SQL statement that could extract all the required at its lowest level of detail. Once this was complete, the ASL*Form front-end work could begin.

The structure of the driving SQL statement is basically as follows:

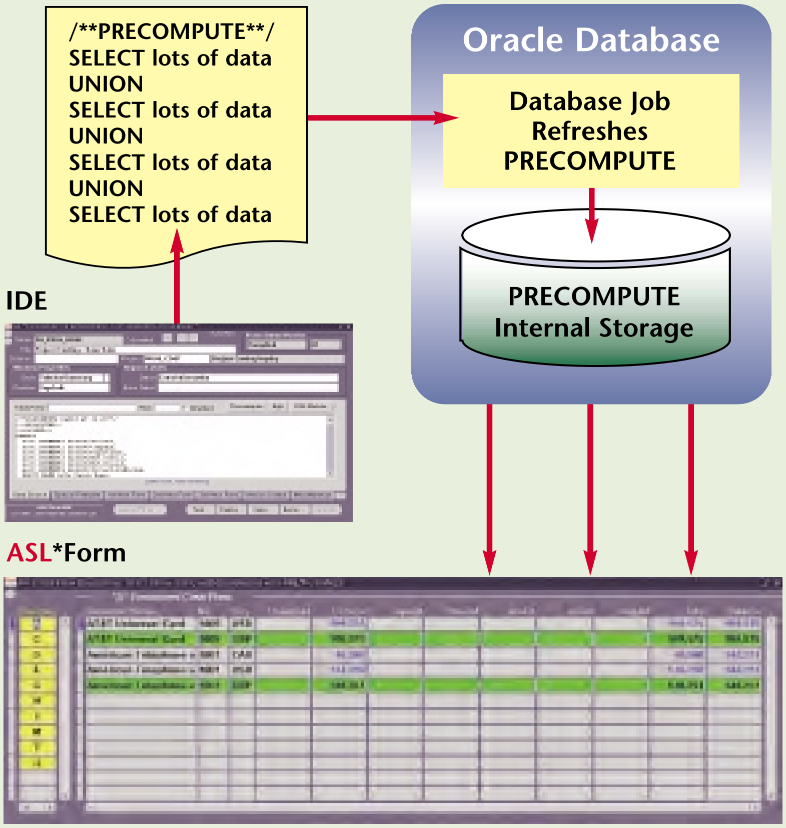
SELECT data from Projects
UNION
SELECT data from General Ledger
UNION
SELECT data from Payables
UNION
SELECT data from Inventory

The actual SQL is a little more complex (about 13 pages worth to be exact). The SQL statement is constructed of four different SELECT statements that pull together all the information needed for the solution, thus effectively removing the need for the additional data imports.

Such a huge and complex SQL statement comes with a few problems:

- Takes a very long time to run (a number of hours); and
- Unwieldy to work with due to its size.

These two issues were promptly solved using ASL*Form. First, the speed issue was solved using ASL*Form's built-in Precompute functionality (Figures 1A and 1B), which enables the results of any SQL statement to be collected at any given time or interval. The size issue was solved automatically by using the



```
/**PRECOMPUTE**/
SELECT lots of data
UNION
SELECT lots of data
UNION
SELECT lots of data
UNION
SELECT lots of data
```

Oracle Database
Database Job Refreshes PRECOMPUTE
PRECOMPUTE Internal Storage
IDE
ASL*Form

Figure 2: Precompute Functionality

Precompute functionality.

Precompute creates a new table based on the results, and once the initial ASL*Form had been created, many of the other ASL*Forms could use the Precomputes table. The Precompute details can be entered directly into the SQL, or the Precompute dialog can be used.

The Precompute functionality works in one of two ways:

- If the SQL statement is relatively simple, then a standard Oracle materialized view is created based on the SQL. This is then refreshed at the specified interval.
- For complex SQL statements, materialized views are not sufficient, so an ApplTop version of the materialized view is implemented. Basically, the results are stored in a table that is truncated and refreshed at the specified time or interval (Figure 2).

A couple of ASL*Form screens were created, which served to Precompute the results of the SQL statement and simplify the rest of the solution. An additional 12 ASL*Form screens were created that take the lowest level of data, now provided by the Precomputes and summarize it at various levels of detail (Figure 3),

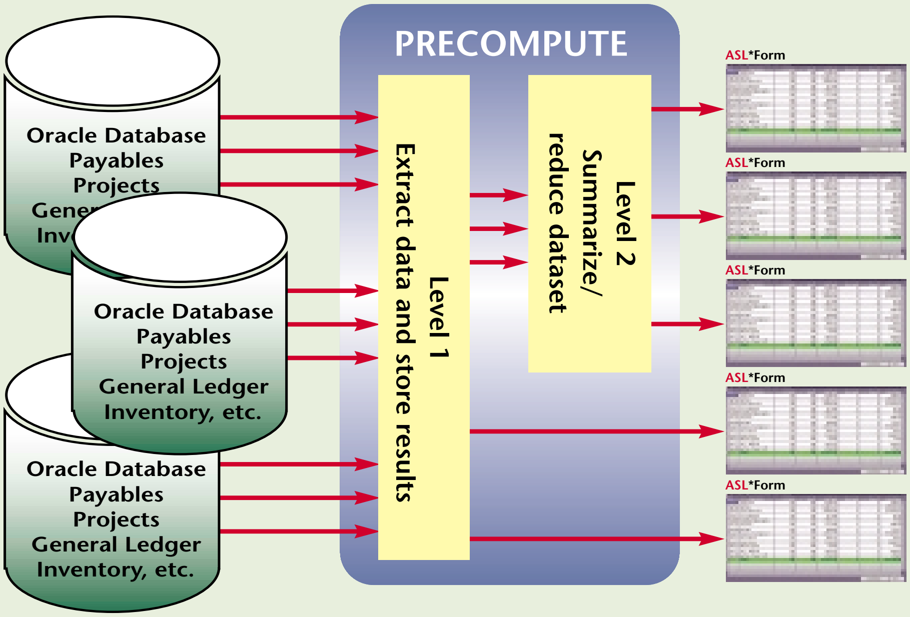


Figure 3: Precomputing the results to improve performance and reduce datasets

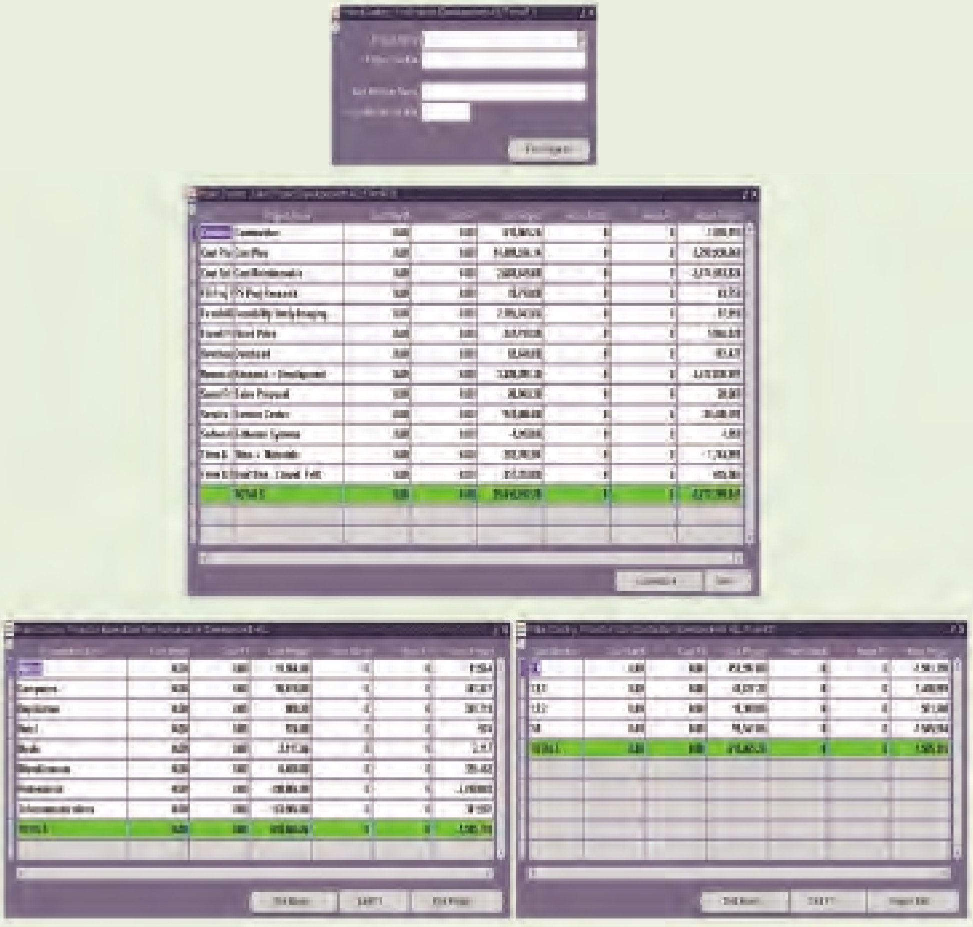


Figure 4: A sample of the Project Costing Solution

giving the user all the required data drill downs (Figure 4).

The Results

The end result provides the user with everything they need, including:

- All the required drill downs are available;
- No data is imported into Oracle Projects;
- An open query across all Projects generally returns results within a second;
- The solution is available directly from the end users menu;
- The solution takes on the native Oracle E-Business Suite look and feel;
- Development costs were minimal;
- Development times were very low; and
- The solution can be supported and further developed by Pace's own internal team.

Had Pace opted for one of the more traditional methods of satisfying their requirement, the front-end application alone would have taken many weeks to complete. Using ASL*Form, by far the most expensive element of the whole project was producing the driving SQL statement. The actual ASL*Form screens themselves, all 12 of them, were created in less than six hours in total; that's from raw data to deployed solution in less a day.

Craig O'Neill is an Oracle Applications technical consultant. Craig has been in the IT industry for 19 years and specifically in the Oracle Applications field for the last eight years. He has worked on numerous Oracle Applications implementations, as well as being the founder of ApplTop Solutions Limited. He can be contacted at craig.oneill@appltop.com.

Contact Details

Company:	ApplTop Solutions Limited
Web:	*appltop.com*
E-mail:	info@appltop.com
Phone/Fax:	+44 (0) 1274 620942