Facing the Challenges of Changing Fiscal Year

Dave Richards, CPIM
Shade Shutters, CMA
Rockford Corporation

Introduction

Though infrequent in practice, there are several reasons that may lead a company to change its fiscal year. Perhaps the greatest challenge facing one of these companies is the burden of reconfiguring its information systems. This paper details the problems encountered by one such company, Rockford Corporation, and the solutions it developed to successfully change the fiscal year in its Oracle database.

Rockford Corporation is an industry leader in the manufacture and distribution of high-end mobile and professional audio components.

Rockford Corporation’s computing environment:

Hardware:
- Main Server: Data General Avion 5800 (3way)
- Memory: 1gb
- Storage: 64gb
- Data Base: 20gb
- Client: PC’s (32mb)

Software:
- UNIX
- Win95
- Network 4.1.1
- NT

Oracle release: 10.7/Prod. 16

Oracle application modules:
- General Ledger
- Accounts Receivable
- Accounts Payable
- Purchasing
- Inventory Control
- Costing
- WIP
- Engineering
- BOM

Planning
Order Entry
Shipping

Other feeder applications:
- Payroll
- Budget (access database)
- Customer call management
- Globalship
- Sales forecasting (access database)

Initial Concerns

Because the financial calendar is one of the fundamental setups in an implementation of Oracle Financials, very few changes can be made to a live system. The apps do not allow changes to the number of periods per year or the beginning and ending dates of periods already defined, without creating a new calendar. Furthermore, even if a new calendar is created the apps will not allow it to be applied to an existing set of books.

Analyzing the Alternatives

After extensive research, Rockford narrowed its possible solution to three viable alternatives.

OPTION 1, CONVERSION TO NEW SET OF BOOKS - Rockford’s first approach was to create a new set of books and then create a new calendar to attach to it. Once set up, historical financial data would be converted to the new set of books. However, the sheer volume of work involved in this alternative made it an unattractive option.

OPTION 2, CHANGE LENGTH OF FUTURE PERIODS - While the number of periods within a year cannot be changed through the apps, Oracle still offers flexibility with the begin and end date of a period. With this ability Rockford would set the end date of its next fiscal year to
December 31 instead of September 30. This would create a fiscal year 15 months long. And since the number of periods would have to remain at 12 per year, the twelfth period of the year would have a begin date of September 1 and an end date of December 31.

In order for this option to succeed, Rockford’s accounting department would have no period to close at the end of September, October and November. Furthermore, the period closed on December 31 would contain four months of data. Rockford determined that this absence of segregation in its accounting data was unacceptable. Rockford would lose the ability to do the year-over-year comparisons that it normally performed with FSG reports.

In a variation of this option, Rockford explored the possibility of creating a fiscal year with a length of only three months. However, this would require 12 periods to be defined over a 3 month time frame. So the periods would not correspond to true months and Rockford’s accounting department would be faced with problems similar to those above. Also it was determined that Rockford could not set up three accounting periods and nine adjusting periods to fill out the short year.

OPTION 3, DIRECT CHANGES TO THE DATABASE THROUGH SQL - This solution required only a few small SQL scripts to change calendar parameters that could not be changed using the apps. The first script changed the end date of the current fiscal year from September 30 to December 31. The second script changed the definition of a year from 12 periods to 15 periods. Once these scripts were run 3 additional months could be set up through the usual period definition form and the year could be closed normally on December 31, after 15 months.

The drawbacks of this method were not only that it was unsupported by Oracle, but that it had also never been used at a live Oracle site as a means to change fiscal years.

After carefully weighing the pros and cons of each option, Rockford chose option 3 as its preferred solution. Though relatively simple in its application several unforeseen issues arose which required additional man-hours to resolve.

Changing the Database

The SQL scripts to perform the change in fiscal year were short and simple and were run as follows:

**Script #1:**

```sql
UPDATE GL_PERIOD_TYPES
SET NUMBER_PER_FISCAL_YEAR = 15
WHERE USER_PERIOD_TYPE = 'Year'
```

**Script #2:**

```sql
UPDATE GL_PERIODS
SET END_DATE = '31-DEC-1997'
WHERE PERIOD_NAME = 'FY-97'
```

The scripts were run three months into FY-97. Once the extra time had been added to the end of the year, three additional months were defined after SEP-97.

At some point during FY-97, Rockford plans to define FY-98 with a begin date of 01-JAN-1998 and an end date of 31-DEC-1998. Rockford will also go ahead and define the first four or five months of FY-98. Then after FY-97 is hard closed, Rockford will reverse the above SQL script #1 and change the definition of a year back to 12 periods. The rest of the 12 months of FY-98 can then be defined normally and FY-98 can be closed without any SQL intervention.

As for external reporting, Rockford would prepare 2 sets of tax returns and financial statements during the 15 month long period. One set for the three month 'stub' period and one set for calendar year 1997.

Minimizing The Risk

In a conference room setting, utilizing a secondary processor, Rockford emulated the business processes associated with the enterprise. The system date was changed to reflect the core spanning period being processed, executed controlled cross-functional transactions for each
respective period, and performed the closing process. This feat started with FY-96 and expanded into the opening of FY-98. Rockford encountered one issue associated with the opening of Fiscal Year 98, the initial period expanded six months. We believe this problem occurred due to the sequence in which we executed the SQL scripts and the opening process. Senior management agreed to move ahead with the change process and will address the problem, post-implementation. The testing process has and will continue until the solution has been fully implemented.

Post-SQL Problems - Retained Earnings

The problems appeared almost immediately. Since Rockford was only three months into its fiscal year, the prior fiscal year, FY-96, was still open to allow posting of audit and other adjustments. However, the last period defined for FY-96 was the twelfth period of the year, SEP-96. And since the definition of a year had been changed to 15 periods, the twelfth period was no longer recognized by Oracle as the final period of the fiscal year. And since SEP-96 was no longer the final period of the fiscal year, closing it would not cause the year-end routine to run which posts the year activity to retained earnings. The result was that regardless of how many adjustments were posted to FY-96, they had no effect on FY-97’s beginning retained earnings balance and Rockford’s balance sheet would be perpetually out of balance by the amount of adjustments posted to FY-96.

Rockford’s first response to this issue was to try to use the Oracle G/L feature of a one-sided journal entry. However, there is no such thing as a true one-sided entry in Oracle. Even though on the journal entry form Rockford only posted one amount to retained earnings, Oracle automatically posted an offset amount to a suspense account.

Rockford’s next attempt at a solution was to change the definition of a calendar back to 12 months. The thinking was that Oracle would once again recognize SEP-96 as the final period of FY-96 and would therefore run the specific year-end close routine that posts to retained earnings. This attempt also failed. Even though the year was redefined as 12 periods, closing SEP-96 had no effect on the next month’s beginning retained earnings balance.

The ultimate solution was again an unsupported use of SQL to make a direct change to the database tables; in this case GL_BALANCES. The following script was used in which code_combination_id 1000 was Rockford’s retained earnings account:

UPDATE
GL_BALANCES
SET
BEGIN_BALANCE_CR = [desired amount]
WHERE
CODE_COMBINATION_ID = 1000
AND ACTUAL_FLAG = 'A'
AND PERIOD_YEAR = 1997
AND CURRENCY_CODE = 'USD'
AND SET_OF_BOOKS_ID = 1

This script changed the balance of every open month of FY-97 to the year’s correct beginning balance. As of the writing of this paper Rockford has encountered no problems from using this method to correct its retained earnings balance.

Post-SQL Problems - FSG Reports

Another problem area after the change was its effect on FSG reports which analyze data from more than one period. These problems arose because of Oracle’s use of period offset numbers in its FSG reports instead of absolute period names.

The first changes had to be made to any FSG report that used a ‘prior year’ column of data to compare to a ‘current year’ column of data. The source for these ‘prior year’ columns were originally defined in Rockford’s reports as 12 periods before the current period under analysis. But since the definition of a year had been changed to 15 months, to compare a current month to the same month in the prior year, the reports had to be changed to look back 15 periods instead of 12. This was true even though the prior fiscal year had only 12 periods defined.

The next problem with FSG reports arose from the fact that four months into the 15 month year, Rockford needed to begin reporting externally as
if it had just started a new fiscal year. But JAN-97 was actually period number four in the Oracle
database. This adversely affected all FSG revenue reports that included year-to-date
columns as those columns would contain data
going back to OCT-96.

The solution to this problem required a creative
use of Oracle's budgeting capabilities. One of
the options for a column on an FSG report is
year-to-date actual minus year-to-date budget.
With this in mind Rockford created a special
budget which contained actuals in the months
OCT-96, NOV-96 and DEC-96 and zeros for the
rest of the 15 month FY-97. Rockford then
changed all of its year-to-date actual columns to
year-to-date actual minus budget. And since the
budget was simply the actuals for the year's first
three periods, the net result on Rockford's FSG
reports was actuals for only JAN-97 onward.

Applications Upgrades

The testing and implementation of Rockford's
selected course of action was performed on
version 10.6 of the applications. Two application
upgrades have since been performed with no
apparent impact on the change of the fiscal year.

Conclusions

As of the writing of this paper Rockford has
successfully solved each of the challenges that
arose from the unsupported changes to its
database.

The project will not be completely finished until
FY-98 has been redefined as a 12 period year
and has been successfully hard closed.

Rockford currently reports to external agencies
and interested parties on its new fiscal year of
January 1 through December 31.

About the Authors

Dave Richards is the vice-president of
information systems at Rockford Corporation
and can be reached by electronic mail at
richarda@rockfordcorp.com.

Shade Shutters is the manager of financial
planning and analysis at Rockford Corporation
and can be reached by electronic mail at
shuttesh@rockfordcorp.com.