

Challenges Implementing Oracle HR/Payroll after Financials

Ken Conway and Bill Stratton
BOSS Corporation

Introduction

Many companies are faced with the situation where they desire to implement Oracle HR/Payroll after they have previously implemented Oracle Financials. This paper will describe how we successfully solved this challenge. Topics addressed include shared data between Oracle Human Resource Management Systems (HRMS) and Financials as well as Critical Implementation Factors and Realistic Project Expectations.

Shared Data between HRMS and Financials

Shared data between Oracle HRMS and Financials presents both functional and technical challenges when attempting to implement Oracle HR/Payroll after Financials. There are often changes required to the Financial modules to accommodate the requirements of HRMS.

Shared Data: Cost Allocation

From an Oracle Payroll perspective, things are fairly easy. Generally, you should map segments of the Payroll Cost Allocation Key Flexfield to the corresponding segments of the Oracle General Ledger (GL) Accounting Key Flexfield. Payroll requirements are not going to require a change to the costing decisions already established by GL. In essence, it is usually easier if GL is implemented before Payroll because the GL team does not need to change the requirements for the Accounting Key Flexfield.

After you have gone live with Oracle HR/Payroll, the GL folks may request a future change to the Accounting Key Flexfield. If this occurs, consider the following steps. Note that some of these steps must be accomplished through the System Administrator responsibility. Do not try the following in your production environment until after this has been thoroughly tested in another database environment.

1. Change Cost Allocation Key Flexfield Value Sets to point to new GL value sets.
2. Ensure that GL Flexfield Map is now only referencing the correct segments based on the new GL requirements.
3. Change the costing values for Oracle HR/Payroll Organizations, Element Links, and Payrolls for every organization and element to ensure that the new GL values are selected.
4. Change Cost Allocation Key Flexfield Qualifiers.
5. Change Cost Allocation Key Flexfield displayable segments if necessary.
6. Confirm that Oracle HR/Payroll Organizations, Element Links appear okay.
7. Run payroll for one employee or small group of employees.
8. Run Costing.
9. Run Transfer to GL.
10. Analyze the GL_INTERFACE table.
11. Have GL team import records to further identify problems.

Above we reviewed shared data with Oracle Payroll. However, when looking at issues affecting Oracle HR, the issues are more challenging. The following areas of shared data must be addressed:

Shared Data: Organizations

Because Oracle Projects and other modules use organizations, you should prepare for additions and likely changes to organizations that are currently set up due to different HRMS security, costing, and reporting requirements. Note that it is not a requirement that HR use the exact same organizations that have been set

up for the other modules. However, it does make good sense when a common organization is being represented.

The top level Business Group organization must be addressed. During most Oracle HRMS implementations, it is recommended that you change your Business Group to something other than the Setup Business Group. However, whenever Oracle Financials has been implemented prior to HR/Payroll, it is common for the Financial Applications to have used the Setup Business Group in its Financial System Parameters. Thus, it becomes necessary for the HRMS team to also use the Setup Business Group. In this case, change the name of the Setup Business Group to the appropriate name of your company.

Note, however, that the Business Group drives the legislation used for Human Resources and Payroll purposes. If your implementation involves employees in multiple countries, then you will likely need to use several different Business Groups, one for each country where your employees exist. This assumes you do not desire to customize multiple legislative requirements into one Business Group.

Shared Data: Locations

If the same locations that were set up for Financials are planned for use by HRMS, you should ensure that each location has an address. Moreover, prepare each address to obtain the appropriate Vertex validation by opening and saving each location from within HRMS. Otherwise, employees cannot be assigned and paid using this location.

In the event that the HRMS project team desires to use totally separate locations than are being used by Oracle Financials, we recommend establishing a naming convention for locations. This will be necessary so that the HRMS users will not inadvertently select a non-HR location for an employee's assignment. The naming convention can be as simple as starting every HRMS created location name with "HR-".

Shared Data: Jobs/Positions

Oracle HR nearly always requires changes to jobs and positions that have been set up for use in Oracle Financials. This is because of management and government reporting requirements. In particular, implementers of Oracle Purchasing require either jobs or positions in order to identify buyers. However, if a position has been set up with one segment titled "Buyer", then the HR department will almost assuredly require this to be changed.

The solution to this problem is less technical and more functional. While the HR department may have some flexibility on its management reporting requirements, they will not be able to be as flexible when addressing the government reporting requirements. The functional issues involve gaining acceptance of new job and position names by the users of the other Oracle modules.

Shared Data: People/Assignments

Some data within the underlying tables for people and assignment might need to be altered, depending on exactly how your data has been stored by Oracle Financials. These are usually more minor issues when compared to the other items of shared data. For instance, the last name and first name of employees might have been entered in uppercase. Because this is not as desirable for HR reports or payroll checks, you might want to change items such as this. It is generally possible for HR to reuse the employee information that has already been set up by the Financials users after the data undergoes a minor massage.

In an extreme case, it might be necessary for the users to totally re-enter employees within HR either manually or through conversion using the HR APIs (application program interfaces). If this is necessary, then the Financials users would also require new set up so that vendors and buyers are reestablished to point to the new employee records.

Shared Data: Addresses

Employee addresses set up for use by Oracle AP require attention because AP does not establish the appropriate Vertex validations required by Payroll. You will likely need to write SQL scripts or prepare to delete and reenter certain addresses.

Shared Concurrent Manager Processing

Before bringing your new payroll system live on Oracle Payroll, prepare for detailed system testing which includes both Financials and HR/Payroll. This type of testing is not merely a review of the processing times of the Oracle Payroll system. The focus in this section is on the competing processing requirements for the Concurrent Manager. The Oracle Payroll calculation engine can be quite system intensive. You must understand your system processes and reports to know how other Financial users will be affected when payroll is running. Conversely, you will want to understand how the payroll system is affected based on the types of processes being initiated by Financial users.

Additionally, you will want to review the typical financial processes such as month end close. How are these financial processes affected by the Oracle HR and Payroll reports and processes? Prepare to research the various options within the Oracle Concurrent Manager. You can establish Concurrent Manager calendars where certain users or certain processes are given higher priorities in the waiting queues of the Concurrent Manager. On a practical note, you might want to establish a business procedure to run certain payroll and financial processes at night.

Critical Implementation Factors

Thus far, this paper has focused on the technical issues involving shared data between Oracle Financials and HR/Payroll. However, the greatest challenges in obtaining a successful Oracle HR/Payroll implementation go well beyond the issues of shared data. Below are a number of critical implementation factors that should be addressed. We believe these factors are important for any Oracle HRMS implementation regardless of whether Oracle Financials was implemented first. However, we recommend that any client who has already brought Oracle Financials live be aware of the following. Human Resources and Payroll systems are quite different from Financial systems. Your success implementing a Financial system might not translate into a successful HR/Payroll implementation.

Critical HRMS Implementation Factors: Initial Phase

A detailed requirements analysis must be performed along with a high-level gaps analysis. This must occur early in the project to obtain good estimates of the expected project duration. This issue is closely tied to the next critical implementation factor of resources in that at least one experienced Oracle HR/Payroll resource should be involved. A solid project plan that identifies expected workload and assigns client resources to as many of the tasks that are reasonable should result from the requirements and gaps analyses. If this analysis is neglected, unpleasant surprises almost certainly will arise later in the project. The time to implement an HR/Payroll implementation (regardless of whether it is Oracle, SAP, or PeopleSoft) is often underestimated by a client.

Critical HRMS Implementation Factors: Resources

It is important to have a mix of functional and technical consultants experienced with the Oracle modules as well as functional and technical personnel from the client. There is no perfect number of consultants. The scope of the project, number of employees, complexity of the client's business, time allotted before go-live date, and knowledge base of the users all factor into how many project-dedicated personnel are required.

Interaction between technical and functional project team members is necessary for project success. Although it might be important to have separate technical and functional teams, there should not be a permanent dividing line between these teams. Having good communication and cooperation between the technical and the functional personnel increases the chances of having a successful project and producing good results for the client.

Functional Consultants: The Oracle HRMS product is very flexible and highly configurable. Because of the greater the flexibility of a product, the greater the number of decisions that must be made during the project. It is recommended that functional consultants in both Human Resources and Payroll be used, although this expertise might be achieved with one person. At least one consultant should have experience with the Oracle HR/Payroll products themselves. Again, this product is highly configurable. Experience with Human Resource and Payroll systems in general is not sufficient. The user community might have to live with set up decisions for years to come. The number of decision points is high. The cost of wrong, or less-than-optimal decisions is high.

Technical Consultants: The Oracle HRMS database is somewhat complex. Two mistakes commonly made on projects are (1) assuming that experienced Oracle Financials technical consultants will easily translate into effective Oracle HR/Payroll technical consultants and (2) assuming that the client IT staff can handle these tasks without outside assistance. These issues are not about aptitude or potential. The issue is about experience.

An Oracle Financials consultant can be trained on Oracle HR/Payroll and certainly become effective. The mistake is to assume that the learning curve for the Oracle HR/Payroll database is the same as that of Oracle General Ledger. The result of this mistake is to significantly increase project duration, resulting in higher project costs possibly causing important deadlines to be missed.

The client IT staff must be represented and participate on the project. However, be careful not to assume that this staff will quickly master the Oracle HR/Payroll database. They will rely heavily on the experience of technical Oracle HR/Payroll consultants. If the learning curve for an experienced Oracle Financials consultant (who might already know the Oracle Tools of Forms and Reports) is high, the learning curve for IT staff not previously exposed to Oracle will be higher.

It truly is a matter of spending the time and money to bring in experienced resources to help do the job right. The alternative is the increased price for bad, hastily made, or incomplete decisions.

Finally, the emphasis here has been on bringing in experienced functional and technical leadership. However, this leadership should not work in a vacuum. The user must be able to maintain the system after it goes live. Thus, it is imperative that clients have fulltime functional and technical staff to work on the project. This staff must become the experts so that dependence on consultants is not necessary for a long-term basis after the new systems go into production.

Critical HRMS Implementation Factors: Executive Sponsorship

A vital component of the HR/Payroll implementation is the involvement and support of upper management. It is important for the team personnel to be able to rely on the support of management. The management team must be available to answer questions, make decisions, and communicate to other client management on the progress and impediments of the project.

This can be vitally important when the user community has used the same HR/Payroll systems for many years. Old habits die hard, and some users might refuse to assume tasks that they consider to not be their jobs. The new Oracle HRMS system brings much opportunity but also introduces potential frustrations as employees' roles might need to change. Upper management must remain involved to make decisions that affect employees' future responsibilities.

Part of this support by management is financial. Management needs to stay informed on where the project stands in relation to project time-lines and allotted budget. The best consultants can resolve almost any

business requirement. However, time and money add up. Management must be prepared to step in and make decisions and change business requirements or identify requirements of lower priority that can be solved in a future phase of implementation.

Critical HRMS Implementation Factors: Documentation

The user should keep thorough, accurate, and complete documents of processes, procedures, and decisions made during the project. If done correctly, this documentation can lead to enhancements and further refinement by client personnel (not consultants) after the project has completed and the new HR/Payroll system is live. Training manuals should be developed and constantly updated as testing is performed and changes are made to the system. Documents should be developed for the following categories:

- User documentation
- Solution design
- Technical specifications
- Training

It can be frustrating and frightening for a user to need to make changes to the system without sufficient documentation. The user might not fully understand why the system was set up in its' current state. Thus, project documentation can provide not only the understanding of why the system was set up in the way that it was, but also what steps were necessary to provide the solution.

Documentation must be clear, concise, and accurate. The client during the project implementation should review it. Proper documentation provides the roadmap to make new system decisions. Sticky notes posted on a wall do not stand the test of time.

Critical HRMS Implementation Factors: Parallel Testing

After rigorous system testing has occurred, parallel testing should occur on Oracle Payroll implementations. First, we will define parallel testing. Parallel testing involves duplicating operations between the legacy system and the Oracle Payroll system. However, duplicate operations do not imply duplicate results. Duplicate results are certainly the goal. The parallel test is necessary to identify test results that do not match the results produced by the legacy system. Also, the parallel test is necessary to provide hands-on training for the actual users who will operate the eventual production system.

If sufficient payroll testing or conference room pilot testing has already occurred, parallel testing would not be necessary. This is true, but the keyword is sufficient. Some items can slip through rigorous payroll testing.

For example, a data conversion for tax information might have been tested for many employees. However, you will never really know whether the correct federal, state, and local taxes are being withheld until every employee has been tested. Various states have reciprocity laws for employees who work in one state but live in another. Because of these many combinations, parallel testing ensures that every combination has been met. Moreover, the tax information calculated during the parallel test might not match because Vertex calculates differently from the legacy system. Usually, the discrepancies are a result of incorrect legacy system calculations. However, nothing should be taken for granted.

The parallel test should be performed primarily by the actual end users who will be operating the future production system. This gives the users hands-on, real-world scenario training to reinforce any user training classes that have been conducted. Certainly, this effort causes a burden on client staff because they are being asked to perform two jobs at once.

Some users simply do not have the manpower to handle data entry in two systems to run a true parallel of the one system and the new Oracle system. A good alternative to this is to use some kind of auditing tool. There are several tools of this kind on the market. The Implementation Team Leader would designate a Test Coordinator for each day that testing is being conducted. Having a Test Coordinator designated by

day provides greater flexibility of scheduling, particularly if work is being done on more than a regular eight-hour day and through weekend and holiday time.

The role of the Test Coordinator is to prepare the testing document, hand out assignments to the testing team, assist individual team members with assignments, and log the results of the testing. The Test Coordinator gives problems identified by category to individuals knowledgeable in the area to resolve the problems. After a designated number of problems are found on one issue, no further testing on that function is done until the problem or problems are resolved. This eliminates valuable testing time being wasted on an already known problem. At the end of the testing period, the Test Coordinator documents the results of the test items and presents the findings to the Implementation Team Leader. It is recommended to have Subject Matter Experts from the various areas (Benefits, Compensation, Payroll, and so on) on-hand to assist as members of the testing team and to answer questions regarding their areas of expertise. Support from the Technical function and DBAs is crucial.

How long should you parallel test? This answer varies from project to project. As a general rule, consider three full parallel runs over a three-month period. Essentially, you spend the first couple of weeks of each month performing the parallel test and the second couple of weeks of each month resolving problems and issues from the previous parallel test and preparing the system for the next parallel.

How should you structure and prepare for each parallel test? First, your DBA must establish a clean test database environment. Then, the DBA must prepare this environment to mimic a specific point in time that will correspond to the payroll period to be tested. The pay period to be paralleled does not have to be the current pay period. It can be a period back in time; however, this requires advance planning to ensure that the DBA has the appropriate production system backups and the client has the correct legacy system data backups prior to the desired payroll period dates. This is accomplished differently for different projects.

Some clients might already have a production Oracle system with Financials live. These same clients might have already converted some of their HR data into the Oracle production system prior to the official go-live date for HR/Payroll. In this situation, the DBA should perform a complete backup of code and data from the production system and refresh this information onto the test environment as of a specific point in time. Then, any other data conversions or migrations that are planned for the future HR/Payroll system must be performed. For example, the Oracle production system might already have employees, addresses, and assignments loaded in addition to the existence of other Oracle Application modules. After this information is refreshed into the test environment, other HR/Payroll scripts (or data entry) must occur (for example: Personal Payment Methods, Salary, and Element Entries). If a mid-year conversion is planned, the Balance Initialization routines must be initiated.

The Balance Initialization process should be elaborated on. On the first parallel test, the balance initialization might not provide as much value as it will on the future parallel tests. The first parallel test is likely to have many test results that are inaccurate. One small mistake can affect the results for numerous employees. These mistakes are often a result of user errors. Remember this is a training exercise for the actual end users and provides an excellent learning experience for users without the risk of error to a real paycheck.

Other mistakes that can affect numerous employees involve data conversions. For example, one data conversion routine for benefits might have *operated* correctly to load employee bonuses; however, the correct operation might have been using an incorrect source data file for bonuses from another pay period. Another example could be a data conversion routine that incorrectly loaded pretax deductions for only one element. This one element causes the taxes and net pay to be incorrect for a large population of employees.

Thus, having correct balances prior to the first parallel test might not have improved the test results significantly. However, the Balance Initialization should seriously be considered by the time of the second parallel test.

After the parallel test has completed and after the project team has corrected programs, procedures, and so on, you are ready for the next parallel. It might be unlikely that you can reuse the same test environment.

Thus, the DBA must again refresh this environment to mimic the specific point in time that will correspond to the payroll period to be tested. As you approach the second and third parallel test, you might have the opportunity to attempt back-to-back payroll period parallels without a refresh. This has an added advantage by allowing you to test month-to-date balances that are sometimes required in interface programs to third-party administrators.

If you want confidence when the new Oracle HR/Payroll system is ready to go live, you should have it after conducting parallel tests.

Critical HRMS Implementation Factors: Celebrating Successes

An HR/Payroll implementation takes a significant amount of time to complete. The stress level will get high and it is vital to have a cohesive, productively functioning team. The difficulty is that there is only one finish line. The finish line is to successfully go live. However, this might be many months or even more than one year away. The team must recognize accomplishments along the way. There is no magical formula, but there are numerous small and large things that can be done to help the team feel motivated to continue their drive for success.

Common Unrealistic Project Expectations

One of the most common responses observed after a company purchase of any new software package is the expectation that the new system will finally do everything they always wanted. It will also do all these tasks faster, simpler, and automatically. One of the hardest tasks a project manager must accomplish is to properly manage these expectations. This section will outline some of the more common surprises that can surface while implementing Oracle HR/Payroll.

Unrealistic: HR/Payroll Is Just Another Oracle Application

This is generally the initial reaction of companies who have already implemented Oracle Financials and are preparing to implement HR/Payroll. They expect that the knowledge they have gained from the other applications will automatically transfer to the HR/Payroll modules. They also expect that any external resources they used for implementing the Financial modules will be equally experienced in implementing HR/Payroll.

You will quickly discover that the only areas that remain consistent in HR/Payroll are in constructing Key Flexfields and defining responsibilities and task flows. From there, the module takes on a life of its own. A completely different development group located in the United Kingdom developed the HR/Payroll module. Some of their unique terms show up every now and then, such as cheque and spinal points. The underlying HR/Payroll database table structures are very complex. At some implementations, we have developed SQL scripts to retrieve payroll results that required SQL Joins to 22 different tables.

Some companies plan to modify Oracle Payroll's Check Writer to conform to their own payroll check design. This is usually because they have been successful in completing this same task on their AP check form and do not expect any problems. Oracle Accounts Payable provides hooks in their check writing process that enable companies to insert the necessary escape sequences for the MICR coding. These same hooks do not appear in the Payroll Check Writer process. A simple task in AP can become a major modification in Payroll.

As stated before, successful Oracle Financials implementers do not automatically make successful Oracle HR/Payroll implementers. Many projects have missed implementation dates and exceeded budgets because the learning curve for HR/Payroll was grossly underestimated.

Unrealistic: Vertex Will Handle All My Taxing Issues

One of the key selling points of Oracle HR/Payroll is its tight integration with Vertex. Vertex refers to payroll tax engine product produced by Vertex Corporation that is bundled with Oracle Payroll. Indeed,

there is minimal setup required to have the proper taxes withheld in a normal payroll run. Vertex automatically recognizes the proper taxes to withhold based on the zip code of the employee's primary address and work location. Vertex also recognizes all full and partial reciprocity laws between states when employees work in one state and live in another.

Initially, this is a blessing to many payroll departments because they will no longer have to keep up with all the changing tax laws in all their jurisdictions. Some companies are surprised to discover that their legacy systems were not withholding all the proper taxes from their employees. Generally, this occurs at the local and school district levels. The initial reaction is to turn the tax off and continue to tax in the same manner as the legacy system. Until recently, the only way to turn the tax off was to navigate to the W4 window of all employees who have this tax and mark them exempt. This solution required careful monitoring of every payroll run because there is no easy way to identify these employees beforehand. A patch now enables companies to explicitly identify which local taxes to take at the GRE level. Before utilizing this solution, companies should research every tax body to determine whether they are required to withhold this tax.

The next area of frustration with the Vertex integration occurs when a tax has been withheld in error. This generally happens when a zip code was entered incorrectly and the problem was not discovered until the employee received her check. Oracle does not provide an easy method to return this amount back to the employee. You must make a balance adjustment to remove the withheld balance, as well as the Subject To tax balance. These adjustments can be complex and might cause problems later during the quarterly and year-end closes. The actual money is usually refunded to the employee through either an AP check or petty cash.

Unrealistic: No More Manual Wage Attachment Calculations

Oracle HRMS provides an easy set up process to establish involuntary deduction elements for all the different wage attachments. The federal and state laws concerning wage attachments are complex, and Oracle HR/Payroll is designed to support all these laws. Once again, this is a blessing to the payroll department due to the complexity of these laws. However, Oracle does not provide any backup to support the calculation of a wage attachment. You have no idea what rule has applied when the amount is deducted from an employee's check. This can prove to be embarrassing and frustrating when an employee inquires about the calculation of the amount.

There are many stories where companies complete a parallel run and discover that the child-support calculations do not agree. After a detailed and time-consuming investigation, it has been discovered that court orders have been issued in violation of the law and that the judges needed to reissue new orders. The point here is that although Oracle was correct in applying the federal and state laws, it provided no backup to assist the payroll department in researching the issue.

Because of the ever-changing tax laws, there are many associated patches for wage attachments. As you go through your testing phase, do not assume that the Oracle calculation is always correct. You might need to contact Oracle Worldwide Support to discover that a missing patch is causing the calculations to be incorrect.

Unrealistic: It Is Straightforward To Update the FIT Taxable Balance

One of the first tasks in preparing an implementation plan is to decide on a target implementation date. The easiest date to begin any new payroll system is the first paycheck of a new calendar year. All year-to-date totals start at zero on January 1, and there is no need to load balances from your legacy system. However, this is generally the busiest time of the year for a payroll department. Also, many companies want to phase different locations into the new system throughout the year. For many reasons, some companies have to transfer payroll balances from the legacy payroll system into Oracle Payroll.

Many companies assume that this will be a straightforward and simple task. They quickly discover, however, that their legacy tax balances do not map easily to the many components of the tax balances in

Oracle. The FIT Taxable balance within Oracle Payroll is a good example of this challenge. To calculate the FIT Taxable balance, you must load in the following balances:

- Gross Earnings
- Supplemental Earnings for FIT
- Supplemental Earnings for NWFIT
- Def Comp 401K
- Section 125

Clearly, there is not a one-to-one correspondence between the balances in the legacy system and in Oracle Payroll. Do not underestimate the time it will take to prepare the numbers that Oracle requires for a balance load. We recommend that you simulate a quarterly and year-end close during your testing of the balance loads. Many mistakes that are made during the balance load do not resurface until you attempt to create the necessary quarterly and year-end files and reports. Corrections to these balances can be complicated and time consuming.

Unrealistic: When We Are on Oracle, Payroll Should Run Much Faster

This is a very delicate issue and there are no universal answers or predictions on how long a payroll will run in Oracle. One of the most powerful features of Oracle Payroll is that balances are not explicitly stored. Whenever a balance is referenced, Oracle dynamically **calculates** the balance by adding up all payroll run results for the requested period. This enables you to query on a year to date based on any date in the past through the power of Date Tracking. Although this is a powerful feature not found in many payroll systems, it can have a negative effect on the processing times.

Because the early implementers of Oracle Payroll were unsatisfied about the processing times, Oracle instituted a process of storing the latest balances in tables to improve performance. This has improved processing times, but other performance-tuning might be necessary. One of the reasons we previously stressed the importance of performing complete parallel runs is to discover whether there will be any performance issues.

Some performance tips to try if you experience unsatisfactory run times include the following:

- Summarize timecards. Instead of processing 5 timecards at 8 hours each, summarize this to one time card of 40 hours.
- Experiment with the THREADS parameter of the PAY_ACTION_PARAMETERS table. The general rule of thumb is two threads per CPU.
- Any balances that are referenced in a Fast Formula are updated in the latest balances table. If you suspect that your processing times are spent calculating a particular balance, place a reference to this balance in a Fast Formula that will execute for every employee.

These are just a few of the more common techniques for improving performance. You might need to work closely with you DBA and MIS staff for any hardware performance solutions.

Unrealistic: Once Oracle Is Implemented, We Will Not Have to Rely on MIS Support Anymore

This comment comes from both the user and MIS departments. Most legacy systems require heavy MIS involvement to keep the system in compliance with tax laws. There is usually a backlog of enhancement requests from Payroll and Human Resources as well. Many companies elect to purchase these systems to lessen the burden on in-house resources.

During the marketing phase, Oracle stresses the flexibility of Fast Formulas and the availability of easy-to-use reporting tools. Unfortunately, this is often interpreted to mean that an end user will be able to write her own reports or modify a Fast Formula. Granted, the tools being referenced are very easy to use. The

problem centers on the complex structure of the Oracle HR/Payroll table design and detailed programming logic incorporated in the Fast Formulas.

As mentioned previously, a simple SQL query to retrieve a list of all employees with a United Way deduction might result in a complicated table join. At some implementations, we have observed complex Oracle Payroll joins through SQL requiring access to 22 tables. Date Tracking also complicates simple queries because there can be multiple records in a table for the same employee. Clearly, preparing reports from payroll results require some extremely intense knowledge of table structures and SQL coding techniques.

The entire gross to net calculation in Oracle Payroll is accomplished with the use of Fast Formulas. One of the strongest assets of Oracle Payroll is that you can control this process by modifying or writing your own Fast Formulas. However, one look at an existing formula reveals that the everyday end user lacks the necessary skills needed to properly change a calculation.

MIS involvement during the implementation process is essential to the success of the project. This is especially important if you are using external consultants. MIS must be able to step in and support the system when it is time to let the consultants go. It is unwise to allow all of the knowledge gained from implementing your system to leave with the consultants when they walk out your door.

These are some of the recurring issues when implementing Oracle HR/Payroll. This section is not intended to scare you. Its purpose is to keep you from making some of the more common incorrect assumptions while you implement your system. Careful planning and constant testing yield a powerful and important tool in achieving the goals stated in your enterprise's mission statement.

Conclusions

There are certainly challenges in implementing Oracle HR/Payroll after the previous implementation of Oracle Financials. Many of these challenges can be solved by clear and compromising communication between the users of the different project teams. The technical challenges regarding shared data between the Oracle modules is overcome without too much difficulty if you have access to someone who has been through this type implementation before. Overall, we believe the biggest challenge in the HRMS implementation is not the fact that data is shared with Financials. We believe the greatest challenges involve the flexibility and complexity of the Oracle Payroll module itself. To greatly increase your chances for a successful implementation and to reduce your long-term budget for both the implementation and for future maintenance, it is highly recommended that you gain access to skilled resources with direct, hands-on experience on the Oracle HR and Payroll modules.

About the Authors

Ken Conway and Bill Stratton are consultants who are exclusively dedicated to Oracle HRMS (HR, Payroll, Advanced Benefits, Time Management, and Training Administration) implementations for BOSS Corporation (Better Organization Service Solutions) (www.bosscorporation.com). Ken and Bill have contributed in both a management and hands-on role in the implementation of numerous Oracle HR/Payroll projects including several of the first successful implementations that went into production in the United States. They have been involved in the Oracle BETA program during the early releases of the Oracle HR/Payroll products. As members of the BOSS Corporation HR/Payroll Systems Division, Ken and Bill join an experienced team, which provides consulting services to clients throughout North America.