



NLR-TP-2001-019

Oracle HRMS and External Payrolls: The missing link

A. Kassies and T. Vergult



NLR-TP-2001-019

Oracle HRMS and External Payrolls: The missing link

A. Kassies and T. Vergult

This report is based on a presentation held at the OAUG Europe 2001 Conference, Geneva, Switzerland, 20-23 March 2001.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.

Division:	Information and Communication Technology
Issued:	January 2001
Classification of title:	Unclassified



Summary

The Oracle Human Resources (HR) module offers a powerful means to define gross pay elements. Due to the complicated structure of the local Dutch taxes and deductions, and the lack of an existing Oracle Payroll for the Netherlands, we are virtually forced to use an external payroll system. In order to combine the advantages of both worlds an interface between Oracle HR and the external payroll system has been developed.

A generic concept has been used for the design of the interface. All pay-relevant data of an employee (including e.g. name and address) are mapped onto Oracle HR elements using the standard HR batch mechanism. These batches are run in each pay period in order to reflect the changes in employee data in the HR elements. To enable retroactive changes some elements (e.g. the regular salary) are generated for each pay period in the current fiscal year. Other kinds of benefits or compensations are maintained directly within the HR elements.

The definition of each HR element that needs to be interfaced contains the external identification of the pay element (in a descriptive flex-field). For each pay period the interface keeps track of the changes in the HR elements, using a mechanism that we call "PeriodTrack". Please note that the HR elements can be "Updated" or "Corrected", so it is not sufficient to rely on the date-track mechanism for all changes. Given the date-tracked state of all the pay elements in each pay period, mutations between two successive periods can be generated. These mutations are stored in a table and may be viewed using an Oracle Discoverer workbook. Finally, the generated mutations are exported to a file with a dedicated format for import in the external payroll system.

Using the methods described above we were able to use Oracle HR for the definition and maintenance of all gross pay elements, while still using an external payroll system. The mutation-driven interface itself contains almost only generic code (except for the specific format of the export file). The definition of the HR elements reflects of course our local situation.



Contents

1	Introduction	4
2	Payroll data in Oracle HRMS	5
3	Oracle HR elements for all the gross payroll details	7
4	Design of the interface to an external payroll system	8
5	Interface processing	10
6	Conclusions	11
	About the Presenters	12

5 Figures

(12 pages in total)



1 Introduction

The National Aerospace Laboratory NLR is an independent non-profit research institute based in the Netherlands that performs contract research for national and international customers. NLR independently renders services to government departments and international agencies, aerospace industries and aircraft and spacecraft operators. Customers include various organizations based in the Netherlands, in Europe and elsewhere. NLR is one of the Large Technological Institutes of the Netherlands. NLR's staff of about 950 currently includes about 450 university graduates, and about 300 graduates from advanced technical colleges. About 75 per cent of NLR's activities consist of research and development under contract. For more information visit our website at: www.nlr.nl.

Oracle Applications has been introduced at NLR in 1999. Besides the Human Resource module the following modules are used: Financials (GL, AR, AP, en FA), Purchasing (PO), Inventory (INV), and Projects (PA). 50 up to 100 core users are using the system on a daily basis. NLR is a project-oriented organization, which is reflected by the nearly 3000 active projects. All working hours, use of facilities, and other costs are accounted with the Projects module. More than 600 employees have so-called 'Requisition input' and 'Projects browse' responsibilities. Oracle Discoverer is used for reporting purposes with dedicated business areas and workbooks.

At the start in 1999 the Human Resource module was only used for the definition of employees, organization units, and other required data for the other Oracle Application modules. Since the last quarter of the year 2000 the Human Resources module is used independently for the registration of all employee data, including the gross pay elements. The setup of this module, and the design of dedicated extensions, has been realized along with consultants of Ernst&Young and Arinso International.

ARINSO International is a group of ICT (Information and Communication Technology) consulting firms that focus on ERM (Employee Relationship Management) and provides HR solutions for some of the world's largest employers. This includes implementations of ERP systems like SAP, Meta4 and Oracle, design of e-HR™ solutions, as well as Business Process Outsourcing and ASP services. ARINSO International is a strategic Oracle Apps HRMS partner that currently counts more than 50 experienced Oracle HRMS consultants worldwide. For more information visit our website at www.arinso.com.

This paper describes the payroll interface between Oracle HRMS and an external payroll calculation engine. However we will only discuss the data flow from Oracle HR to the external payroll system, not the data flow into the opposite direction which includes the interface from



the payroll system to Oracle Finance to integrate the accumulated results by cost. Using the standard import possibilities within Oracle Finance, this type of interface can be developed. An upload of all the net results back into Oracle HRMS might as well be part of an implementation and can be realized with the recently released functionality of backfeed tables.

This paper will give further details on the general setup of payroll data in the Human Resource module and more specifically on the use of HR elements needed for gross pay. In a second part the design of the interface to the external payroll system will be described extensively, including the use of the interface procedures. Finally we will come to some conclusions on this business case.

2 Payroll data in Oracle HRMS

This paragraph describes how we can administer all gross payroll related data within Oracle HR, needed by an external payroll system. This part is not intended to be complete or exhaustive, because most of the relevant information can be found in the existing documentation sets, but serves as an overview supplemented with some general remarks.

The simplest example of data needed by a net payroll system is personal data, such as name and address (although not used by the payroll as calculation input, but rather for information or reporting needs). For personal data that does not exist in Oracle HR but is necessary for the payroll system, descriptive flexfields and Special Information types can be defined.

External payroll systems also need assignment-related data, both for information (function name) and calculation (full time/part time) purposes. The latter will also require the starting date to enable the payroll engine to keep track of payroll history (Datetrack) and to enable retroactivity if supported. Also costing information can be stored at assignment or organizational level, as defined and qualified by the Cost Allocation Key Flexfield. Furthermore an employee is linked to a particular payroll in the assignment screen. If specified, this will determine to which payroll system the data needs to be transferred.

Bank account numbers are also primary data used by several payroll processes. Oracle HRMS therefore provides Personal Payment Methods that are based on Organizational Payment Methods. Unfortunately, those standard screens cannot be used if your business group doesn't belong to one of the standard 'localized' legislation (US, UK, etc). Error messages will pop up if you decide to use them anyway, because data is missing in some low-level (payroll) database tables. You could get around it by using flexstructures to rebuild the needed functionality of



bank account numbers or by developing a similar customization. As an alternative you could identify and add the minimal missing data, in order to activate the existing screens.

Salary is another obvious example of data stored in and managed by Oracle HR that is necessary for payroll processing. Depending on the kind of organization, you either choose the grades and scales functionality, or the Salary Administration screen, or both. Along with scales and spinal points the grade keyflexfield has proven to be very flexible and has offered a perfect fit in most implementations. It is advisable to test several alternatives to find out the definitive setup for this complicated set of possibilities. Salary Administration (extended with extra functionality in recent releases) provides the HR department with workflow 'avant la lettre'.

Most payroll-related data is stored as elements and input values within Oracle HR. Element classifications are needed to define an element type. The list of values of those classifications will appear to be empty in an environment of a non-standard legislation code. The only possibility to populate this list is to execute an SQL-script that inserts data in the appropriate tables. The result is a powerful set of screens to define the exact behavior of each element. The configuration is executed on three levels. First of all you define the nature of the element type. You specify if it is recurring or non-recurring, if multiple entries within the same payroll period are allowed, etc. Some fields are only relevant when implementing Oracle payroll. Next, you define the validation of each input value of an element type (up to fifteen). You can control the data type of the input value plus minimum or maximum values. You can also associate a list of values (quick code) with the input value, or use a Fast Formula for validation in more complex cases. Do not forget to pick character as data type if you want to use quick codes containing numbers. The final step is to link your element type to assignment conditions or to define eligibility rules. The validation of every input value can be narrowed down on this link level.

Once the steps above are executed, you can enter element entries on the assignment level. In case where a lot of element entries have to be inputted for several employees, the Batch Element Entries (BEE) can be used. The Bee functionality includes screens for defining the behavior of a batch, dynamic screens for quick data entry and processes (Mix concurrent programs) to validate, upload, purge or rollback the data. Also a screen to see all the erroneous data together with self-explaining error messages is available. The user can also generate automatically lines for a batch, based on assignment sets. This functionality shows however to be a challenge for your support analyst.

Oracle HR proves to be a suitable system that can be configured and extended to manage all the data needed by an external payroll system to execute a complete payroll calculation.



3 Oracle HR elements for all the gross payroll details

Previous implementations of payroll interfaces from Oracle HRMS learned that selecting and formatting personal and assignment data takes 70 to 80% of the development effort. The rest of the time was spent on the data stored as element entries. In contrast with the previous statement, the largest amount of data to be transferred to the external payroll is stored precisely in the element entry tables.

The disproportion of the development effort against the small quantity triggered the idea to transform all the non-element data, needed by the payroll, into elements. Soon it became clear that the standard functionality of the Batch Element Entries fits perfectly to perform this transfer. The final result is thus that all the gross payroll relevant data become element entries.

A simple example explains the concept. The external payroll engine needs amongst other things the first and last name, title and date of birth of an employee who is linked to this payroll. A new element type with four input values (first name, last name, title and date of birth) is defined to hold this information and a batch header is created. Next, a custom program is developed in order to populate the batch with these four personal fields. Before the payroll interface transfers every month all the element data to the external payroll, the custom program fills the batch. Then the batch is validated and transferred to the element mentioned (figure 1).

This concept has plenty advantages. First of all, a lot of seeded Applications functionality is used to build and configure the payroll interface. Batch Element Entry processes are used to convert the data, by which the error-reporting screens can be used. The BEE processes are also concurrent programs, offering the salary administrator the possibility to schedule the validation and upload of the batches, or to bundle all the pre-payroll processes together into one request set. Also the validation

The screenshot shows a window titled "Entry Values (Vergult, Mr. Thierry)". The window contains a menu bar with "Action", "Edit", "Query", "Go", "Folder", "Special", and "Help". Below the menu bar, the form is titled "Entry Value". It has several input fields: "First Name" with the value "Thierry", "Last Name" with "Vergult", "Title" with "Mr.", and "Date of Birth" with "21-JAN-1970". There are several empty input fields below these. At the bottom, there are fields for "Date Earned", "Payee Details", and "Processing Priority" with the value "500".

Figure 1: Example of entry values



possibilities of element types and input values makes it possible to define extra validation rules for the non-element data when converting them to elements.

Building the interface consists now of two parts. First, scripts have to be developed to generate all the necessary batch lines. Secondly, payroll interface specific procedures have to be developed. This task becomes less complicated because all the data resides only in the element entry database tables. The latter is explained in more detail further on.

A consequence of this payroll interface concept is that the non-element payroll data is duplicate within the Oracle HR tables. This redundancy results in some extra space requirements.

4 Design of the interface to an external payroll system

At first the different kinds of elements are discussed. The pay elements in Oracle HR are either recurring or non-recurring. Recurring elements are automatically used for each pay-period within the definition scope, whereas non-recurring elements are limited to one particular pay-period.

The majority of the pay elements are generated with the batch element entry method of the HR module. These batches are run for each pay period. The output consists mainly of elements without date-tracks (like non-recurring elements and recurring elements like name, address, etc.). But some elements are generated with date-tracks. For instance the monthly salary is generated as a recurring element with a date-track for each pay period. This opens the opportunity to make retroactive changes to these pay elements. Finally, some elements are generated through manual input, and in this case may have any kind of date-track information.

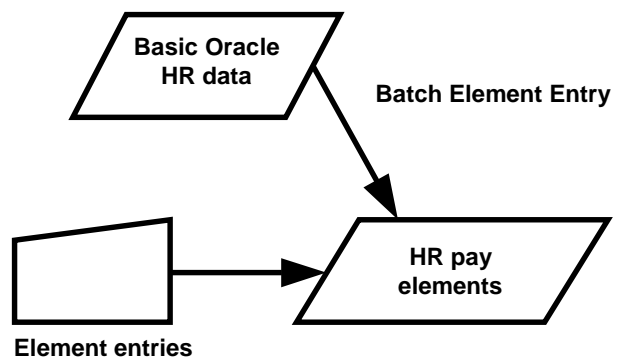


Figure 2: Sources of HR elements

A generic model is used for the design of the interface of the Oracle HR elements to the external payroll system. The input of the interface can consist of a collection of HR elements with any kind of date-track information, and the output contains all modifications to these pay elements



up until the current pay-period. Only new or modified elements appear in the output. Recurring elements without a change do not generate any data. Descriptive flexfields on element types are used to store the external code of a certain pay element. Only elements with such an external code are interfaced. So the HR elements can still be used for other purposes. Another advantage of storing the external code in a flexfield is that the interface itself does not depend on changes in the kind of pay elements. If, for instance, the external system requires a new pay element, it is sufficient to define a corresponding HR element (with the external code in its flexfield), and all generated elements of this new type are automatically processed by the interface.

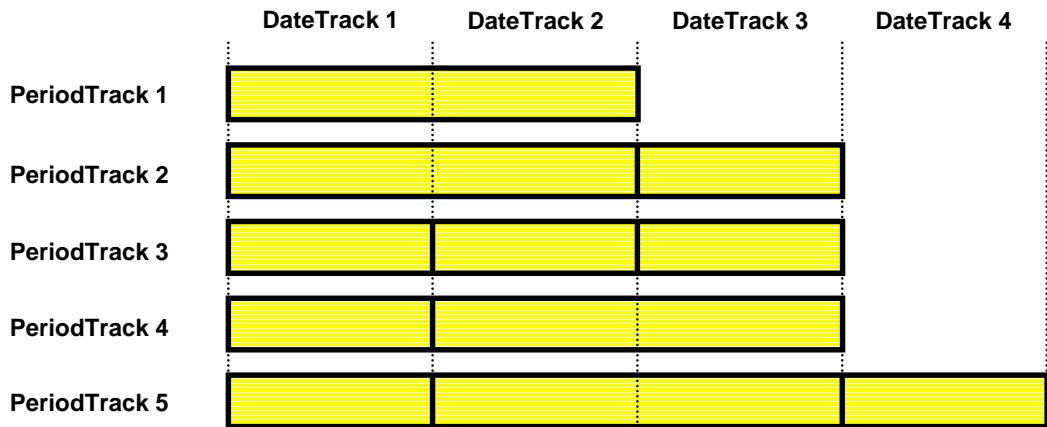


Figure 3: Example of PeriodTrack method for one pay element

In order to be able to generate the mutations in the pay elements it is at least required to store the status of the pay elements in the previous pay-period. Please note that “status” here means the complete date-tracked status of the pay elements in the previous pay-period. For optimal flexibility the interface stores the complete date-track history of all pay elements in each pay-period. We call this the PeriodTrack method (figure 3). Using this kind of history it is possible to generate the mutations in pay elements between any two successive pay-periods.

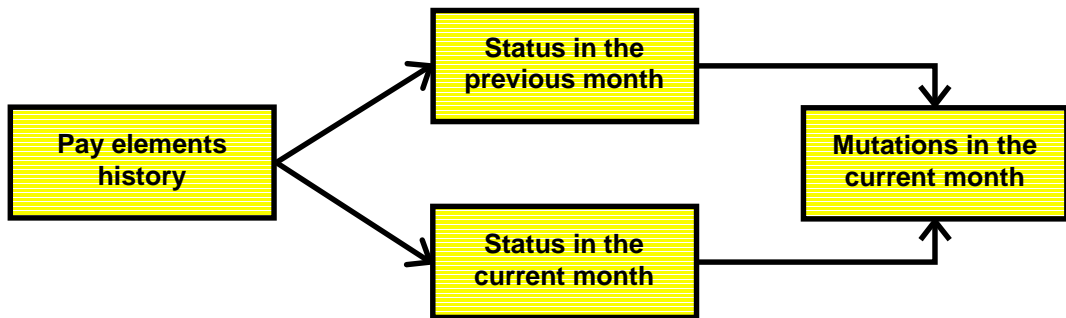


Figure 4: Generation of mutations in the current pay period

In figure 4 the history of the pay elements contains the complete PeriodTrack status of all elements in all relevant previous periods (one set of two-dimensional data). The status of pay elements in the current and previous month contains the complete date-tracked status of all elements (two sets of one-dimensional data). The mutations in the current month can include retroactive changes to recurring elements.

5 Interface processing

The interface is controlled by a number of processes that must be run in each pay-period. First of all a new “pay-period to be interfaced” must be initialized. This period is usually the actual pay-period, but may also contain an additional period for closing a fiscal year. After the initialization the history table with PeriodTrack data must be filled with mutations in the HR elements. Each pay element date-track period that has been inserted, updated, or deleted generates a new record in the PeriodTrack table. Finally the mutations of pay elements in the current pay period are generated, and transferred to the external payroll system. It is possible to interface a part of the elements. For instance, a change in the pay scales that effect all employees may be transferred (and validated) one week before the usual bunch of monthly mutations.

The mutations of pay elements in the current month are generated in a temporary table in the Oracle database. Each record in this table corresponds with one line of input data for the external payroll system. Oracle Discoverer with a dedicated business area and workbook is used for a first validation of the mutations. The powerful ad-hoc queries of Oracle Discoverer may also be used to filter the mutations. A simple script maps the output of Oracle Discoverer on the format required by the external payroll system.

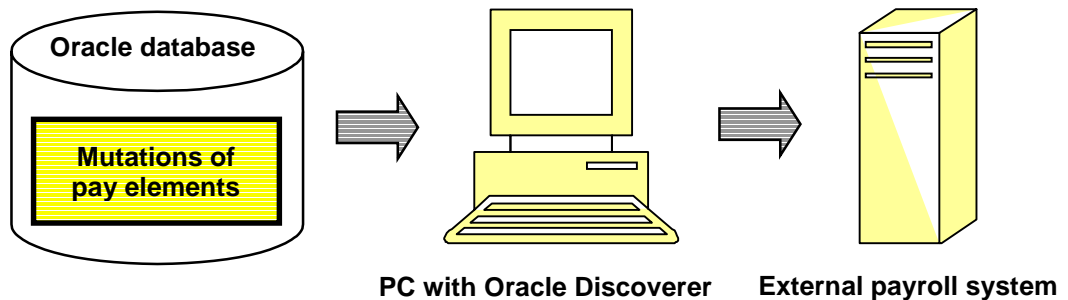


Figure 5: Transfer of mutations to the external payroll system

At NLR, we use a Dutch external payroll system (Gemal / Beaufort of Getronics). The mutations are imported in the external system by a standard process for processing indirect mutations. Because of the generic concept it will be quite easy to make an interface to another mutation-driven external payroll system. For the feedback of the external payroll system to Oracle General Ledger the standard Oracle ADI module is used.

The interface is operational since September 2000. After some months with shadow runs it became fully operational in December 2000.

6 Conclusions

Oracle HRMS is well suited for the generation and maintenance of all gross pay elements. But due to the lack of an existing Oracle Payroll for the Netherlands, we have to use an external payroll system.

A key concept in the design of a flexible interface is the mapping of all pay elements that are needed by the external payroll system onto HR elements. Not only the traditional pay elements, but also all other elements like name, address, etc. This opens up the way for a generic concept of the interface, including the possibility of retroactive changes. The code of the interface consists of less than 1000 lines of PL/SQL code.

By storing the external code of the pay elements into a descriptive flexfield of the element types, the code of the interface is independent of the kind of pay elements. This kind of design preserves the use of HR elements for other purposes in the Human Resource module.



About the Presenters

Name: Bert Kassies

Company: National Aerospace Laboratory NLR

Role: System Designer and Developer

Time working with Oracle Applications: 1.5 years

Email: kassies@nlr.nl

Name: Thierry Vergult

Company: Arinso International

Role: Senior Oracle HRMS Consultant

Time working with Oracle Applications: 2.5 years

Email: thierry.vergult@arinso.com