

Utilizing HR/Payroll APIs and 9iAS to Simplify Employee Load/Maintenance

Lev Moltyaner and Mark Langner

Introduction

Most HR/Payroll implementations require initial and sometimes periodic data loads of employee data, such as name, address, W4 information, and deductions. This information can come from a wide variety of external systems. In addition, HR managers quickly find that they would like to not only simplify and speedup employee maintenance, but also protect themselves from potentially serious data track errors that can occur. These requirements are mission critical for Maritech. As a payroll provider, Maritech has to be able to load new customers and provide a self-service employee maintenance vehicle. Additionally, Maritech needed to provide it's customers with a way to view various employees reports without getting direct access to the Oracle Apps. Maritech partnered with Procase Consulting to solve their business needs by developing Employee Web Maintenance (EWM) system. EWM includes a process to extract employee data from Oracle Apps, a web interface to maintain this data, and Oracle APIs to transfer it back into Oracle Apps. It also includes structures and processes to stage and load new employee data from external sources.

In this paper, we will describe EWM's architecture and illustrate how it solved Maritech's critical business requirements. We will then provide you with the details required to design a similar system to meet your business needs. In this walk-thru we will identify pit-falls and emphasize areas which will require special attention.

EWM Architecture

EWM has five major components:

1. EWM Tables - consolidated tables to handle Maritech's custom requirements
2. External Sources to EWM – load into staging, validate, and insert into EWM tables
3. Oracle Apps to EWM - extract data from Oracle Apps and insert into EWM tables
4. Web Interface - allows users to maintain and report employee information
5. EWM to Oracle Apps - extract all changes from EWM tables and load them into Oracle Apps

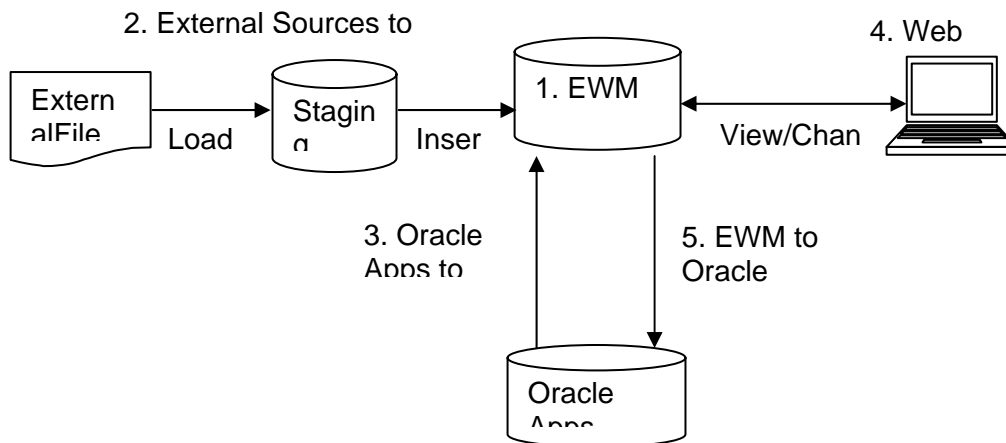


Figure 1. EWM Architecture Diagram

Requirements

Most Oracle HR/Payroll implementations will require some of the following functionality:

1. Load a batch of employees with all their historical information into Oracle Apps
 - a. Original inserts (eg. first time data conversion and load)
 - b. Addition inserts (eg. add a division or company, implement a merger)
 - c. Copy to new installation (eg. test a new configuration, migrate to a new Oracle release)
2. Perform batch updates of employee information (eg. re-org)
3. Provide employee maintenance outside of Oracle Apps
 - a. To simplify employee maintenance (eg. provide a consolidated subset of supplied forms, avoid date track errors)
 - b. To provide web-based employee maintenance without violating security policies (eg. no access to production database, encrypting sensitive data)
 - c. To insulate the end user from Date Track errors without losing the tremendous benefit within the application itself.

Employee Web Maintenance system answers all of these requirements.

The most important requirement is to provide fast and easy to use employee maintenance outside of Oracle Apps. This is accomplished as follows:

1. At the start of the payroll cycle, the Oracle Apps to EWM process updates EWM tables with last period's payroll data and latest HR data. It performs a complete replace of employee information for a specific payroll.
2. Employee Web Interface is used to update data on the web and to view reports.
3. Prior to running the payroll, after all employee maintenance has been completed, audit reports are used to validate all changes made during the period.
4. Once the changes are validated, the EWM to Oracle Apps process is used to update Oracle Apps with all employee changes.
5. An administrator running the payroll is allowed to make any changes directly in Oracle Apps if errors during the payroll run require such.
6. After the payroll is completed, the administrator begins the cycle for the next period.

Similarly it is easy to copy employee data to a new installation for testing or any other purpose. After the extracting from Oracle Apps, EWM tables can be exported and moved to the new database. Once there, steps 3-6 can be used to load data into the new installation.

The final requirement of loading data from external systems (one time or periodically) also fits nicely into this architecture. The External Sources to EWM process is used to load data into EWM tables and steps 2-6 are repeated. The same validation rules required for employee maintenance also insure data integrity during such data loads.

Maritech's Business Benefits

EWM has fit extremely well into Maritech's business model by bringing it closer to its customers. By going to the web with a product like EWM, Maritech brought a powerful tool to its customers while protecting its production system from both intrusion and customer introduced problems. With a single investment, Maritech was able to provide its customers with a powerful and flexible tool, while fulfilling its obligation to perform data loads.

The interface is easy to use, which is especially important to customers who have employees who are not highly skilled with computers. Adding new employees via a single simple screen rather than more than 10 screens of Oracle Apps is an incredible time saver. Date tracking problems are eliminated

because all data changes are set to the start of the next payroll period. EWM provides the ability to handle multiple payrolls independently. Thus, the payroll cycle can be performed for each customer independent of another. Finally, easy to use audit reports provide a convenient way for managers to validate all data changes for each payroll.

1. EWM Tables

EWM tables are based on the Oracle Apps tables but are much more streamlined because they focus strictly on our application needs (ie. EWM does not require the flexibility of an off-the shelf product). EWM tables are divided into three major subject areas.

1.1 Reference tables – contain all the reference information required for employee maintenance

EWM Tables	Description
Business groups, payrolls, common lookups, etc	Contains company and payroll reference information to validate employee information
Counties, Cities, States, Zips	Contains the address info for employee address and tax pick lists
Deductions and levies	Contains all the deductions and levies configured in Oracle Apps

1.2 Employee tables – contain all the employee information

EWM Tables	Description
Persons and address tables	Contains employee personal and employment information
Federal, state, county and city tax rules	Contains all the levels of tax rules, may have multiple records for work and resident states
Deductions and levies	All employee deductions and levies in a single denormalized table
Direct Deposits	Bank account info for direct deposits in a single table

1.3 Payroll tables – contain payroll history (not essential but provide valuable employee information)

EWM Tables	Description
Gross To Net Details	Contains all the payroll details such as earnings, deductions, bonuses, and hours in a few tables
Check, Check Details	Contains the check stub

2. External Sources to EWM

This is a fairly simple component because it does not interface with Oracle Apps. The most common need for loading data from an external source is for the first time data conversion from the legacy system. Each new interface may require its own set of staging tables because of the differences in source systems. However, based on our experience, they are all fairly similar and small changes to staging tables may suffice. The data is loaded into staging tables using SQL*Loader. In Oracle 9i, external tables can be used to eliminate the SQL*Loader process. Once the data is loaded, a validation process is used to detect any problems and create a report if necessary. If all data is valid, a process is used to move data from staging tables into EWM tables. To avoid hard coding, consider creating a table to store default values for business rules.

3. Oracle Apps to EWM

Copying data from Oracle Apps to EWM requires an intimate understanding of Oracle Apps tables. The simplest design is to create one process per subject area: reference tables, employee tables, and payroll tables.

3.1 Copying reference tables

This is the simplest of the three processes. Most tables require a simple query, which retrieves a subset of rows and columns (eg. hr_organization_units, hr_locations, pay_payrolls, etc.). Tax jurisdiction, zips, and deductions warrant special attention. Tax jurisdiction is made up of 3 portions: states, counties, and cities. The three corresponding tables in Oracle Apps are pay_us_states, pay_us_counties and pay_us_city_names. You must use the primary flag to pick a unique city name per jurisdiction because city can have multiple records per jurisdiction (ie. jurisdiction is known by several names). The zips table is used to provide pick lists for addresses and it is derived from pay_us_zip_codes. Oracle has zip_start and zip_end to represent continuous zip ranges in one row, which we converted to one row per zip to enhance the functionality of our pick list. Finally, deductions are derived from pay_element_links_f and pay_element_types_f. Attribute definitions such as default value, mandatory flag, and display sequence come from pay_input_values_f. You must exclude attributes with mandatory_flag = 'X' because they are managed by Oracle.

EWM Tables	Mapping
Business groups, payrolls, lookups, etc	hr_organization_units, hr_locations, pay_payrolls, fnd_common_lookups, etc
Counties, States, Zips	pay_us_states, pay_us_counties, pay_us_city_names, and pay_us_zip_codes
Deductions and levies	pay_element_types_f, pay_element_links_f and pay_element_values

3.2 Copying employee tables

This process copies all the employee specific information. One challenge for this process is to merge multiple independently date tracked tables such as per_people_f and per_assignments_f into one consolidated table. We encountered a performance problem when we used a "brute force" algorithm for merging these tables. A much more efficient solution is a single query with complex joins on dates. Addresses and tax rules are simple mappings from their corresponding Oracle tables. The only challenge with tax rules is to map filling statuses, which vary by state. Deductions are derived from pay_element_entries_f and pay_element_entry_values_f. Again we encountered a performance problem here because these tables maybe very large. In Maritech's case, they are currently 7.5M and 22M respectively. However, Oracle has a number of indexes on these tables and you should be able to improve performance by utilizing hints. Direct deposit information did not present any problems.

EWM Tables	Mapping
Persons and address tables	per_people_f, pay_people_groups, per_assignments_f and per_addresses
Federal, state, county and city tax rules	pay_us_emp_fed_tax_rules_f, pay_us_emp_state_tax_rules_f, pay_us_emp_county_tax_rules_f, and pay_us_emp_city_tax_rules_f
Deductions and levies	pay_element_entries_f and pay_element_entry_values_f
Direct Deposits	pay_external_accounts, pay_org_payment_methods_f and pay_personal_payment_methods_f

3.3 Copying payroll tables

This is the most complex process because it requires an understanding of Oracle's payroll process and all the Oracle internal tables used to store the payroll details. It is further complicated by

performance considerations because of the volume of data. In Maritech's example, the two largest Oracle Apps tables involved in this process are pay_run_results and pay_run_result_values have 130M and 550M records respectively. However, payroll information can be extracted efficiently for each payroll run because it is indexed appropriately. We extract all the payroll information into two denormalized tables called payroll_summary and payment_summary, which currently contain 25M and 3.5M records respectively. Maritech's payroll reporting is efficient because it is based on these two tables. Here is a sample query to populate the payroll summary table:

```

insert into payroll_summary
select ....
from pay_element_classifications c
  ,pay_element_types_f et
  ,pay_input_values_f iv
  ,pay_run_result_values rrv
  ,pay_run_results rr
  ,pay_assignment_actions aa
  ,pma_actions_summary s
-- get all R actions for this payment action
where s.payment_action_id = in_payment_action_id
-- get all assignment actions for R actions
and aa.payroll_action_id = s.payroll_action_id
and rr.assignment_action_id = aa.assignment_action_id
-- get run results and values
and rr.assignment_action_id = aa.assignment_action_id
and rrv.run_result_id = rr.run_result_id
-- value should not be 0 or null
and nvl(rrv.result_value,'0') <> '0'
and iv.input_value_id = rrv.input_value_id
and id_end_date between iv.effective_start_date and iv.effective_end_date
-- get element name and classification
and et.element_type_id = iv.element_type_id
and id_end_date between et.effective_start_date and et.effective_end_date
and c.classification_id = et.classification_id
-- 3 types of elements:
-- 1. Normal - Use Pay Value rather than Amount
-- 2. Fees - Withheld Fee Amount
-- 3. Information - PMA values are Amount, Oracle are Pay Value (Arrears)
and ((iv.name = 'Pay Value'
and c.classification_name <> 'Information')
or iv.name = 'Withheld Fee Amount'
or (iv.name in ('Amount', 'Pay Value')
and c.classification_name = 'Information'));
  
```

EWM Tables	Mapping
Payroll_summary	pay_payroll_actions, pay_assignment_actions, pay_run_results, pay_run_result_values, pay_element_classifications, pay_element_types_f, pay_input_values_f
Payment_summary, checks	pay_payroll_actions, pay_assignment_actions, pay_pre_payments, pay_org_payment_methods_f, pay_personal_payment_methods_f

4. Web Interface

Web interface is designed to simplify employee maintenance. Search page allows users to enter search criteria. If more than one employee matches the criteria, a list page is displayed to allow the user to select a particular employee.

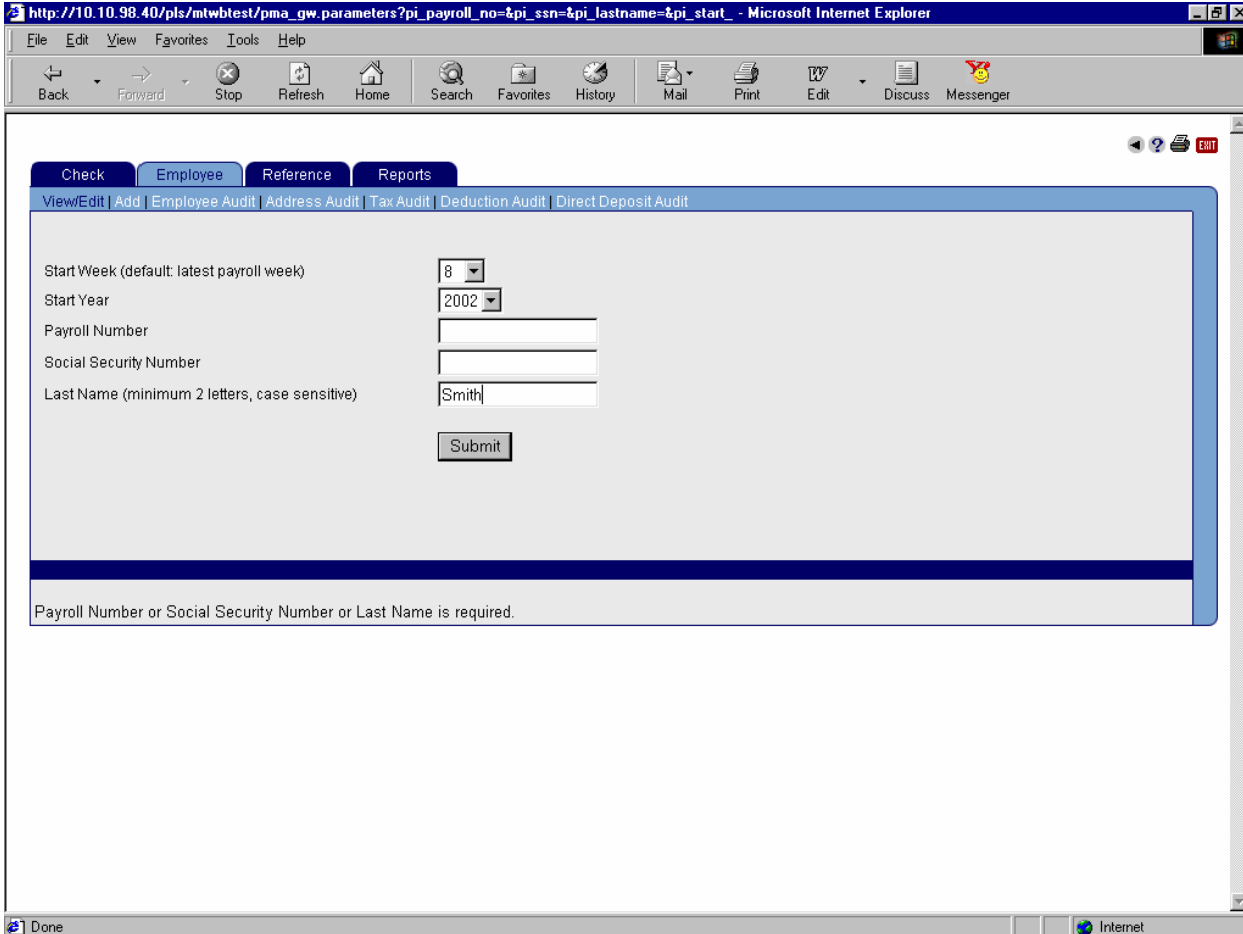


Figure 2. Search Page

View Employee page shows the current employee information. It allows users to answer most business questions with a single glance. This page is also used as the launch pad for all view, edit, and insert operations for each employee. In addition, it is used to delete records if delete operation is permitted. Add/edit/delete icons are not displayed for users who do not have update privileges.

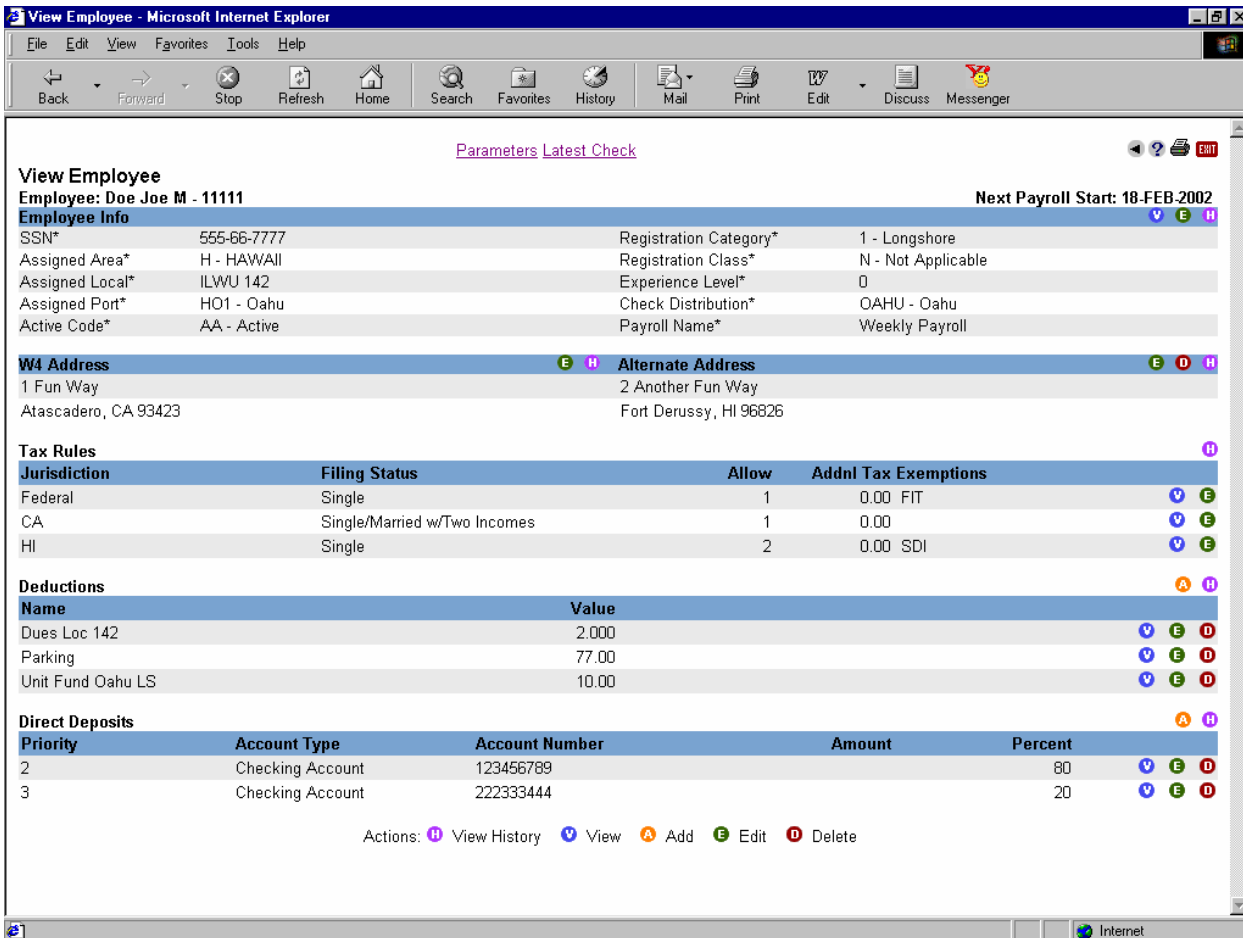


Figure 3. View Employee Page

For each section of employee data there is a history page. Each history page provides all the detailed information required to answer complex questions. History pages are formatted as lists, showing each change, who made it and when it were made. For each record in the history list, the user can drill down to see the details. There are 6 history pages: Personal Info History, Address History, Tax Rules History, Deductions History, Levies History, and Direct Deposits History.

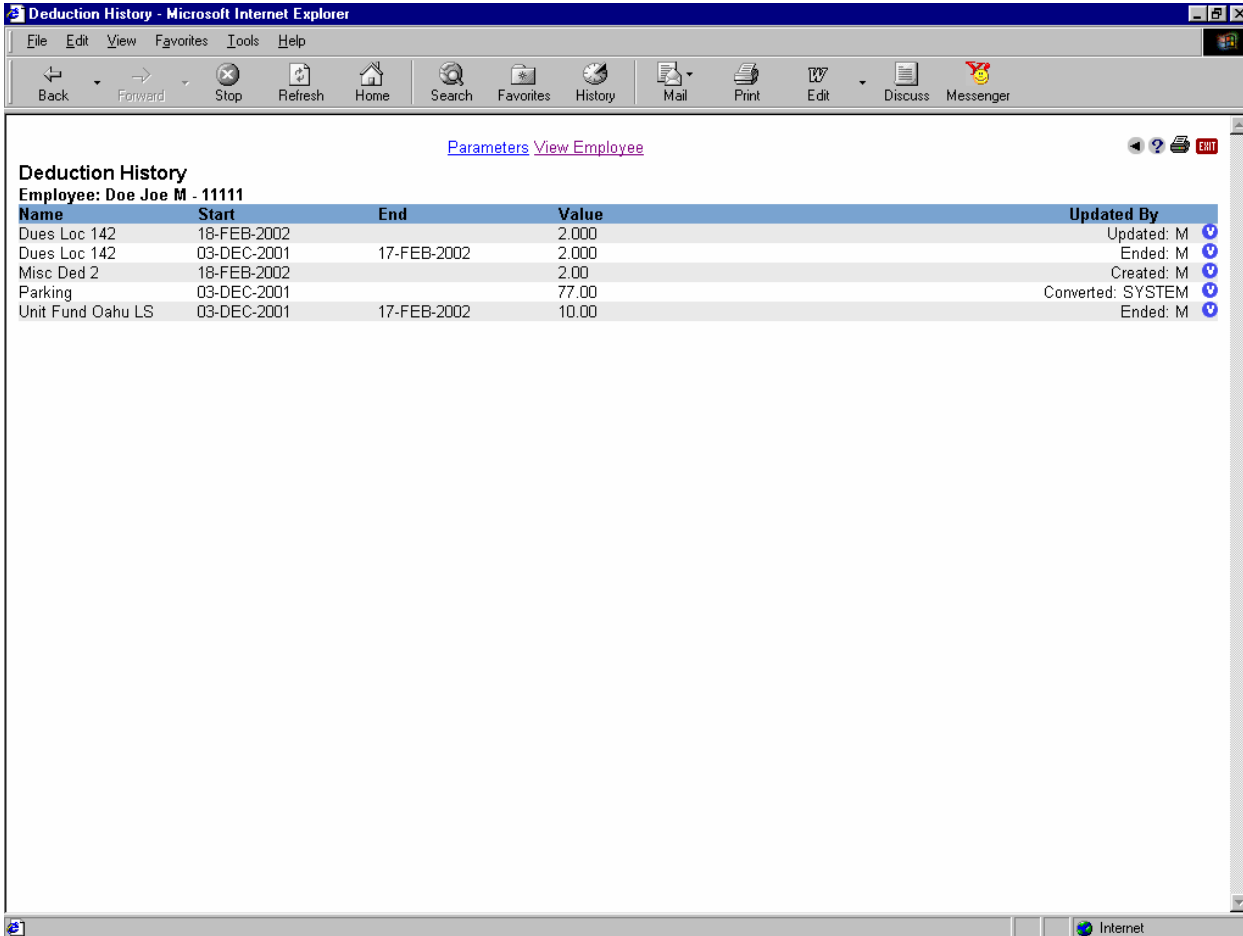


Figure 4. Deductions History Page

View/insert/edit pages are available for each section of employee data. A page in view mode simply displays the information. A page in insert mode displays the context information at the top of the page and blank fields for data entry. A page in edit mode displays the current information and allows updateable fields to be changed. Date tracking is simplified because all changes are effective on the start date of the next payroll run.

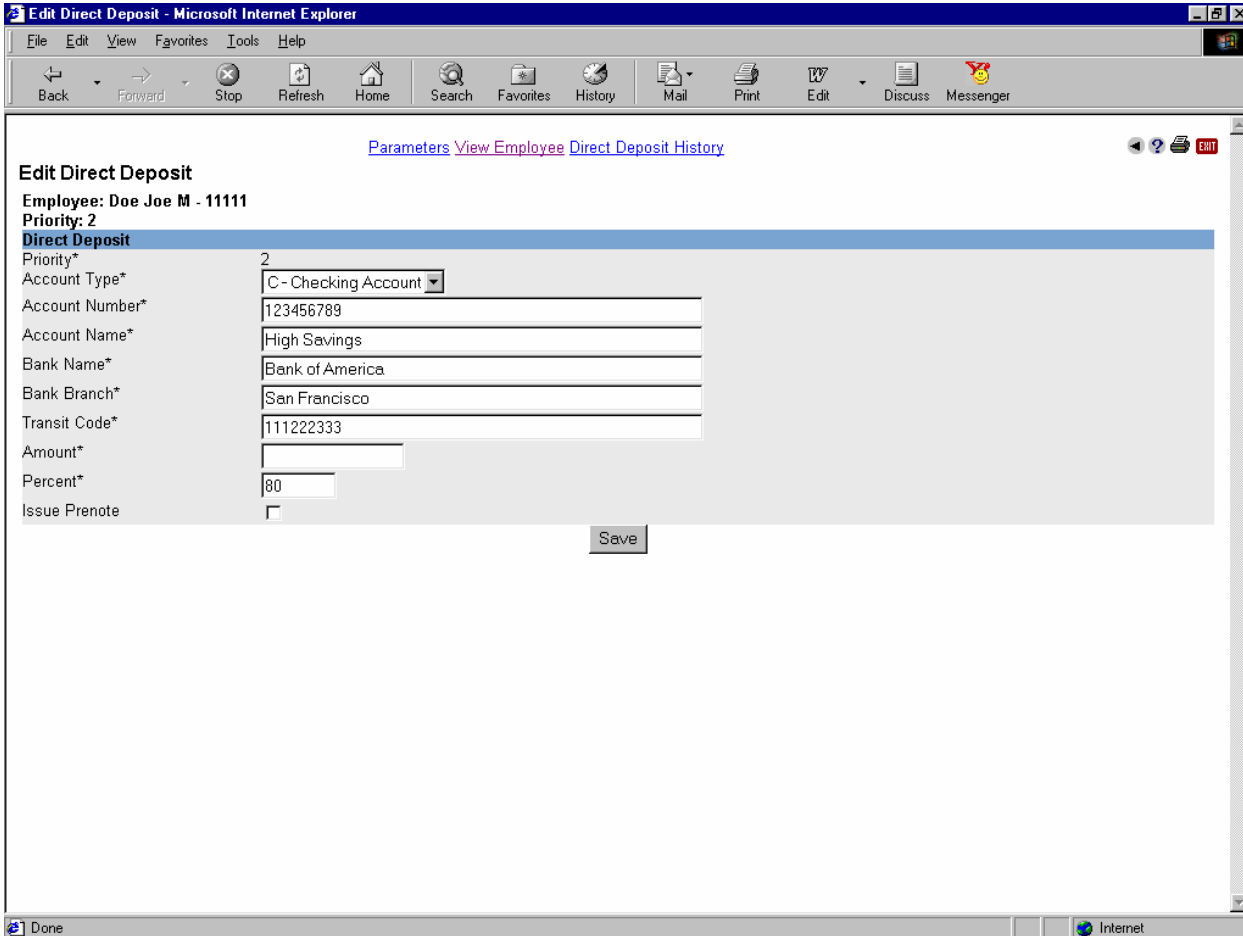


Figure 5. Edit Direct Deposits Page

Add Employee page greatly simplifies data entry by providing a single page for adding employees. It even requires the user to specify tax information, which is then used to generate tax rules based on employee's W4 address and company's address. It replaces up to 10 steps in Oracle Apps screens.

Add Employee - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss Messenger

[Parameters](#)

Add Employee

Employee Info

Payroll Number*
SSN*
Last Name*
First Name*
Middle Name
Suffix
Assigned Area* H - HAWAII
Assigned Local* ILWU 142
Assigned Port* HH1 - Hawaii
Check Message
Active Code* AA - Active
Registration Category* 1 - Longshore
Registration Class* A - Category A
Experience Level* 0
Filing Status* Single
Allowances*
Check Distribution* KAUAI - Kauai
Levy Check Distribution* LEVY - Levy Distribution (Req)
Address On Check

W4 Address

Address 1*
Address 2
Address 3
City*
State*
Zip*
County*

Alternate Address

Address 1*
Address 2
Address 3
City*
State*
Zip*
County*

City, State, Zip, and County: use Search icon to auto-populate or enter each field manually

Save

Done Internet

Figure 6. Add Employee Page

HR and Payroll reports are included to provide additional employee information. For Maritech, the most critical report is the check stub, which contains all the year to date payroll information. Links at the top of each page allows users to jump between various reports and employee maintenance pages.

Check - Microsoft Internet Explorer

Parameters View Employee Check List Previous Check Next Check

Check

Reg No	Port	SSN	Last Name	First Name	MI	PP#	Issue Date	Fed Status	State Status	LFS Exp	CD Hours	Check Number	Dist Code
11111	HO1	555-66-7777	Doe	Joe	M	3	17-JAN-02	S-0	S-0	0		3014	OAHU

Company No	Sft	Port	Wrkd	Date	Occ	Gross Inc	Cargo Pen	Cargo Pen	Allow Non-Txbl	Allow Txbl	Mileage	ST Hours	OT Hours	POT Hours	PPOT Hours	ST Rate
8003-0	2	HO1		1220	ST		81.36							2.00		27.12
8003-0	1	HO1		0108	ST		135.60					5.00				27.12
8003-0	1	HO1		0109	ST		325.44					6.00	4.00			27.12
8003-0	1	HO1		0110	ST		325.44					6.00	4.00			27.12
8003-0	1	HM1		0111	ST		135.60					5.00				27.12
8003-0	2	HO1		0112	ST		203.40						5.00			27.12
8003-0	1	HO1		0113	ST		183.06						4.50			27.12
8003-0	1	HO1		0113	ST		20.34						.50			27.12
Total							1,410.24					22.00	20.00			

Description	State	Current	YTD	Description	State	Current	YTD
Earnings		1,410.24	5,224.00	Regular Hours		42.00	143.50
FIT		-303.94	-1,209.79	Holiday	HI		732.24
Medicare EE		-20.45	-75.75	Wages Regular	HI	1,410.24	4,491.76
SIT	HI	-101.64	-380.41	Gross Wages Fd		1,410.24	5,224.00
SS EE		-87.44	-323.89	Check Amount		868.57	
Dues Loc 142		-28.20	-104.47				
Parking			-77.00				
Unit Fund Oahu LS			-10.00				
Net Pay		868.57	3,042.69				

To the order of **Message**

Doe, Joe M
 1 Fun Way
 HONOLULU HI 55555

Figure 7. Employee Check Stub

5. EWM to Oracle Apps

The last component is also the most difficult one, because of the complexity and risk of updating Oracle Apps tables. Oracle strongly recommends that customers should not update Oracle tables directly without using Oracle APIs. However, APIs are not well documented and in some cases APIs don't even exist. If APIs cannot be found, there is no choice but to update the tables directly. In either case, thorough testing is essential. In general, there is one API package per Oracle Apps table. Each package has a number of procedures whose names normally begin with create, update or delete. Create procedure is used to create a record for the first time. Update procedure is used to update an existing record. For tables which are date tracked, update can be performed in correction mode which will simply update the existing record or in update mode which will add a record with an effective date. The entire process can be divided into six sections:

5.1 Add a New Employee

Steps	API/Insert/Update
1. Insert person	hr_employee_api.create_employee
2. Update people group (was created by the person API)	hr_assignment_api.update_emp_asg_criteria
3. Update assignment (was created by the person API)	hr_assignment_api.update_us_emp_asg
4. Insert base salary	insert into per_pay_proposals hr_entry_api.insert_element_entry
5. Insert check as the payment method with priority=99	hr_personal_pay_method_api.create_personal_pay_method

5.2 Update Employee Information

Steps	API/Insert/Update
1. Update personal information if it changed, if a record exists on that date, set mode to correct, else to update	per_per_upd.upd
2. Update assignment and people group information if it exists, set mode to correct or update	hr_assignment_api.update_emp_asg_criteria hr_assignment_api.update_us_emp_asg

5.3 Insert or Update Addresses

Steps	API/Insert/Update
1. Insert new addresses	hr_person_address_api.create_person_address
2. Update existing address records	hr_person_address_api.update_person_address

5.4 Insert or Update Tax Rules

Steps	API/Insert/Update
1. Update federal tax rules (inserted by earlier API)	pay_federal_tax_rule_api.update_fed_tax_rule
2. Insert state tax rules if they have not been inserted by earlier API, otherwise update	pay_state_tax_rule_api.create_state_tax_rule pay_state_tax_rule_api.update_state_tax_rule
3. Insert county tax rules if they have not been inserted by earlier API, otherwise update	pay_county_tax_rule_api.create_county_tax_rule pay_county_tax_rule_api.update_county_tax_rule

4. Insert city tax rules if they have not been inserted by earlier API, otherwise update	pay_city_tax_rule_api.create_city_tax_rule pay_city_tax_rule_api.update_city_tax_rule
--	--

5.5 Insert or Update Deductions

Steps	API/Insert/Update
1. If deduction is ended, use the delete API to set end date	hr_entry_api.delete_element_entry
2. Update deduction if exists, else insert it (hot defaults should not be updated because they are set at company level)	hr_entry_api.insert_element_entry hr_entry_api.update_element_entry

5.6 Insert or Update Direct Deposits

Steps	API/Insert/Update
1. If direct deposit is ended, use the delete API to set end date	hr_personal_pay_method_api.delete_personal_pay_method
2. Insert bank account if it does not exist	insert into pay_external_accounts
2. Update direct deposit if exists, else insert it	hr_personal_pay_method_api.create_personal_pay_method hr_personal_pay_method_api.update_personal_pay_method

Summary

We would also like to suggest that the entire Employee Web Maintenance system would be useful for many Oracle HR/Payroll installations. However, you may also consider implementing a subset of the system. Some ideas are:

Sub-system	Required Components (EWM tables are assumed in all cases)
Data loading	External Sources to EWM and EWM to Oracle Apps
Self-service reporting	Oracle Apps to EWM and Web Interface (view only)
Employee maintenance (no reporting or data loading)	Oracle Apps to EWM (no reporting tables), Web Interface and EWM to Oracle Apps

We hope that this detailed architecture walk-thru provided you with the information and the ideas you need to solve your immediate business requirements or to simply enhance your existing solutions.