

**Oracle® Utilities Work and Asset  
Management**

RF Bar Coding User Guide

Release 1.9.0.3

July 2011

Oracle® Utilities Work and Asset Management RF Bar Coding User Guide for Release 1.9.0.3

Doc v1 rev.0 7/12/11

Copyright ©1986, 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: TUGBU Documentation

Contributing Author: n/a

Contributor: n/a

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any lost, costs, or damages incurred due to your access to or use of third party content, products or services.

---

---

# Contents

<b>What's New .....</b>	<b>5</b>
<b>1 Basic Operations .....</b>	<b>6</b>
Using the Hand-Held Device.....	6
Entering Data .....	6
Working with Lists .....	6
Using the Function Keys .....	7
Selecting a Storeroom.....	7
Logging On and Off.....	7
Checkouts/Returns.....	8
Work Order Task Checkout.....	8
Work Order Task Return.....	9
Account Checkouts/Returns.....	10
Asset Checkouts/Returns .....	10
Checkout Requests.....	11
Other Checkouts/Returns .....	11
Stock Transfer .....	11
Using Stock Transfer Transactions .....	12
Receiving a Stock Transfer .....	13
Physical Inventory.....	13
Reorder Handling.....	14
Printing Issue Tickets .....	15
Label Printing .....	15
Printing Receiving Report .....	16
Stocking.....	16
Receiving .....	17
Multi-Step Receiving .....	18
Recording Deliveries.....	19
Placing Items in Receipt.....	21
Receiving Items into the Storeroom .....	21
Vendor Performance and Returns.....	22
Using Vendor Performance and Returns .....	22
Troubleshooting.....	24
Fields/ Lists Not Displaying Data .....	24
Fonts Not Displaying Correctly.....	24
Transfers .....	24
<b>2 Installation &amp; Configuration.....</b>	<b>25</b>
Overview .....	25
General Processing .....	25
Supported Transactions .....	26
System Requirements .....	26
Host Server.....	26
Network.....	27
Oracle Database .....	27

Client Devices ..... 28

Compatibility ..... 28

Installing RFGEN ..... 28

    RFAssistant..... 29

    RFComserver ..... 29

    RFAdministrator..... 29

Installing Software ..... 30

    Configuring Printers ..... 31

    Installing Bar Code Fonts..... 32

Installing RF Bar Coding..... 33

Configuring the ODBC Data Source ..... 34

Referencing the DSN in RFGEN..... 34

Establishing Bar Code Users ..... 35

Configuring Bar Code Business Rules ..... 35

Code Table Configuration..... 37

**Cue Cards ..... 38**

---

---

# What's New

This section describes some of the major feature and functionality changes that were implemented for this release.

## **New Features for Release 1.9.0.3**

Changes in these releases include the following:

Added support to print [issue tickets](#) and [receiving reports](#) from the handheld device.

Generated: July 12, 2011 11:16 am for Product Versions: 1.9.0.3

Release 1.9.0.3

---

# Chapter 1

---

## Basic Operations

RF Bar Coding is an addition to the Oracle Utilities Work and Asset Management application that allows you to perform a variety of system functions using radio frequency (RF) devices. These hand-held computers with bar code scanning capabilities allow you to move freely about the warehouse to perform work while maintaining a real-time connection to the Oracle Utilities Work and Asset Management application.

*Note:* The RF Bar Coding [Installation & Configuration](#) Guide includes a description of how to install the software required to support bar coding functionality.

Once you have the proper bar-code software installed, you can sign on to the system using a bar code reader. The Main Bar Code Menu includes the following transactions:

- [Checkouts/Returns](#)
- [Stock Transfer](#)
- [Physical Inventory](#)
- [Reorder Handling](#)
- [Printing Issue Tickets](#)
- [Label Printing](#)
- [Printing Receiving Report](#)
- [Stocking](#)
- [Receiving](#)
- [Multi-Step Receiving](#)

### Using the Hand-Held Device

The hand-held device has a small keypad, including function and arrow keys, in addition to the scanner button and display screen.

### Entering Data

You can enter data by either scanning a bar code label or by entering text from the keypad. If you enter text from keypad, you need to press the Enter key to complete the step. If you scan a label, you do not need to press the Enter key.

The keypad also includes arrow keys for moving the cursor on the screen. In most cases, pressing the down arrow to move out of the current field is the same as pressing the Enter key.

### Working with Lists

It is usually much faster to select stock codes, work orders, purchase orders and other long numbers from a list of values, rather than using the small keypad on the hand-held device. Also, you are less likely to make an error if you select from a list. For these reasons, it is preferable to either scan data or select from a list of values if one is available.

To see a list of values associated with a field, move the cursor to the field and press the Enter key. When the list displays, you can use the arrow keys to move the highlight up and down the list to select the value you want to use. Press the Enter key to select the highlighted value and return to the previous display. You can also enter a partial value and press the Enter key to see a list of values matching the partial criteria you entered.

## Using the Function Keys

The following function keys are defined for RF Bar Coding:

F1	Clear current field
F2	Refresh the screen
F3	Options for current field
F4	Return to the previous menu
F5	Insert a new record

## Selecting a Storeroom

If you have more than one Storeroom defined for RFGEN in your User Profile, the Storeroom Selection screen displays when you enter your employee number. Use the arrow keys to select the Storeroom you want to use and press the Enter key.

All of the work you do during the session is restricted to the Storeroom selected. If you want to work on records for another Storeroom, you must logoff and begin a new session to select the new Storeroom.

The Storerooms listed on this screen are taken from your User Profile. To define Storerooms in your User Profile, enter the Key Name RFGEN on an empty line in the grid at the bottom of the User Profile window. Then enter the Storeroom IDs in the Key Value columns to the right. If only one Storeroom is listed in your User Profile, that Storeroom is the default storeroom for all your bar coding sessions.

## Logging On and Off

### How to Log On

- 1. Enter or scan your employee number at the prompt.**

Entries are case-sensitive, so be sure you use the proper case. Typically, data values are stored in upper case.

- 2. Press the Enter key if you entered your number from the keypad.**

You do not need to press the Enter key if you scan the value.

- 3. Select a Storeroom if prompted.**

If you use more than one Storeroom, the Storeroom Selection screen opens where you can select the Storeroom you want to use for this session. If you do not use more than one Storeroom, the cursor moves to the Password prompt.

- 4. Press the Enter key to bypass the Password prompt.**

Currently there is no password required. Authorization to use the system is maintained within the system based on your employee number.

The Main Menu or the Storeroom Selection menu displays. The Storeroom Selection menu displays only if you have multiple Storerooms set up for RFGEN in your User Profile.

### How to Log Off

Do any one of the following:

1. **Select <Logout> from the Main menu, or**
2. **Press the F4 key repeatedly until the Log On screen displays, or**
3. **Turn off your hand-held device.**

## Checkouts/Returns

You can use Checkout/Return transactions to issue stock items from the storeroom and to return stock items you haven't used on a job.

Checkouts and returns can be charged against work order tasks, accounts and assets/functions. Each type of checkout follows a similar procedure. In each case, you begin by entering an employee number and selecting the type of checkout you want to perform. No matter which type of checkout you are performing, you cannot checkout more than the current on-hand quantity.

When returning against a work order task, you cannot return more than was checked out against the current work order. When returning against an Account or an Asset, however, you can return any quantity of any stock item.

Settings in the CHECKOUT ALLOWABLE CHARGE TYPE business rule determine which charge types are valid for these transactions.

### How to Initiate a Checkout/Return session

1. **Select Checkout/Return from the Main Menu.**

To select a menu option, use the up and down arrow keys to highlight the one you want, then press the Enter key on the bar code scanner.

2. **Enter or scan your employee number.**

If you provide only a partial entry and press the Enter key or the down-arrow, the system displays a list of employee numbers matching your partial entry.

Move the up/down arrow keys to highlight your employee number, then press the Enter key. You can also select another employee's number if you want to attribute your work to another employee.

After you have selected an employee number, the system displays a list of the types of checkouts and returns you can perform.

3. **Highlight the task you want to perform and press the Enter key.**

You can select Work Order Task, Account Asset/Function, Checkout Request, Service Request or Project/Subproject.

## Work Order Task Checkout

Select Work Order Checkout if you are checking out stock items to be charged against a specific Work Order/Work Order Task.



### How to Checkout a Stock Item against a Work Order Task

1. **Select Work Order type from the Checkout/Return Type menu.**
2. **Scan or select the Work Order number.**

To display a list of Work Order numbers you can select from, press the Enter key or the down-arrow key. If there are too many outstanding Work Orders, you'll be asked to narrow your search. In that case, enter a partial Work Order number and all matching Work Orders will be listed. When the list displays, highlight the one you want to select and press the Enter key.

3. **Enter the appropriate Work Order Task number.**

Press the Enter key to see a list of tasks you can select from. If there is only one Work Order task, the system will automatically enter the Work Order Task number when you press the Enter key.

When you enter the Work Order Task number, the system assigns an Issue Ticket number and opens the Transaction Menu where you can indicate if you are performing a checkout or return against the Work Order Task. The Transaction Menu also gives you the option to quit the Checkout/Return process and return to the Main Menu.

4. **Select Checkout and press the Enter key.**

The Checkout Detail screen for the Work Order task opens.

5. **Scan the stock code you want to checkout, or press the Enter key to choose from a list of stock codes.**

Highlight the stock code you want and press the Enter key. If the stock code you select is a Lot Managed item, you will be asked to select a Lot number.

You can not checkout Direct stock items.

6. **Enter the quantity you want to checkout and press the Enter key.**

The requested quantity cannot exceed the on-hand quantity.

When you enter the quantity, the system records the checkout transaction and displays a PROCESSED message.

The system generates a unique issue ticket number. While it looks as if this number can be edited, any changes you make to it are not saved and the original issue ticket number is used to identify this transaction.

7. **Press the Enter key to return to the Transaction menu.**

From the Transaction menu you can continue to checkout or return stock items against this Work Order Task or return to the Main Menu.

### Work Order Task Return

A Work Order Task Return transaction is identical to the Checkout procedure until you reach the point where you indicate the type of transaction you want to process. In this case, you select "Return" from the Transaction menu.

### How to Return a Stock Item against a Work Order Task

1. **Select Work Order from the Checkout/Return type menu.**
2. **Scan or select your Work Order number.**
3. **Enter the Task number.**

When you enter the task number, the Tans action menu opens.

**4. Select Return and press the Enter key.**

The Checkout Detail screen displays.

**5. Scan or select the Stock number you are returning.**

You can return only items that have previously been checked out against the Work Order.

**6. Enter the quantity that you are returning.**

The ITD (Issued To Date) quantity on the screen reminds you of the quantity of this item issued against the current Work Order. Depending on settings in the Checkout Processing Business Rule you may or may not be able to return more than what was issued.

When you enter the quantity, the system records the return transaction and displays a PROCESSED message.

**7. Press the Enter key to return to the Transaction menu.**

From the Transaction menu you can continue to checkout or return stock items against this Work Order Task or return to the Main Menu.

## Account Checkouts/Returns

Select Account from the Checkouts / Returns Type menu to begin an account-based stock issue or return. Account Checkouts and Returns follow a similar process as described above for Work Order Task checkouts and returns, but with two differences:

- After selecting Account from the Checkout / Return Type menu, you must enter an account number for the transaction. To see a list of accounts, provide a starting value in the account prompt, and press the Enter key. It's important to provide an initial value since, without one, there may be too many numbers to list effectively.
- Unlike Work Order Return transactions, Account Returns allow you to specify any positive quantity of any stock code. There is no constraint of having to previously checkout material against the account number.

## Asset Checkouts/Returns

Asset Checkouts and Returns follow steps similar to other checkouts and returns, but you must first provide an Asset Record Type and an Asset ID.

### How to Checkout a Stock Item against an Asset

**1. Select Asset/Function from the Checkouts/Returns Type menu.**

The Asset Checkout/Return menu opens.

**2. Press the Enter key to display a list of Asset Record Types.**

**3. Move the highlight and press Enter to select an Asset Record Type.**

When you select the Asset Record Type, a list of assets displays.

**4. Select the Asset you want to use**

Now you can use steps similar to those in other checkout procedures to checkout and return any stock item in any quantity against the Asset.

## Checkout Requests

Checkout Request transactions follow similar steps, but you must first provide a Checkout Request number.

Since Checkout Requests are designed for immediate, one time use, parts can not be returned against them. Instead, parts must be returned against an account number, project number, or another charge type.

### How to Checkout a Stock Item against a Checkout Request

1. **Select Checkout Request from the Checkouts/Returns Type menu.**

The Checkout Request menu opens.

2. **Scan the Checkout Request number, or press the Enter key to select from a list of Checkout Request numbers.**

When you do, the system assigns an Issue Ticket number.

3. **Press the Enter key to continue.**

The Transaction Menu opens where you can indicate if you are performing a checkout or return.

4. **Select Checkout.**

A new screen opens where you can select the stock code and quantity you want to checkout.

5. **Scan the stock code you want to checkout, or press the Enter key to choose from a list of stock codes on the Checkout Request.**

6. **Enter the quantity you want to checkout and press the Enter key.**

The system processes the checkout and returns to the Transaction menu where you can checkout additional items against the Checkout Request or return to the Main Menu.

## Other Checkouts/Returns

The remaining types of checkout and return transactions (Service Requests and Project/Subproject) follow a similar process to the one described above for Work Order Task checkouts and returns.

## Stock Transfer

A Stock Transfer transaction allows you to move stock from one location to another. Select Stock Transfer from the Main Menu to open the Stock Transfer screen where you can create a new Stock Transfer request or work with existing Stock Transfer requests.

Stock Transfer requests identify the stock item and quantity being transferred, the two storerooms involved in the transfer, and the identity of the person making the request. Once the transfer data is complete, separate transactions are required to issue and receive the stock in the appropriate storerooms.

From the Stock Transfers window, press Enter or the down arrow to display the Item Transfer menu, showing a list of the following transfer options:

**Request** - Select Request to add additional item to an existing Stock Transfer Request.

**Issue** - Select Issue to move the requested material from the issuing storeroom.

**Receive** - Select Receive to record the receipt of the requested material in the receiving storeroom.

**Cancel** - Select Cancel to return to the Stock Transfer window.

## Issuing a Stock Transfer

You can only complete an issue transaction if the issuing storeroom is your default storeroom for this session.

When all the material to transfer is specified on the request, the person initiating the move performs an issue transaction to record that the material is leaving the issuing storeroom.

### Related Topics

[Stock Transfer](#)

## Using Stock Transfer Transactions

[How to Create a Stock Transfer](#)

[How to Add Stock Items to a Transfer Request](#)

[How to Issue a Stock Transfer](#)

### How to Create a Stock Transfer

1. **Select Stock Transfer from the Main Menu.**

2. **Press the F5 key.**

The New Stock Transfer screen displays.

3. **Enter the date.**

You must enter the day and month in separate steps. After you enter the day number, a list of months is presented from which you can select, and the year defaults to the current year.

4. **Enter the Storeroom that will to issue the stock.**

You can press the Enter key to select from a list of storerooms.

5. **Enter the Receiving Storeroom or select it from the list of Storerooms.**

The system generates a stock transfer request number. No description of the receiving storeroom displays on this screen, only the storeroom ID.

You can now add Stock Items to this record.

6. **Press the Enter key to create another Transfer Request.**

If you do not want to create another request, press the F4 key to return to the previous menu.

### How to Add Stock Items to a Transfer Request

1. **Select Stock Transfer from the Main Menu.**

2. **Scan or select the Transfer Request you want to open.**

You can press the Enter key to display a list of available requests.

When you scan or select the number, the system opens the Transfer Record.

3. **Press the Enter key to display the Item Transfer options window.**

4. **Select the Request option.**

The Stock Transfer Detail screen opens.

5. **Scan or select the Stock Code for the items being transferred.**

The available list contains all the available stock items in the issuing storeroom except the ones already referenced by the request.

When you enter the Stock Code, the display shows the on hand quantity (O/H) in the issuing storeroom.

**6. Enter the quantity you want to transfer.**

You can ask to transfer more than the current on hand quantity.

After you enter a quantity, the system displays a PROCESSED message confirming that the item has been added.

**7. Press the Enter key.**

The Item Transfer options screen opens where you can continue working with this record. If you want to return to a previous menu, press the F4 key.

### How to Issue a Stock Transfer

**1. Open the appropriate Stock Transfer record, navigate to the Item Transfer window, and select Issue.**

**2. Scan or select the Stock Code for the item being transferred.**

You may not issue material against this request that has already been issued.

**3. Enter the quantity being transferred.**

When you enter a quantity, the system displays a PROCESSED message confirming that the item has been issued.

**4. Press the Enter key.**

The Item Transfer options screen opens where you can continue working with this record. If you want to return to a previous menu, press the F4 key.

You cannot issue material against this request that has already been issued. If you are unsure of what remains to be issued, return to the Stock Transfer window and press Enter in the Issue Storeroom field to see the stock quantity that remains to be issued.

## Receiving a Stock Transfer

Receive transactions record when the transferred stock is placed in the destination Storeroom.

Receive transactions follow similar steps as Issuing transactions and result in a confirmation of the receipt activity. You can only complete a Receive transaction if the Receiving Storeroom is your default Storeroom for this session. You must receive the entire quantity issued.

## Physical Inventory

Physical Inventory transactions count stock items to ensure inventory accuracy. Physical inventory requests are generated on-line in the database. You cannot create a Physical Inventory record using the hand-held scanner. You can, however, use bar coding to process Physical Inventory records once they are created.

To conduct a Physical Inventory with a bar code reader, select the stock item from the list, and enter the quantity of the item. Whether the on hand inventory quantity is displayed, or not, depends on how the Physical Inventory Rules business rule is configured in the Administration Subsystem. If the Blind Count option in the business rule is set to Yes, the current on-hand quantity is hidden during Physical Inventories.

If the quantity you counted differs from the quantity expected, you'll be warned and given a chance to recount the item. You may not enter a zero quantity.

Depending on how your organization has configured the system, inventory adjustments may or may not be conducted using the hand-held scanners. If adjustments are not allowed and your count differs from what the system expects, you'll have to abandon the current counting by pressing F4. Report the discrepancy, then choose another request.

### How to Process a Physical Inventory Record

1. **Select Physical Inventory from the main menu.**
2. **Press the Enter key to display a list of Physical Inventory records.**

You can also enter the Physical Inventory directly if you know it, or enter a partial Physical Inventory number to narrow the list. Only Physical Inventory records for your default storeroom appear on the list.

3. **Select appropriate Physical Inventory record from the list.**
4. **Scan or select a stock code you want to count.**

Lot managed stock is not supported in the current release.

5. **When you scan the stock label or select the stock item from the list, the system supplies additional details about the item.**

Whether or not the on hand inventory quantity is displayed depends on how your organization has configured the Physical Inventory Rules business rule. If the Blind Count key value in the business rule is set to YES, the expected quantity does not display.

6. **Count the stock item and enter the quantity.**

If the quantity you counted differs from the quantity expected by the system, you'll be warned and given a chance to recount the item.

Depending on how your organization has configured the system, inventory adjustments may or may not be conducted using the hand-held scanners. If adjustments are not allowed and your count differs from what the system expects, you'll have to abandon the current counting by pressing the F4 key. Report the discrepancy, then choose another request.

## Reorder Handling

RF Bar Coding supports two types of reorder handling transactions:

- **Keep Fill Requests** are completed by vendors during periodic re-supply or consignment visits. During these visits, vendor personnel can use a hand-held bar code reader to enter required quantities for the stock items they maintain.
- **Reorder Review Requests** are similar to Keep Fill Requests, except they are initiated by storeroom personnel as they notice what appear to be shortages. Otherwise, both procedures are identical and both result in reorder review records.

When either type of request is processed, the system issues a reorder request for the quantity of the stock item entered. If a blanket contract exists for the stock item, the reorder request is drawn against that blanket contract.

### How to Process a Keep Fill Request or a Reorder Review Request

1. **Select Reorder Handling from the Main Menu.**

Select Reorder Review Request instead you want to process a Reorder Review transactions. Other than your selection from this menu, the procedures for the two Reorder transactions are identical.

2. **Select Keep Fill Request from the submenu.**
3. **Scan or select the appropriate stock code.**
4. **Enter the quantity to be ordered.**

When you enter a quantity, the system displays a PROCESSED message.

When a Keep Fill request is processed, the system issues a reorder request for the quantity of the stock item entered. If a Blanket Contract exists for the stock item, a reorder request is drawn against that Blanket Contract.

5. **Press the Enter key.**

The display returns to the Stock Code prompt where you can enter the next stock item.

When you are finished entering Keep Fill Requests, press the F4 key to return to the Main Menu to select other functions or end your bar coding session.

## Printing Issue Tickets

You can print issue tickets from your handheld device. In order to use this feature, you must install the SRW database package. Until the SRW package is run, the stored database procedure SDBP\_RFGEN\_PRINT is invalid. For more information, see [Database Scripts](#).

### How to Print Issue Tickets

1. **Select Issue Ticket from the Printing sub menu.**

In order to view the Issue Ticket menu option, the Issue Ticket rule key in the Bar Code Configuration business rule must be set to ON.

2. **Enter the Issue Ticket No of the record that you want to print.**

3. **Select the printer alias.**

Once you select the printer alias, the actual network name of the printer also gets populated.

4. **Enter the number of copies of the issue ticket that you want to print.**

5. **Press Enter.**

The issue ticket prints on the specified printer.

## Label Printing

Label formats are defined in the Software bar-code printing system that supports RF Bar Coding. See the RF Bar Coding [Installation & Configuration](#) Guide for specifics on setting up these label formats.

Given that the labels are defined, you have to provide only two bits of information to begin the Print Label process - the output printer and the label type.

In addition, you must configure two elements within Oracle Utilities Work and Asset Management before you can print labels.

- List the Bar Code Print Label function in the Responsibilities of users who will need to access this functionality from the Multi-Step Receiving module.
- Set the Bar Code Label Drop Directory in the Default Directories business rule.
- Define printer labels in code table 80.

After the necessary software has been installed and configured, you can print labels by using either the Print Bar Code Labels action or a hand-held bar code reader. In either case, you specify a printer to use, the type of label you want to print, and the information to include on the label.

### How to Print Labels

1. **Select Label Printing from the Printing sub menu.**
2. **Select a destination printer.**
3. **Select a defined label from the list and press Enter.**
4. **Enter the details regarding the selected label.**

The details you can enter include stock number and quantity to print.

5. **Press Enter.**

The label prints on the specified printer.

## Printing Receiving Report

You can print the receiving report from your hand-held device.

### How to Print Receiving Reports

1. **Select Receiving Report from the Printing sub menu.**

In order to view this menu option on the handheld device, set the Receiving Report rule key in the Bar Code Configuration business rule to ON.

2. **Enter the Purchase Order No of the receiving report you want to print.**
3. **Select the printer alias.**

Once you select the printer alias, the actual network name of the printer also gets populated.

4. **Enter the number of copies of the receiving report that you want to print.**
5. **Press Enter.**

The receiving report prints on the specified printer.

## Stocking

A Stocking transaction allows you to place items in the Storeroom without requiring that you perform a receipt or Stores Issue / Return. For example, if some unclaimed parts were found in the work area, you could use a Stocking transaction to return these items back into the Storeroom. Stocking transactions increase the Inventory Quantity for the entered stock/storeroom code and adjust the average unit price for the storeroom.



### How to Add Items to the Storeroom

1. **Select Stocking from the Main Menu.**
2. **Scan or select the Stock Number you want to add to the Storeroom.**
3. **Enter the quantity of the stock Item you are placing in the Storeroom.**

When you enter a quantity, the cursor moves to the Unit Price field and defaults to the current Average Unit Price for the item.

4. **Accept current price or enter a new price for the items you are putting in the Storeroom.**

If you enter a new price, the system calculates a new Average Unit Price for the item.

5. **Press the Enter key.**

The system displays a PROCESSED message indicating that the quantity and price information has been added to the Storeroom.

6. **Press the Enter key.**

The cursor moves to the Continue or Quit prompt, where you can press Q to return to the main menu or C to return to the Stocking screen where you can scan or select the next part number to add to the Storeroom.

## Receiving

You can use Receiving transactions to match items received from vendors against Purchase Order line items. Receiving transactions record the full or partial receipt of the items, updates the Purchase Order Line Items with receiving information, and updates the records in the Storeroom Catalog module in main application with a new On-Hand Quantity and calculated Average Unit Price. You can also use Receiving transactions to return items to vendors for replacement or credit.

### How to Receive Stock Items against a Purchase Order

1. **Select Receiving from the Main Menu.**

The Receiving Menu opens where you can record basic information about the Purchase Order.

If you enter Y in the Receive All box, the Transaction Type and PO Item fields disappear. You need only to enter the PO Number and Waybill and Packing Slip information to complete the transaction.

2. **Enter N in the Receive All box.**

Or enter Y if you want to receive all remaining line items on the Purchase Order.

3. **Select Receive from the Transaction Type list and press the Enter key.**

Or select Return or Credit Return if you are doing a return transaction. Returns and Credit returns follow similar steps to a Receive transaction.

4. **Scan the Purchase Order number, or press the Enter key to select from a list of PO numbers.**

If you are selecting a PO number from the list, you may need to enter a partial number in the field before pressing the Enter key to narrow your search.

Depending on settings in the Receiving Configuration business rule, you may or may not be able to receive more items than were ordered on the original PO.

5. **Scan the Line Item number, or press the Enter key to select from a list of Line Items on the Purchase Order.**

6. **Scan the Waybill Packing Slip and Packing Slip information if available.**

Or press the Enter key to bypass these fields.

7. **Press the Enter key to Continue.**

A new Receiving screen opens where you can enter information about the Stock Item you are receiving.

8. **Enter the Quantity of the stock item being received.**

The system prompts with the quantity remaining to be received. You can accept this quantity or enter another quantity.

9. **Scan or Enter Bin number information if appropriate.**

Or press the Enter key to skip the Bin information.

If you want to remove any remaining on order quantities, enter a Y in the Force Complete box.

10. **Press the Enter key to accept the Force Complete “N” prompt.**

Change the Force Complete setting to Y before pressing Enter if you want to remove any remaining on order quantities and mark the item as fully received.

11. **Scan or Enter the Lot or Component ID information if prompted.**

Lot and Component information is only required if the stock item you are receiving is a lot managed item or a trackable stock item. After you enter the Lot or Component ID information, press the Enter key to continue.

12. **Enter Y or N in the Print Labels box.**

Enter Y to print a bar code label for the stock item or enter N if you do not want to print a label. The system prompts with an N in this field.

13. **Press the Enter key to complete the Receiving transaction.**

The display returns to the Receiving menu where you can continue receiving items.

## Multi-Step Receiving

The Multi-Step Receiving process allows you to track the delivery of items through several distinct steps from the time that they arrive at the loading dock until they are placed in the storeroom. With Multi-Step Receiving, deliveries can be processed initially by someone with a basic knowledge of the receiving process and then be completed by someone with additional qualifications who accepts the items into the storeroom.

Before you can use Multi-Step Receiving with a bar code device, your organization must configure the system for Multi-Step Receiving as described in the User Guide. In particular, the Enforce Receiver business rule must be set to ON. In order to capture discrepant shipment information, the Vendor Options business rule must also be configured.

The multi-step receiving process consists of three steps:

**Step 1: [Recording Deliveries](#)** - The multi-step receiving process begins when a shipment arrives and a new MSR record is created to hold the basic delivery information. During this step, you will create the new record, and enter information at the main (header) level.

**Step 2: [Placing Items in Receipt](#)** - Once the basic information about the delivery has been entered, the next step is to identify and count items received. Counting and recording the items

received is called placing items in receipt. You can place items in receipt at the same time you create the MSR record and enter the delivery information, or you can perform this step as a separate process.

**Step 3: Receiving Items into the Storeroom** - Finally, items are accepted and received into the storeroom. Before receiving items, you may need to inspect them or conduct other tests. Items that are identified as “quality” items can only be received by persons authorized by your organization to receive quality items.

You have two options for receiving the items currently In Receipt. You can select the Receive All option to receive all In Receipt items on the MSR record. This makes the items available in the Storeroom and the system prompts you for Vendor Performance and print Barcode Label prompts. If you do not want to use the Receive All action, you can receive each item individually.

**Optional Step: Vendor Performance and Returns** - Items that are not accepted can be returned to the vendor for exchange or refund. Any discrepancies noted during the receiving can be recorded on the MSR record and used to calculate vendor performance.

Vendor Performance functionality allows you to rate a vendor's performance based on delivery time and other attributes established by your organization. Over time, this information helps you to assess a vendor's ability to deliver items by the promised date and in satisfactory condition.

## Recording Deliveries

The Multi-Step Receiving process begins when a shipment arrives and a new MSR record is created to hold the basic delivery information. During this step, you will create the new record, and enter information at the main (header) level.

[How to Create a MSR Delivery Record](#)  
[How to Record Delivery Information](#)

### How to Create a MSR Delivery Record

1. **Select Multi-Step Receiving from the Main menu.**

The Delivery ID screen opens.

2. **Press the F5 key.**

3. **Press the Enter key to accept the current date as the Delivery Date.**

You can change the delivery date later if necessary.

The system creates a new MSR record and displays the Delivery ID number.

If your system is not set to generate Delivery ID numbers automatically, you need to enter a unique ID number to identify the record.

4. **Press the Enter key to Continue.**

The system returns to the Delivery ID screen.

### How to Record Delivery Information

1. **Select the appropriate Delivery ID.**

If a record does not yet exist for the delivery, you must first [create the MSR record](#) and then select the Delivery ID from the list of values.

2. **Enter Y at the MSR Header prompt to add delivery information.**
3. **Enter Y at the Change PO prompt.**

If you do not want to record a PO number, as in the case of mail or parcel deliveries, enter N at the Change PO prompt.

**4. Scan the PO number or select it from the list of values.**

**5. Use similar procedures to enter Container and Carrier information.**

Enter Y at the Change prompt and select the desired value from the list.

**6. Change the Delivery Date if desired.**

As you move through the Month, Day and Year fields, the system defaults the date information from the previous screen. You can change the date information or press the Enter key each time to accept the default and move to the next field.

**7. Press the Enter key to continue.**

The system opens a new window where you can enter Marker and UPC information and indicate the number of containers received.

**8. Scan or Enter a Marker and UPC, if applicable.**

A marker is a temporary, highly-visible identifier you assign to help locate the shipment at your facility.

The UPC is the carrier's identifying number for the shipment.

**9. Enter the number of containers in the shipment.**

**10. Press the Enter key to Continue.**

The PO Item List screen opens showing the total number of line items on the PO to be received.

**11. Enter Y or N at the Pick All prompt.**

"Picking" an item copies the item from the Purchase Order to the MSR record and makes it available for further processing.

**Y** - If the shipment contains several different items, enter Y at the Pick All prompt to create line items on the MSR record for all open PO line items. Using the Pick All feature is faster than creating the line items individually and there is no penalty for creating line items that you do not use. If you enter Y at the Pick All prompt, the screen changes to allow you to select the special options listed in the sidebar. To prepare for the next section, select Go to Item Detail to open the Item Detail screen.

**N** - If you want to select the items individually, enter N at the Pick All and Item List prompts, then enter Y at the Item Detail prompt to open the Item Detail screen.

When you enter Y and the Pick All prompt, the system displays the following Pick All options:

**In Receipt** - Moves all To Receive quantities to In Receipt.

**Receive** - Moves all In Receipt quantities to Receive.

**Receive and Accept** - Combines the previous options and moves all To Receive quantities to Receive.

**Go To Item Detail** - Opens the Item Detail screen.

**MSR Main Menu** - Opens the initial MSR menu screen.

*Note:* The Pick All options do not process Lot, Quality, or Component items. These items must be received individually.

## Placing Items in Receipt

Once the basic information about the delivery has been entered, the next step is to identify and count items received. Counting and recording the items received is called placing items in receipt. You can place items in receipt at the same time you create the MSR record and enter the delivery information, or you can perform this step as a separate process.

### How to Place Items in Receipt

If you need to navigate to the Item Detail screen, select the Delivery ID on the MSR Main Menu, then enter Y at the PO Item prompt, N at the Pick All prompt, N at the Item List prompt, and Y at the Item Detail prompt.

1. **On the Item Detail screen, select the Item you want to process.**

2. **Enter N at the Vendor Rating prompt, if displayed.**

The Vendor Rating prompt only displays if the record already has In Receipt. Using Vendor Rating is discussed in more detail on page 22.

3. **Select In Receipt from the Type list and press the Enter key.**

4. **Enter Y and the Continue prompt.**

5. **Scan or enter a Bin number.**

The system supplies the default Bin ID if one exists. To select another bin from the list of values, clear the default Bin ID by pressing space, then press the Enter key to display the list of values.

6. **Enter the quantity you want to place in receipt.**

If necessary, you can enter more than the quantity ordered.

7. **Enter Y at the Place in Receipt? prompt and press the Enter key.**

The system makes this quantity of the item available for receiving into the Storeroom.

8. **Enter Y or N at the Process Another? prompt.**

Enter Y if you want to select another item for this record. Repeat steps 2-7 until you have placed all items in receipt that you want to process.

Enter N if you are finished selecting Items.

## Receiving Items into the Storeroom

Finally, items are accepted and received into the storeroom. Before receiving items, you may need to inspect them or conduct other tests. Items that are identified as "quality " items can only be received by persons authorized by your organization to receive quality items.

You have two options for receiving the items currently in receipt. You can select the Pick All > Receive option to receive all In Receipt items on the MSR record. When you do, the screen will change to show the number of items updated. If you do not want to use the Pick All > Receive option, you can receive each item individually.

You can not use the Receive All option to receive Lot, Quality, or Component items. These items must be received individually.

### How to Receive Items

1. **Select the appropriate Delivery ID.**
2. **Enter Y at the PO Item List prompt.**

**3. Enter N at the Pick All prompt.****4. Enter Y at the Item Detail prompt.**

Or enter Y and the Item List prompt if you have previously picked all and now want to receive all items. If you do this, you can skip the next step.

**5. Select the item you want to receive from the list of values.****6. Enter N at the Vendor Rating prompt.**

If you wanted to enter Vendor Rating information, you would enter Y and proceed as discussed in the following section.

**7. Select Receipt from the Type list and press the Enter key.****8. Enter Y at the Continue prompt.****9. Enter the Quantity you want to receive.**

You cannot enter a quantity larger than the In Receipt quantity.

**10. Enter Y or N at the Print Label prompt.**

If you enter Y, the system opens a new window where you can select a printer, enter a serial number (if appropriate), and specify the number of labels to print.

**11. Enter Y or N at the Force complete prompt.**

You can use the force complete to indicate that no additional quantities will be received for this item, even though more were ordered.

**12. Enter Y at the Apply Receiving prompt.**

The system marks the item as received and makes the quantity available in the Storeroom.

**13. Enter Y or N at the Process Another? prompt.**

If you select Y, the system returns you to the Select Item screen where you can choose another item on this MSR record.

Repeat steps 5-13 until you have received all of the In Receipt quantities you want to process.

## Vendor Performance and Returns

The Vendor Performance option is only available when the Vendor Options Business Rule is configured to track Vendor Performance.

Items that are not accepted can be returned to the vendor for exchange or refund. Any discrepancies noted during the receiving can be recorded on the MSR record and used to calculate vendor performance.

Vendor Performance functionality allows you to rate a vendor's performance based on delivery time and other attributes established by your organization. Over time, this information helps you to assess a vendor's ability to deliver items or services by the promised date and in satisfactory condition.

## Using Vendor Performance and Returns

[How to Record Vendor Rating Information](#)

[How to Return Items for Exchange or Credit](#)

**How to Record Vendor Rating Information**

1. Select the appropriate Delivery ID.
2. Enter Y at the PO Item List prompt.
3. Enter N at the Pick All prompt.
4. Enter Y at the Select Item prompt.
5. Select the item you want to record a performance rating for.
6. Enter Y at the Vendor Rating Prompt.
7. Press Enter to display the list of Discrepant Attributes.

The list of Discrepant Attributes is controlled by the Vendor Performance Attributes Business Rule in the main application.

8. Select the appropriate attribute and press the Enter key.
9. Enter Y or N at the Vendor Caused prompt.

Enter Y if the discrepancy was due to vendor error and you want it to count against the vendor's performance rating.

Enter N if the discrepancy was not due to vendor error and you want the vendor to receive the maximum possible score for the delivery.

10. Enter Y at the Apply Update prompt.

The system records the attribute and adjusts the Vendor Performance rating for the delivery.

11. Enter Y at the Process Another? prompt.

The cursor returns to the Discrepant Attribute field, where you can press Enter to select another discrepant attribute for this item.

If you enter N at the Process Another prompt, the Select Item screen opens, where you can chose from several options for continuing to work with the MSR record or returning to the Main Menu.

**How to Return Items for Exchange or Credit**

1. Select the appropriate Delivery ID.
2. Enter Y at the PO Item List prompt.
3. Enter N at the Pick All prompt.
4. Enter Y at the Select Item prompt.
5. Select the item you want to return from the list of values.
6. Enter Y or N at the Vendor Rating prompt.
7. Select the type of return you want to process.

You can select either Exchange Return or Credit Return.

8. Enter Y at the Continue prompt.
9. Select a Bin if necessary.

If the stock item is in more than one bin, the system displays a list of bins for you to select from.

If you are returning a lot-controlled item, or a trackable stock item, the system will also prompt for lot and component ID information as appropriate.

**10. Enter the quantity to be returned as a negative number.**

You cannot return more than the net received quantity shown on the screen. For trackable stock items, the return quantity must be -1.

**11. Enter Y at the Apply Return prompt.**

The system adds the quantity returned to the Returned for Replacement or Returned for Credit count, and updates the To Receive and Net Receipt counts as necessary.

**12. Enter Y or N at the Vendor Caused prompt.**

Enter Y if the discrepancy causing the return was due to vendor error and you want the return to count against the vendor's performance rating.

Enter N if the discrepancy was not due to vendor error and you want the vendor to receive the maximum possible score for the delivery.

After you enter your Vendor Caused selection, the system opens the Pick Item screen where you can continue to process items on this MSR record or return to the Main Menu.

## Troubleshooting

If you have connection problems or experience unexpected results with data, verify connection settings and business rule settings.

### Fields/ Lists Not Displaying Data

If you find that you have problems connecting or experience issues with data [verify how the DSN is referenced by RFGen](#).

### Fonts Not Displaying Correctly

Check the Web Configuration Rule to see that the appropriate references are set to obtain the correct fonts. Also review the procedure for [Installing Bar Code Fonts](#) to ensure that fonts are installed correctly.

### Transfers

The Receiving Storeroom must be your default Storeroom to complete a transfer request, to receive a stock transfer, or to issue a stock transfer. If your default Storeroom is not correctly set, the system displays an authorization failure message when you try to select the request option. To resolve this issue, log off and log in again using the appropriate storeroom.



---

# Chapter 2

---

## Installation & Configuration

RF Bar Coding supports a subset of application functionality through radio frequency (RF) devices. These devices are typically IBM compatible handheld computers with bar code scanning capabilities. Such devices allow users to move freely about their work area while maintaining a real-time connection to the central Oracle Utilities Work and Asset Management system.

### Overview

Barcode label printing is only supported and certified using Loftware label printing software. Loftware allows you to create label layouts in their software and link the layouts to text file outputs from WAM/RFGEN to use to populate the labels. No barcode label printing will occur without the Loftware software being in place and both Loftware and Oracle Utilities Work and Asset Management being configured for this. Also, hardware needs to be purchased that Loftware will support to do the actual printing.

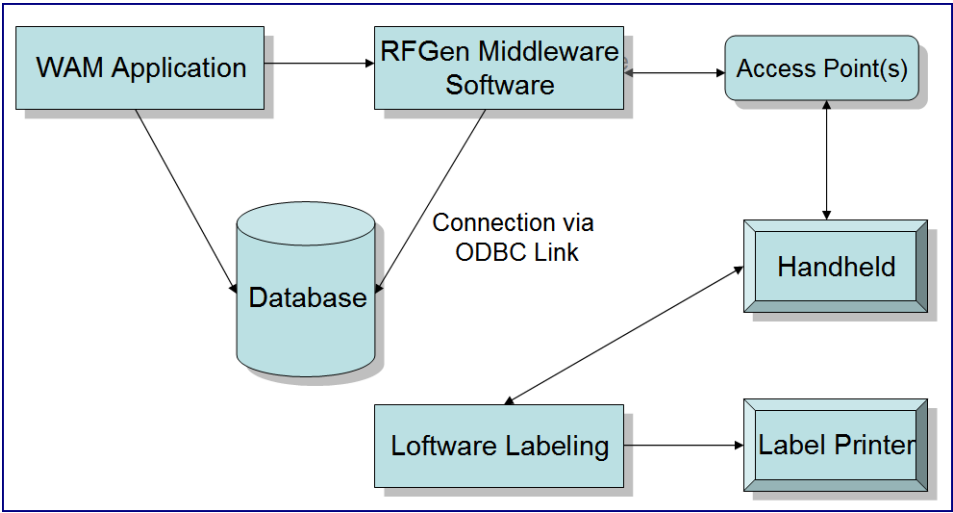
Implementing the system involves the following:

- [Installing RFGEN](#)
- [Installing Loftware](#)
- [Installing RF Bar Coding](#)
- [Configuring the ODBC Data Source](#)
- [Referencing the DSN in RFGEN](#)
- [Establishing Bar Code Users](#)
- [Configuring Bar Code Business Rules](#)
- [Code Table Configuration](#)

### General Processing

An RFGen middleware software module, the 'RFComserver' (RF Communications Server), is required to enable RF network communications. RFComserver is a Telnet Communications Server/Session Enabler that allows data collection devices to interact with your SQL database(s) in a multi-user mode.

The printing of bar code labels is provided by another third-party product named LLM-WIN32 from Loftware, Inc. LLM-WIN32 integrates a graphic label designer with enterprise-wide network printing functionality. LLM-WIN32 supports spooling and printing through Microsoft's Print Manager.



Supported Transactions

The RF Bar Coding module supports these transactions:

- Checkouts and Returns
- Stock Transfers
- Physical Inventory
- Consignment ("Keep Filled") Requests
- Reorder Revisions
- Stocking
- Receiving
- Multi-Step Receiving
- Label Printing

System Requirements

The system requires the following to function properly:

- Host Server
- Network
- Oracle Database
- Client Devices

Host Server

The following are to be considered minimum requirements for the RF Bar Coding host server.

IBM PC compatible - Intel architecture
Windows 95/98/NT/2000/XP
Not less than Pentium II w/256KB cache
Not less than 200MHz

Not less than 64MB + 4MB per expected user
The components of RF Bar Coding require only minimal disk space.
Must support the TCP/IP protocol.

## Network

Your network must support the TCP/IP protocol.

You'll need one IP address for the server and one for each handheld bar coding scanning device.

Both the Oracle Utilities Work and Asset Management database and the RF equipment must be accessible to the RF Bar Coding host computer over the network.

*Note:* The Loftware printer must have a fixed IP address as a network printer for the printing to work correctly. Bluetooth technology is not supported.

## Oracle Database

RF Bar Coding is compatible with Oracle9i Enterprise Edition Release 9.2.0.1.0 or above.

Because of ODBC limitations, however, RF Bar Coding cannot be used with earlier versions of Oracle 9i.

## Database Scripts

In order to print issue tickets from your handheld device, you must install the SRW database package shipped with the Oracle Reports product. Install the srwAPIins.sql and srwAPIgrant.sql scripts which are delivered in <Oracle Home>\reports\admin\sql. For more information on installing the SRW database package, please refer to Oracle Reports documentation which can be accessed from [http://download.oracle.com/docs/cd/B14099\\_17/bi.1012/b14048/pbr\\_evnt.htm](http://download.oracle.com/docs/cd/B14099_17/bi.1012/b14048/pbr_evnt.htm)

Until the SRW package is run, the stored database procedure SDBP\_RFGEN\_PRINT is invalid.

## Client Devices

The RFGen product used by RF Bar Coding for interfacing with client hardware purports to support a wide range of RF devices, including those from the following manufacturers:

1. Intermec
2. Symbol
3. Teklogix
4. Percon
5. Telxon
6. Hand Held Products

We make no recommendation regarding hardware or software compatibility in regard to RF Bar Coding. However, the client devices used must support:

1. TCP/IP
2. TelNet
3. VT100 or VT220 terminal emulation

Note: You need to configure the handheld to VT220 in order for the function keys F5 - F12 to work. VT100 does not support the function keys F5 - F12.

## Compatibility

RF Bar Coding is currently certified as compatible with RFGen Versions:

<b>RFGEn Version</b>	<b>WAM Version</b>
3.0.1.04	1.7.15.x
3.2.1.1	1.7.15.3.x
3.2.1.16	1.8.1.x
3.2.1.28	1.9.x
<b>Software Version</b>	<b>WAM Version</b>
9.5	All Versions

## Installing RFGEN

The RFGen CDROM version 4.0 is set to 'autoplay' when the CD is inserted into your CDROM drive. If a software loading menu does not appear, click Start, then Run, then open 'CDSetup.exe' on your CDROM drive. A menu selection is provided for each module to be loaded. Normally, the RFGen software is loaded to the "\\Program Files\\RFGen" directory.

Before loading the RFGen products, however, you must install the Oracle ODBC 9.02.00.00 or above and the Microsoft Data Access components. ODBC is used to interface with the various databases in the system, while the Data Access components provide implementation-specific database access capabilities to the RFGen products. Both of these must be installed prior to attempting the installation of the RFAssistant and RFComserver.

### How to Install RFGen

Follow these steps from the RFGen Setup screen. When installing on an NT platform, one must have 'Administrator' privileges to successfully install RFGen.

1. **Click on Install Products.**
2. **Click on RFGen Programmers Assistant.**

Choose your brand of RF hardware, and follow the on-screen instructions.

### **3. Click on RFGen Communications Server.**

Follow the on-screen instructions to install RFComserver. Select your brand of RF Hardware when prompted.

## **RFAssistant**

The RFAssistant software module requires at least one RF data form, one menu, and one RF user. The installation program creates a database installed called RFGEN.MDB, which meets these basic requirements.

## **RFComserver**

When installed, the RFComserver automatically starts as a background task when you boot your computer system.

RFComserver enables RF Bar Coding to work in a multi-user mode with your RF network. Technically, RFComserver is what is known as a 'Telnet Server.' A Network Administrator (program) selection is available on the RFGen menu to view and manage the workings of the RFComserver.

You must configure your RF devices (or controller) to telnet to RFComserver, either directly, or through a controller. Your RF devices/scanners must be set to emulate VT100 or VT220 (recommended) terminals. If you need help configuring your RF hardware, contact your hardware manufacturer or provider.

After your RF hardware is configured for Telnet communications, and when RFComserver is active, you can login to your database via Telnet from your RF device. The only software that is required to be on your remote device is TCP/IP Telnet, with a VT terminal emulator (VT100 or VT220). For more information, see the RFComserver documentation.

The RFComserver software enables the objects (forms, menus, and users) created for RF Bar Coding to work remotely, via a client-server RF network, with your selected RF devices. RFComserver relies on the fact that you have configured your RF devices (such as those offered by Intermec, Norand, UBI, Symbol, Teklogix, Handheld and Percon) as wireless telnet clients for a PC-based Win 95/98 or NT system.

The RFComserver software module must be loaded onto the same computer system that contains the RFAssistant software. You will need an authorization code to permanently install the RFComserver. Call RFComserver within seven (7) days of installation to receive your permanent activation number.

After you have loaded the RFComserver, you should reboot your computer. When your system reboots, if you selected the 'automatic start option', the RFComserver will automatically be started as a windows task. If you wish, the RFComserver can be started manually by running the RFAdministrator.

## **RFAdministrator**

The RFAdministrator allows you to see, view, and manage the sessions running under the RFComserver. The RFAdministrator Authorization menu selection is used to permanently install the RFGen Software with the number of users that have been licensed from the DataMax Software Group.

### How to Start the RFAdministrator from the Windows Desktop

1. **Select Programs from the Start menu.**
2. **Click on RFGGen.**
3. **Click on RFGGen Network Administrator.**

The RFAdministrator starts the RFComserver if it is not already running.

## Installing Software

The Loftware LLM-WIN32 family of products requires a hardware license "key" in order to print labels. The key is included in your package. For the hardware license key to be recognized on Windows NT, you must be logged in as the Systems Network Administrator when you run the setup program. After the setup is complete, you may reboot and log on as a normal user.

The license key must be installed on any LPT (Parallel) port on your PC or network server. The key must be attached to the parallel port, even if you are using a serial bar code printer. The arrow on the key points toward the PC when the key is correctly installed. If you have a laser or dot matrix printer, attach the key to the PC and then plug your Centronics parallel printer cable into the key. The key will not interfere with the normal operation of the printer. For more information about installing the key, visit Loftware's website at: <http://www.loftwareinc.com/> or [www.llmwin.com](http://www.llmwin.com).

All programs are installed by the SETUP.EXE program that starts automatically when the Loftware CD is inserted. If you wish to do a "quick install" of the ActiveX Control, WatchDog-32, WatchDog-NT or the core LLM-WIN32 product, you can also run the SETUP.EXE programs from their respective directories on the CD.

Choose a local installation if you intend to use the LLM-WIN32 product on an individual PC. All application files and configurations will also be stored on this PC. Even if you choose a local install, you may choose to share your labels and data with other PC's on your network. Select Options / File Locations from the Loftware menu for setting default shared directories. Local installs are the easiest way to get LLM-WIN32 up and running in the shortest period of time.

Before performing any install, make sure your hardware key is plugged into your parallel port as described in the previous section. Setup will detect the key and make the appropriate settings in the software. If necessary, it is also possible to set the hardware key after installing as described in the Loftware manual.

### How to Install Software

1. **Place the Software CD in your CD-ROM Drive.**  
Autorun launches the setup program. If autorun fails to launch, click on SETUP.EXE in the root directory of the CD-ROM.
2. **Click on Install Products.**
3. **Click on Install LLM-WIN32 to launch the core setup program.**
4. **Re-boot your computer when prompted.**

All files are installed to the Program Files\LLMWIN32 and Windows\System folder by default. The programs can run "stand alone" with no server involved. Even though this is a local install, file locations can be set on each PC for sharing serial numbers, labels, databases and images with other PCs. You may change file locations in design mode by choosing Options / File Locations.

## Configuring Printers

Before you can print labels, you must configure a printer and assign it to one of the available print queues. You configure printers and set the active printer(s) using the Configure Printers dialog box. To open the Configure Printers dialog box

- Select Printers from the Software file menu, or
- Click on the Printers button from the Test Print window.

### Notes

Always use Software drivers when available. Only use third party drivers when Software does not have a native driver for your printer.

The Software printer must have a fixed IP address as a network printer for the printing to work correctly. Bluetooth technology is not supported.

The Configure Printers dialog box lists both Available Printers and Configured Printers.

Available Printers are printers that have installed printer drivers. This list includes any thermal/thermal transfer bar code printers and the Software Windows printer installed by LLM-WIN32. If you have added Generic Text Drivers for Spooling through print manager, they are also shown.

Configured Printers are printers that have been assigned a connection (port, baud rate, parity, etc.) and have been setup for options which are unique to the selected printer. It also includes Windows printers and third-party drivers available from your Control Panel. A configured printer is a totally defined printer, ready to be used for printing. You can have any number of configured printers as you want.

### How to Configure a Printer

1. Select the printer to configure from the list of available printers.
2. Click the Add New Printer button.
3. Define the printer connection, specifying the following options.

**Printer Alias** - You may optionally name your printer here. For instance, you may use a name like "Warehouse Printer 1" instead of the default name that would be something like: "Printer 1: Intermec 3400B on COM1". This is especially useful if you are using WatchDog-NT with the status tree view enabled.

**Port** - Choosing a specific port limits your printing capability to those printers attached locally to the server itself. Typically, you will want to make the label printing capability distributed among several printers throughout the warehouse, or to printers in accessible areas (leaving the server in a secure area). However, if you expect to use only one printer and you expect to place the server in an accessible area (not advised), you may select a specific printer port.

You can choose a parallel port (LPTx), or a serial port (COMx). The available list of ports will depend on your PC's hardware and installed drivers. Consult the Software manual or your printer documentation for printer setting information.

**Print Manager/Spooling** - It is recommended that you use the Windows Print Manager to make remote printing available. Print Manager/Spooling sends print jobs to local or networked printers using the window's Print Manager to spool files to the port. Enabling Print Manager/Spooling instead of printer Port allows addressing shared printers across the network and fast job handoff since Print Manager queues the requests and the label printing software becomes immediately available to service the next request.

#### 4. Run Printer Setup for the selected printer.

In addition to a printer model and connection information, a configured printer also contains printer setup information. This information is unique to the printer you have selected. These settings apply to all labels designed for this configured printer.

### How to Set a Configured Printer as an Active Printer

After you have configured one or more printers, you need to tell LLM-WIN32 which printers you want to use as the active printers. You can have up to four active printers defined at a time with LLM-WIN32.

#### 1. Open the active printer list.

You do this by selecting Printers from the Software file menu.

#### 2. Choose the current active printer if you want to change active printers.

#### 3. Choose a configured printer you wish to make active.

When you do, the button in the Configured Printers area changes to "Assign to printer n" where n is the printer you selected.

If the printer has not been configured, you must configure it first. Refer to the instructions at the beginning of this section for information on configuring printers before assigning them.

#### 4. Click on the Assign to printer n button.

The name of the selected printer now appears in the active printer list.

### How to Test a Printer

It is advisable to test the configuration by printing to the printer just configured. You can use the sample labels Software supplies for this purpose.

#### 1. Double-click on the LLM-WIN32 / Label Design icon.

#### 2. Open one of the provided Label files (\*.lwl).

#### 3. Click on the Test Print icon in the standard toolbar.

#### 4. Choose which of the assigned printers you want to use.

If the program displays a "Demo Mode" warning message, the key has not been installed properly. This is okay if you intend to run the program in demo mode, otherwise review the procedure for attaching the key. For additional information on installing local and network keys go to Loftware's Internet Home Page: [www.loftwareinc.com](http://www.loftwareinc.com) or [www.llmwin.com](http://www.llmwin.com).

## Installing Bar Code Fonts

Conduct an internet search on "free barcode font" to download a font set to use with the Oracle Utilities Work and Asset Management bar coding functionality. Install the font on your application server and on the client machines then set business rules so that fonts are used on barcode reports.

*Note:* You must install the fonts on both the application server and on the client machine.

### How to Install Fonts in the Windows Operating System

To install the fonts, do NOT simply copy them to the C:\Windows\Fonts directory. Even though the fonts appear to install correctly, they may not be visible on the barcode type reports.



1. **Open the Control Panel (Start> Settings> Control Panel).**
2. **Select Fonts.**
3. **Select File -> Install New Font.**

The “Add Fonts” dialog box opens.

4. **Find the location where you stored the downloaded barcode fonts.**
5. **Make sure the “Copy fonts to Fonts folder” check box is checked.**
6. **While holding down the Ctrl key, select all of the fonts.**
7. **Click the OK button.**

The font installation is complete.

8. **Restart if necessary.**

### **Setting the system to use the font on Barcode Reports**

Once the fonts are properly installed configure the following business rule settings:

**Web Configuration Business Rule** - Enter the name of the installed font in the Setting field for the REPORTS BARCODE FONT rule key.

**Default Directories Business Rule** - Enter the BAR CODE LABEL DROP DIRECTORY rule key to indicate the server where the barcode files are sent.

## **Installing RF Bar Coding**

The RF Bar Coding installation consists of a Server Installation and a Client installation.

### **Database Server Installation**

The database objects for Oracle Utilities Work and Asset Management must be installed in your database before you install RF Bar Coding.

1. **Copy the files from the Release\Server folder to any directory on the server.**
2. **Run the script BARCODE40.SQL at the SQL\*Plus prompt.**

```
SQL> @barcode40.sql
```

This script can be executed for both upgrades and first time installations. This script will install all Oracle Utilities Work and Asset Management Bar Code scripts to Version 4.0. It creates the output file barcode40.log which you should review for errors.

3. **When prompted, enter the name of the person performing the installation.**

This is retained for auditing purposes.

4. **Recompile all objects until there are no invalid objects.**

```
SQL> @recompil.sql
```

### **Client Installation**

1. **Shutdown any RFGGen Services and servers.**
2. **Locate the folder where the RFGGen program is installed.**

Normally RFGGen is installed at C:\Program Files\RFGGen.

3. **Rename the existing RFGGen.mdb file to RFGGen.mdb.old or similar.**

4. **Copy the file RFGen.mdb from the Release\Rfgen folder to the RFGen folder on the client.**

Start all RFGen Services and Servers.

## Configuring the ODBC Data Source

RFGen requires an ODBC Data Source Name (DSN) in order to successfully connect to your Oracle database.

Before configuring an ODBC DSN for the Oracle Utilities Work and Asset Management database, you must install the Oracle client and networking components. Of particular importance are the tnsnames.ora, sqlnet.ora, listener.ora and “sqlnet” files. Refer to your Oracle documentation for instructions on installing these Oracle products.

### How to Configure the ODBC Driver

1. **Click on the ODBC Data Sources icon.**

You can find The ODBC driver icon in the Windows Control Panel or the Administrative Tools folder. When you click on the icon, the ODBC Data Source Administrator window opens.

2. **Click on the System DSN tab.**
3. **Click on the Add button to create a new Data Source.**
4. **Choose the Oracle ODBC driver.**

The Oracle ODBC driver Version 9.02.00.00 (or higher) is required for RFGen Network Administrator 3.0.1.04.

5. **Click the Finish button.**

The setup screen opens where you provide following values:

**Data Source Name (DSN)** - Enter any name meaningful to your organization. You will need to refer to this name when you configure RFGen to connect to this data source.

**Description** - Enter descriptive text to describe the data source.

**User Name** - the User Name must be a defined and authorized user of the Oracle Utilities Work and Asset Management database. The Oracle DBA will normally establish an account for the RF Bar Coding module and the user id thus created would be supplied here.

**Server** - The Server must be the name of the database.

## Referencing the DSN in RFGEN

Once the ODBC database connection parameters are setup, you'll need to run RFGen's RFAdmin.exe program in order to reference the new DSN.

### How to Reference the DSN

1. **Locate RFAdmin.exe, and run the program.**
1. **From the Configuration menu option, select Database Connection.**
2. **Enter the Name/ID, Data Source, and Database.**

The name and Data Source refer to the DNS that you created in the previous section.

### 3. Set the Connect Using parameter to ADO.

The default is ADO (ActiveX Data Objects). If you find that you have problems connecting or experience issues with data, set this value to RDO (Remote Data Objects).

RDO is mostly used for connections to SQL servers. While can be used to access a wider variety of data sources.

### 4. Select a provider.

### 5. Enter the User ID and Password.

These are the same credentials used to log on to the Oracle Utilities Work and Asset Management system.

### 6. Click Save.

## Establishing Bar Code Users

Privileges for bar coding users must be established in Oracle Utilities Work and Asset Management using the User Profile module. To access the User Profile module, select System Configuration from the Administration Menu and click on User Profile.

### How to Add Bar Coding Privileges for a User

1. Open the user profile for the appropriate user.
2. Click on an empty line in the Key Name column or click on an existing Key Name value and click on the Insert icon.
3. Type RFGEN in the Key Name column.
4. Click on the blank space under Key Value and just to the right of RFGEN.
5. Enter the default Storeroom ID for this user.
6. Enter any additional storerooms in the adjacent Key Value columns to the right.

Users can have more than one default storeroom defined in their user profile, but can only work in one storeroom at a time. All work done during a given bar coding session is restricted to the Storeroom selected at logon. If the user wants to work on records in another Storeroom, you must logoff and begin a new session to select the new Storeroom.

### 7. Click on the Save Icon.

The system saves the User Profile with the Bar-coding privileges you have established.

## Responsibility Setting

List the Bar Code Print Label function in the Responsibilities of users who will need to access this functionality from the Multi-Step Receiving module.

## Configuring Bar Code Business Rules

Much of the RF Bar Coding functionality is controlled by business rule settings. To configure these rules, set the key values as discussed below for the following rules:

[Bar Coding Checkout Rule](#)  
[Bar Code Configuration Rule](#)  
[Bar Code Report Module Rule](#)  
[Bar Code Reports Gateway Rule](#)  
[Checkout Allowable Charge Type Rule](#)  
[Default Directories Rule](#)

### Bar Coding Checkout Rule

The Bar Coding Checkout rule determines whether the temporary upload table is emptied prior to next upload of checkout/return data from the handheld device, or if new entries are appended to any prior error entries. Enter ON in the “OPTION STATUS” column to allow records with errors to be kept in a temporary table. Enter OFF in the “OPTION STATUS” column to remove the remaining records from the temporary table prior to the next upload.

### Bar Code Configuration Rule

The Bar Code Configuration rule determines which options appear on the bar code main menu. Options set to ON appear on the handheld device.

### Bar Code Report Module Rule

The Bar Code Report Module rule determines which report number is to be executed on each of the Printing sub menu items in the handheld device.

### Bar Code Reports Gateway Rule

The Bar Code Reports Gateway rule determines the database connection to be used when generating a report from the handheld device. The SID must be the same as the connection name set in the device.

### Checkout Allowable Charge Type Rule

This rule is used to control the charge types that are allowable for Stock Checkouts. A setting of ALLOW for a Charge Type makes it available on the checkout screen. A setting of DISALLOW will exclude the Charge Type from the checkout screen.

### Default Directories Rule

The setting of the Bar Code Label Drop Directory rule key determines where the queued files are stored when a user prints labels. Enter a valid drive or network location.

### RFGen Bar Coding Setup Rule

Select options in the RFGen Bar Coding Setup rule to configure RF Bar Coding to conform to your organization's business practices.

KEY Name	Where Used	Description
<b>Adjust Physical Inventory</b>	Physical Inventory	Determines if stores adjustments processing is allowed from the handheld device during physical inventory when. Options are YES or No.
<b>Display Expense Codes</b>	Checkouts / Returns	Determines if stock expense codes are visible on the handheld device during stock checkouts/returns. Options are YES or No.
<b>Hide Inventory Quantity</b>	Physical Inventory	Determines if the actual inventory quantity is hidden during physical inventory. Options are YES or NO.

KEY Name	Where Used	Description
<b>Old Stock Code Field</b>	System-Wide	The name of the field (UDF) in SA_CATALOG table where old stock codes are listed. Options are any UDF from ATTRIBUTE1 to ATTRIBUTE10.
<b>Old Stock Code Processing</b>	System-Wide	Indicates if characters are to be stripped while reading the old stock code with the handheld device. Options L (strip characters from the left) or R (strip characters from the right), followed by the number of characters to strip, or NONE.
<b>Override Employee Error</b>	Checkouts / Returns	Indicates if transactions can proceed even if an incorrect employee number is entered for receiving employee during checkouts/returns. Issuing employee number/user name is used whenever needed instead. Options are YES or NO.
<b>Override Expense Codes</b>	Checkouts / Returns	Determines if users are allowed to change default storeroom stock expense codes during checkouts/returns. Options are YES or NO.
<b>Stock Code Order By</b>	Checkouts / Returns	Defines the sort order for the list of values on the Stock Code field in the Stock Checkout detail. The options are STOCKCODE and BIN. If the option is set to STOCKCODE then the list is displayed by stock code in ascending order. If the option is set to BIN then the list is displayed in ascending order by primary bin location as defined in the storeroom for the stock code.
<b>Use Old Stock Codes</b>	System-Wide	Enable the use of old stock codes to facilitate transition to the new system implemented within Oracle Utilities Work and Asset Management. Options are YES or NO.

### Web Configuration Rule

**REPORTS BARCODE FONT** - Enter a value in the setting field for the Reports Barcode Font rule key to specify the font to be used on reports that include a barcode field. The font that is entered should be available in your system fonts. You can find fonts to use by searching the internet for “Free Barcode Fonts.”

## Code Table Configuration

Define printer labels by entering values in code table 80.

---

---

# Cue Cards

## Basic Operations

- How to Log On
- How to Log Off
- How to Initiate a Checkout/Return session
- How to Checkout a Stock Item against a Work Order Task
- How to Return a Stock Item against a Work Order Task
- How to Checkout a Stock Item against an Asset
- How to Checkout a Stock Item against a Checkout Request
- How to Create a Stock Transfer
- How to Add Stock Items to a Transfer Request
- How to Issue a Stock Transfer
- How to Process a Physical Inventory Record
- How to Process a Keep Fill Request or a Reorder Review Request
- How to Print Issue Tickets
- How to Print Labels
- How to Print Receiving Reports
- How to Add Items to the Storeroom
- How to Receive Stock Items against a Purchase Order
- How to Create a MSR Delivery Record
- How to Record Delivery Information
- How to Place Items in Receipt
- How to Receive Items
- How to Record Vendor Rating Information
- How to Return Items for Exchange or Credit

## Installation & Configuration

- How to Install RFGen
- How to Start the RFAdministrator from the Windows Desktop
- How to Install Software
- How to Configure a Printer
- How to Set a Configured Printer as an Active Printer
- How to Test a Printer
- How to Install Fonts in the Windows Operating System
- Database Server Installation
- Client Installation
- How to Configure the ODBC Driver
- How to Reference the DSN
- How to Add Bar Coding Privileges for a User

---

---

# Index

- B
  - Bar Code Fonts 32
  - Business Rules
    - configuring 35
- C
  - Checkouts 8
    - account 10
    - asset 10
    - transactions 11
    - work order task 8
  - Client Devices 28
  - Compatibility 28
- D
  - Data
    - entering 6
  - Deliveries
    - recording 19
  - Devices 28
- F
  - Fonts 32
  - Function Keys 7
- H
  - Hand-Held Device 6
  - Host Server 26
- L
  - Lists 6
  - Software
    - installing 30
  - Logging On & Off 7
- M
  - Multi-Step Receiving 18
- N
  - Network 27
- O
  - ODBC Data Source
    - configuring 34
- Oracle Database 27
- P
  - Physical Inventory 13
  - Printing
    - configuring printers 31
    - Issue Ticket 15
    - labels 15
    - Receiving Report 16
- R
  - Receiving 17
  - Reorder Handling 14
  - Returns 8
    - account 10
    - asset 10
    - transactions 11
    - vendor performance 22
    - work order task 9
  - RF Bar Coding
    - installing 33
  - RFAAdministrator 29
  - RFAssistant 29
  - RFComserver 29
  - RFGEN
    - installing 28
- S
  - Stock Transfer 11
    - receiving 13
  - Stocking 16
  - Storeroom 7
    - receiving 21
  - System Requirements 26
- T
  - Transactions 26
- U
  - Users 35
- V
  - Vendor Performance 22
- Vendor Performance and Returns 22
- W
  - Work Order Task 6
    - return 9



# **Oracle® Utilities Work and Asset Management**

Business Intelligence Guide

Release 1.9.0.3

July 2011

Oracle® Utilities Work and Asset Management Business Intelligence Guide for Release 1.9.0.3

Doc v1 rev.0 7/7/11

Copyright ©1986, 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: TUGBU Documentation

Contributing Author: n/a

Contributor: n/a

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any lost, costs, or damages incurred due to your access to or use of third party content, products or services.

---

---

# Business Intelligence

Oracle Utilities Business Intelligence (BI) for Oracle Utilities Work and Asset Management uses extraction programs packaged with the Oracle Utilities Application Framework (OUAF). The extraction programs for Oracle Utilities Work and Asset Management are written in PL/SQL and are loaded into BI using OUAF.

Users can view batch control records, submit batch jobs and view the batch run tree using the Oracle Utilities Work & Asset Management application.

This section describes the configurations required in Oracle Utilities Work and Asset Management only. For complete Oracle Utilities Business Intelligence implementation and configuration guidelines refer to the online documentation within BI.

Note: It is recommended that you read this entire document before beginning installation.

## Certified Version

Certified versions of Business Intelligence include:

- Business Intelligence v2.2.1 and higher for Oracle Utilities Work and Asset Management releases 1.8 and higher.
- Business Intelligence v2.3.0 for Oracle Utilities Work and Asset Management releases 1.9.0.2 and above.

## Extract Mapping

Please refer to the printed User Guide for extract mapping information.

## Topics

[Basic Steps to Get Started](#)  
[Multiple Plants](#)  
[Extract Output](#)  
[Batch Procedure](#)  
[Business Rules](#)  
[Sequence Numbering](#)  
[Initial Load and Incremental Load](#)  
[Error Handling](#)  
[File Name Conventions and Outputs](#)  
[SABI Triggers](#)  
[Extracts](#)

## Basic Steps to Get Started

1. Enable the Oracle Utilities Work and Asset Management BI database triggers using Installation Parameters.
2. Setup the BI Batch procedure, SDBP\_BI\_EXTRACT\_HANDLER, in the Job Manager module.
3. Configure the OUBI Setup Business Rule.

- 
4. Verify sequence numbering for SA\_CHANGE\_LOG in the Sequence Numbering module.
  5. Run the batch procedure, SDBP\_BI\_EXTRACT\_HANDLER.

## Multiple Plants

If you have multiple plants all configuration settings must be configured and executed separately for each.

## Extract Output

A folder for the extraction file output is required to be created on the database server using the database server operating system. This directory is referred to as the “BI Extract output folder”. A database object called a “Directory” needs to be created to point to this folder on the database server file system.

### Default

The installation includes a pre-configured database directory object that is defaulted to “c:\bi\_extract”. This serves as a target folder for extracted files. SD\_BI\_OUTBOUND is the directory object in the database that maps to the output directory folder for the BI extracts.

### Override the Default

This default can be overridden at the database level by running the following script:

```
CREATE OR REPLACE DIRECTORY sd_bi_outbound AS '<operating system folder
name>'
/
```

### Steps to Create the Database Object

1. The Directory object must be created by a user with DBA privileges.
2. Create the directory object referencing your output folder.  
To [override the default](#), run the script indicated above.
3. Grant privileges to the Directory to the user that will be executing the extract routines. This is the same user that will be creating the objects in the Oracle Utilities Work and Asset Management Job Scheduler.

```
grant read on directory SD_BI_OUTBOUND to synergen;
```

```
grant write on directory SD_BI_OUTBOUND to synergen;
```

## Batch Procedure

The batch procedure, SDBP\_BI\_EXTRACT\_HANDLER, handles and calls all of the extract programs. You can monitor this processing in the Job Manager and the Job Manager Log modules. SDBP\_BI\_EXTRACT\_HANDLER also handles incrementing the value of the sequence number whenever an incremental extract is requested.

## Business Rules

### OUBI Setup Rule

Configure settings to determine which extracts will be completed during regular processing. This rule can also be used to define extraction details such as data source, file names and the output directory. If desired, any extract can be turned OFF so that it is not included in the overall extraction.

#### *Common Extract Parameters*

The first rule key, COMMON\_EXTRACT\_PARAMETERS, defines parameters for all of the extracts that are turned ON. The key values for this rule key differ slightly from the key values for the remaining rule keys.

**Value** is left blank for this rule key.

---

**Value2** is the name of the directory object in the database. This corresponds to the output folder of the extract, SD\_BI\_OUTBOUND.

**Value3** is the Data Source Indicator. Default value is '3'

**Value4** controls which type of extract will be executed. A value of INI means the extract will produce an Initial Extract, an INC means an Incremental Extract. The default value is INI.

After an Initial Extract, the value is automatically updated to INC by SDBP\_BI\_EXTRACT\_HANDLER so that the succeeding extracts are incremental.

#### *General Key Names and Values*

**Value** controls whether the extract is executed. Default value is ON.

**Value2** is the base extract file name. This is the extract identifier used in building the names of the extract files.

**Value3 and Value4** only apply to the Key Name OUBI INV SNAPSHOT. The goal of this metric is to compare inventory levels of like stock items based on periodic snapshots of the quantities in the storeroom. This is not a metric used to compare inventory movement because it does not deal with the transaction volume.

This requires two parameters to be passed to the procedures:

**Value3** - Type of Increment - Monthly or Daily

**Value4** - Number of Historical Snapshots - Any number

### **Sequence Numbering**

Sequence Numbering must be turned on for SA\_CHANGE\_LOG to control the batch number field of the SA\_CHANGE\_LOG table. The Oracle Utilities Business Intelligence triggers use the value of this sequence number to uniquely identify records inserted into the SA\_CHANGE\_LOG table.

### **Initial Load and Incremental Load**

Settings for the OUBI Setup Business Rule COMMON EXTRACT PARAMETERS rule key determine how the data is initially loaded as well as the incremental data loads.

#### **Initial Load**

The Initial Load of the extracts will query ALL records from the target tables and sends the entire set of source data to BI. This load is filtered by PLANT. There is no additional filtering applied to the source information. For example, the Account Transactions Extract targets the SA\_ACCOUNT\_LOG table as the source information. The initial load will take all records in the SA\_ACCOUNT\_LOG table and send the records to BI.

If you need to extract records from another plant, the batch job, SDBP\_BI\_EXTRACT\_HANDLER, must be for the other PLANT and executed. Same with the Incremental extracts.

#### **Incremental Load**

The Incremental Load is based on information found in the SA\_CHANGE\_LOG table.

Records are inserted into this table from a set of BI extract database triggers placed on various tables in Oracle Utilities Work and Asset Management. The incremental extracts begin to have data in the files after the BI triggers have been enabled in the Oracle Utilities Work and Asset Management database schema. If a record is recorded in the SA\_CHANGE\_LOG, a transaction is set to BI.

### **Error Handling**

Errors will not occur unless there is a problem with the extracted data or with configuration settings. If errors are encountered, they will be noted in the Job Manager Log module. Correct any errors then re-run an Initial Load.

---

### To Rerun an Initial Load

1. **Open the OUBI Setup Business Rule.**
2. **Set the Common Extract Parameters rule key to 'INI.'**
3. **Execute the batch job, SDBP\_BI\_EXTRACT\_HANDLER.**

After an Initial Extract, the business rule value is automatically updated to INC by SDBP\_BI\_EXTRACT\_HANDLER so that the succeeding extracts are incremental.

### File Name Conventions and Outputs

File Names follow the standard naming convention expected by OUBI. The OUBI Setup Business Rule handles the prefix file name for all the extracts. This is the value under the Value2 column. These filenames cannot be modified. The structure is indicated here to provide insight into how information can be determined from the file name.

Example: EXTDASSE0000030000000021001.<extension name>

- Extract Identifier: No limit on the size
- Data Source Indicator: Next 6 characters
- Batch Number: Next 10 characters
- Thread Number: Next 3 characters

Each extract produces two files: \*.DAT and \*.CTL.

- **DAT** files contain the actual extract data from the Oracle Utilities Work and Asset Management tables.
- **CTL** files are used by Oracle Utilities BI for log and upload trace details. The file contains one line of information which consists of:
  - Batch Code: First 8 characters
  - Data Source Indicator: Next 6 characters
  - Batch Number: Next 10 characters
  - Thread Number: Next 10 characters
  - Line Count (on the DAT file): Next 10 characters
  - Batch Code Description: Next 60 characters

### SABI Triggers

SABI triggers insert data into the SA\_CHANGE\_LOG table in preparation for extraction. The triggers use sequence number settings to determine the next batch number for the extraction.

The following triggers are used:

bi_adiu_acct_data	bi_adiu_actg_log
bi_adiu_actperdates	bi_adiu_asset
bi_adiu_assetdown	bi_adiu_auth_stt
bi_adiu_catalog	bi_adiu_dept
bi_adiu_dept_oa	bi_adiu_failure
bi_adiu_op_actg_tr_ty	bi_adiu_ouom
bi_adiu_repair	bi_adiu_rootcause
bi_adiu_rule_key_oex	bi_adiu_rulekey_stt
bi_adiu_storeroom	bi_adiu_strm_setup
bi_adiu_strm_trans	bi_adiu_tracwrkordtcr
bi_adiu_tracwrkordtpl	bi_adiu_wrkord

---

bi_adiu_acct_data	bi_adiu_actg_log
bi_adiu_wrkordadjust	bi_adiu_wrkordlbr
bi_adiu_wrkordmat	bi_adiu_wrkordrequire
bi_adiu_wrkordservcont	bi_adiu_wrkordtask
bi_adiu_wrkordtfail	bi_adiu_wrkordty

## Extracts

The following section maps the extract program to the data that it handles as well as the view, batch procedure and trigger that are used to complete the extraction.

Please refer to the extract mapping document titled, WAM BI Extract Mapping for more information.

## To Review the Columns that are Extracted

Access the database and describe the definition of the view. From there you can see the specific columns that comprise the view.

For example, in SQL Plus:

```
SELECT
*
FROM SYNERGEN.SV_BI_EXTRACT_EXTDFAIL Vw
```

# **Oracle® Utilities Work and Asset Management**

Interfaces Guide

Release 1.9.0.3

July 2011



Oracle® Utilities Work and Asset Management Interfaces Guide for Release 1.9.0.3

Doc v1 rev.0 7/12/11

Copyright ©1986, 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: TUGBU Documentation

Contributing Author: n/a

Contributor: n/a

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any lost, costs, or damages incurred due to your access to or use of third party content, products or services.

# Contents

<b>1 Introduction .....</b>	<b>8</b>
Interfaces .....	8
Basic Batch Job Procedure Syntax .....	17
Parameters .....	17
Processing .....	18
Pre- and Post-Processing Routines .....	18
Utility Functions .....	18
wifp_log_job_mgr_job .....	19
wifp_log_job_mgr_log_message .....	19
wifp_generate_key_interface .....	20
<b>2 Account .....</b>	<b>21</b>
Batch Job Procedure .....	21
Keywords and Parameters .....	22
Business Rule Format .....	23
Interface Table Layout .....	23
WAIF_ACCOUNT .....	23
<b>3 Accrual .....</b>	<b>26</b>
Batch Job Procedure .....	26
Keywords and Parameters .....	26
Business Rule Format .....	27
Process Flow .....	27
Interface Table Layout .....	28
WAIF_ACCRUAL .....	28
<b>4 Alert .....</b>	<b>30</b>
Batch Job Procedure .....	30
Keywords and Parameters .....	30
Interface Table Layout .....	31
WAIF_ALERT .....	31
<b>5 Asset .....</b>	<b>32</b>
Batch Job Procedure .....	32
Keywords and Parameters .....	32
Business Rule Format .....	33
Sequence Numbers .....	33
Interface Table Layout .....	33
WAIF_ASSET .....	33
Asset Web Services .....	38
Web Service Methods .....	38
Web Service Security .....	38

Web Service URL .....	38
Web Service Definition Language (WSDL) .....	38
Asset Specification Web Services.....	45
Web Services Methods.....	45
Web Service Security .....	46
Web Service URL .....	46
Specification Tables Data Type.....	46
Specification Tables Detail Data Type.....	47
Web Service Definition Language (WSDL) .....	47
GIS Interfaces for Assets and Specifications .....	53
<b>6 Blanket Contract .....</b>	<b>56</b>
Batch Job Procedure.....	56
Keywords and Parameters.....	56
Sequence Numbers .....	57
Interface Table Layout .....	57
<b>7 Budget.....</b>	<b>62</b>
Batch Job Procedure.....	62
Keywords and Parameters.....	62
Interface Table Layout .....	63
<b>8 Catalog .....</b>	<b>65</b>
Batch Job Procedure.....	65
Keywords and Parameters.....	66
Business Rule Format.....	66
Sequence Numbers .....	67
Interface Table Layout .....	67
WAIF_CATALOG (same fields as SA_CATALOG) .....	67
<b>9 Cost Adjustments .....</b>	<b>70</b>
Batch Job Procedure.....	70
Keywords and Parameters.....	70
Sequence Numbers .....	71
Interface Table Layout .....	71
WAIF_COST_ADJUSTMENT .....	71
<b>10 Customer .....</b>	<b>74</b>
Batch Job Procedure.....	74
Keywords and Parameters.....	74
Customer Interface Rule.....	75
Customer Address Interface Rule .....	75
Customer SA Interface Rule .....	76
Sequence Numbers .....	76
Interface Table Layout .....	76
WAIF_CUSTOMER .....	76
<b>11 Direct Charges.....</b>	<b>79</b>
Batch Job Procedure.....	79
Keywords and Parameters.....	79
Sequence Numbers .....	80
Interface Table Layout .....	80
WAIF_DIRECT_CHARGE .....	80

<b>12 Employee</b>	<b>83</b>
Batch Job Procedure	83
Keywords and Parameters	83
Business Rule Format	84
Sequence Numbers	85
Interface Table Layout	85
WAIF_EMPLOYEE	85
<b>13 Employee Wage Rate</b>	<b>88</b>
Batch Job Procedure	88
Keywords and Parameters	88
Interface Table Layout	89
WAIF_EMPLOYEE_WAGE_RATE	89
<b>14 Fuel</b>	<b>91</b>
Batch Job Procedure	91
Keywords and Parameters	92
Business Rule Format	92
Interface Table Layout	92
WAIF_FUEL_SYSTEM_DATA	92
<b>15 General Ledger</b>	<b>95</b>
Batch Job Procedure	95
Keywords and Parameters	96
Business Rule Format	96
Transaction Types	97
Interface Table Layout	98
WAIF_GL_TRANSACTION	98
<b>16 GIS Interfaces</b>	<b>102</b>
<b>17 Inventory Log</b>	<b>106</b>
Batch Job Procedure	106
Keywords and Parameters	106
Business Rule Format	107
Sequence Numbers	107
Interface Table Layout	107
WAIF_INVENTORY_LOG	107
<b>18 Invoice AP Data</b>	<b>110</b>
Batch Job Procedure	110
Keywords and Parameters	110
Interface Table Layout	111
WAIF_INVOICE_AP_DATA	111
<b>19 Invoice</b>	<b>112</b>
Batch Job Procedure	112
Keywords and Parameters	112
Business Rule Format	113
Interface Table Layout	113
WAIF_INVOICE	113
<b>20 Manufacturer Vendor</b>	<b>119</b>

Batch Job Procedure.....	119
Keywords and Parameters.....	119
Business Rule Format.....	120
Interface Table Layout .....	121
WAIF_CATALOG_MFR_VENDOR .....	121
<b>21 Meter Reading .....</b>	<b>123</b>
Batch Job Procedure.....	123
Keywords and Parameters.....	124
Interface Table Layout .....	124
WAIF_METER_READINGS_INTERFACE .....	124
<b>22 Purchase Order .....</b>	<b>126</b>
Batch Job Procedure.....	126
Keywords and Parameters.....	127
Business Rule Format.....	127
Sequence Numbers .....	127
Interface Table Layout .....	128
<b>23 Receipt .....</b>	<b>135</b>
Batch Job Procedure.....	135
Keywords and Parameters.....	135
Sequence Numbers .....	136
Interface Table Layout .....	136
WAIF_RECEIPT .....	136
<b>24 Requisition.....</b>	<b>138</b>
Batch Job Procedure.....	138
Keywords and Parameters.....	138
Sequence Numbers .....	139
Interface Table Layout .....	139
WAIF_REQUISITION .....	139
<b>25 Service Request .....</b>	<b>145</b>
Batch Job Procedure.....	145
Keywords and Parameters.....	145
Business Rule Format.....	146
Sequence Numbers .....	146
Interface Table Layout .....	146
WAIF_SERVICE REQUEST .....	146
<b>26 Storeroom .....</b>	<b>152</b>
Batch Job Procedure.....	152
Keywords and Parameters.....	153
Business Rule Format.....	153
Interface Table Layout .....	154
WAIF_STOREROOM .....	154
<b>27 Timekeeping .....</b>	<b>158</b>
Batch Job Procedure.....	158
Keywords and Parameters.....	159

Sequence Numbers .....	159
Interface Table Layout .....	159
WAIF_TIMEKEEPING .....	159
<b>28 Vendor.....</b>	<b>163</b>
Batch Job Procedure.....	163
Keywords and Parameters.....	164
Business Rule Format.....	164
Sequence Numbers .....	165
Interface Table Layout .....	165
WAIF_VENDOR .....	165
<b>29 Web Services.....</b>	<b>168</b>
Web Service Components .....	168
Design Element – Service Provider .....	171
Design Element – Client.....	173
Security .....	173
Architecture Overview .....	174
Exposing Business Logic via Web Services.....	175
Consuming Web Services Provided by Other Applications.....	175

---

# Chapter 1

## Introduction

Oracle Utilities Work and Asset Management Standard Connector Library is intended to provide a starting point for interfacing with other applications. The interfaces perform the common interface task that Oracle Corporation has deemed to be reusable, while providing “hooks” to allow custom processing to fit the individual client’s needs. Each interface consists of an interface table and a PL/SQL stored procedure that processes the interface table. Some interfaces are bi-directional in that they can move data in or out of the Oracle Utilities Work and Asset Management application, while others will only move data one way. Business rule settings are required for some of the interfaces, but not all.

*Note:* Please refer to the release notes for the applicable product version for certification information.

[Interfaces](#)

[Basic Batch Job Procedure Syntax](#)

[Parameters](#)

[Processing](#)

[Utility Functions](#)

## Interfaces

The following tables provide a brief overview of the Oracle Utilities Work and Asset Management interfaces currently available.

Interface Name	Account
Direction	IN/OUT
Description	This interface sends or receives account data from other systems. Expense codes can also be processed by this interface. The OPTIONS parameter allows control how data is processed.
Syntax	WIFP_ACCOUNT_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_ACCOUNT
Business Rule	ACCOUNT INTERFACE
Sequence Number	N/A

Interface Name	Accrual
Direction	OUT

<b>Description</b>	Non-invoiced receipts of requested PO line types are gathered and written to the interface table. These accruals are taken at a “point-in-time”.
<b>Syntax</b>	WIFP_ACCRUAL_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_ACCRUAL
<b>Business Rule</b>	ACCRUAL INTERFACE
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Alert</b>
<b>Direction</b>	IN
<b>Description</b>	This interface allows other systems to send alerts to Oracle Utilities Work and Asset Management users. These alerts behave just like on-line alerts with comments and drill-down ability.
<b>Syntax</b>	WIFP_ALERT_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_ALERT
<b>Business Rule</b>	NONE
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Asset</b>
<b>Direction</b>	IN
<b>Description</b>	This interface receives asset data from other systems and creates or updates the assets in Oracle Utilities Work & Asset Management.
<b>Syntax</b>	WIFP_ASSET_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_ASSET
<b>Business Rule</b>	ASSET INTERFACE INTERFACE PARAMETERS where Parameter Name = ASSET ADDRESS PARSER VALUE: wipf_parse_address
<b>Sequence Number</b>	ASSET

<b>Interface Name</b>	<b>Blanket Contract</b>
<b>Direction</b>	IN/OUT



<b>Description</b>	This interface sends and receives blanket contracts or new revisions from other systems.
<b>Syntax</b>	WIFP_BLANKET_CONTRACT_INTERFAC(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_BLANKET_CONTRACT
<b>Business Rule</b>	N/A
<b>Sequence Number</b>	SA_BLANKET_CONTRACT

<b>Interface Name</b>	<b>Budget</b>
<b>Direction</b>	IN/OUT
<b>Description</b>	This interface sends account actuals and budget information to other systems. It also allows other systems to increase or decrease the budgeted amounts.
<b>Syntax</b>	WIFP_BUDGET_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_ACCOUNT_COST
<b>Business Rule</b>	BUDGET INTERFACE
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Catalog</b>
<b>Direction</b>	IN/OUT
<b>Description</b>	This interface sends or receives master catalog data from other systems. The OPTION_IN parameter allows control of what type of data is processed.
<b>Syntax</b>	WIFP_CATALOG_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_CATALOG
<b>Business Rule</b>	CATALOG INTERFACE
<b>Sequence Number</b>	SA_CATALOG

<b>Interface Name</b>	<b>Cost Adjustment</b>
<b>Direction</b>	IN
<b>Description</b>	This interface allows other systems to send indirect charges and journal entries to the Oracle Utilities Work and Asset Management application. This interface also can send "POSTED" cost adjustments out to a other system.
<b>Syntax</b>	WIFP_COST_ADJUSTMENT_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)

<b>Table Name</b>	WAIF_COST_ADJUSTMENT
<b>Business Rule</b>	NONE
<b>Sequence Number</b>	SA_ADJUSTMENT

<b>Interface Name</b>	<b>Customer</b>
<b>Direction</b>	IN
<b>Description</b>	This interface receives customer and address information from other applications. The interface creates the customers and can updates addresses in Oracle Utilities Work & Asset Management.
<b>Syntax</b>	WIFP_CUSTOMER_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_CUSTOMER
<b>Business Rule</b>	CUSTOMER INTERFACE, INTERFACE PARAMETERS where Parameter Name = CUSTOMER ADDRESS PARSER VALUE: wipf_parse_address
<b>Sequence Number</b>	SA_CUSTOMER

<b>Interface Name</b>	<b>Direct Charges</b>
<b>Direction</b>	IN/OUT
<b>Description</b>	Provides for purchasing transactions occurring outside of Oracle Utilities Work & Asset Management to be sent to the work order to properly reflect all relevant charges against an asset. The interface also allows for direct charges entered in Oracle Utilities Work & Asset Management to be sent to other applications.
<b>Syntax</b>	WIFP_DIRECT_CHARGE_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_DIRECT_CHARGE
<b>Business Rule</b>	NONE
<b>Sequence Number</b>	SA_DIRECT_CHARGES

<b>Interface Name</b>	<b>Employee</b>
<b>Direction</b>	IN/OUT
<b>Description</b>	This interface receives employee wage rate data from other systems.

<b>Syntax</b>	WIFP_EMPLOYEE_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_EMPLOYEE
<b>Business Rule</b>	EMPLOYEE INTERFACE
<b>Sequence Number</b>	SA_EMPLOYEE

<b>Interface Name</b>	<b>Employee Wage Rate</b>
<b>Direction</b>	IN
<b>Description</b>	This interface receives employee wage rate data from other systems.
<b>Syntax</b>	WIFP_EMP_WAGE_RATE_INTERFACE(JOB_IN,PLANT_IN, DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_EMPLOYEE_WAGE_RATE
<b>Business Rule</b>	NONE
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Fuel</b>
<b>Direction</b>	IN
<b>Description</b>	Provides for fuel transactions occurring outside of Oracle Utilities Work and Asset Management to be sent to the asset to properly reflect all relevant charges against an asset. Meter readings can also be logged, and based on the set up of the PM Master for the affected assets, Work Orders can be created and placed into the backlog for planning and execution.
<b>Syntax</b>	WIFP_FUEL_INTERFACE(JOB_IN,PLANT_IN, DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_FUEL_SYSTEM_DATA
<b>Business Rule</b>	EXPENSE CODES where Category = CONSUMABLES
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>General Ledger</b>
<b>Direction</b>	OUT
<b>Description</b>	This interface allows all Oracle Utilities Work and Asset Management transactions to be collected and sent to a client's accounting system. This interface can also create the correct offset transactions to satisfy accounting system's two sided general ledger.

<b>Syntax</b>	WIFP_GL_TRANS_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_GL_TRANSACTION
<b>Business Rule</b>	User Defined (see documentation)
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Inventory Log</b>
<b>Direction</b>	IN
<b>Description</b>	This interface receives inventory log transaction data from other systems.
<b>Syntax</b>	WIFP_INVENTORY_LOG_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_INVENTORY_LOG
<b>Business Rule</b>	INTERFACE PARAMETERS where Parameter Name = INVENTORY INTERFACE HANDLING
<b>Sequence Number</b>	SA_INVENTORY_LOG

<b>Interface Name</b>	<b>Invoice</b>
<b>Direction</b>	IN/OUT
<b>Description</b>	This interface allows all Oracle Utilities Work and Asset Management invoices to be collected and sent to a client's AP system. This interface will also load and post invoices from an external AP system.
<b>Syntax</b>	WIFP_INVOICE_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_INVOICE
<b>Business Rule</b>	INTERFACE PARAMETERS where Parameter Name = INVOICE STATUS
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Invoice AP Data</b>
<b>Direction</b>	IN
<b>Description</b>	This interface receives AP data (check number, AP amount, etc.) from an external AP system and updates the proper invoice in the Oracle Utilities Work and Asset Management application.

<b>Syntax</b>	WIFP_INVOICE_AP_DATA_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_INVOICE_AP_DATA
<b>Business Rule</b>	NONE
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Manufacturer Vendor</b>
<b>Direction</b>	IN/OUT
<b>Description</b>	This interface sends or receives catalog manufacturer vendor data from other systems. The OPTIONS parameter allows control of what type of data is processed.
<b>Syntax</b>	WIFP_MFR_VENDOR_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_CATALOG_MFR_VENDOR
<b>Business Rule</b>	MANUFACTURER VENDOR INTERFACE
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Meter Reading</b>
<b>Direction</b>	IN
<b>Description</b>	This interface receives meter reading data collected for certain Assets to initiate routine runtime maintenance activities automatically. The data for the identified assets will be passed on a routine basis to populate the Oracle Utilities Work and Asset Management application Run-Time (meter) Reading log. Based on the set up of the PM Master for the affected assets, Work Orders can be created and placed into the backlog for planning and execution.
<b>Syntax</b>	WIFP_METER_READING_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_METER_READING
<b>Business Rule</b>	N/A
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Purchase Order</b>
<b>Direction</b>	IN

<b>Description</b>	This interface allows Oracle Utilities Work and Asset Management to accept POs from an external purchasing system. POs are processed as if created on-line (inventory updated, change orders created).
<b>Syntax</b>	WIFP_PURCHASE_ORDER_INTERFACE.PO_INTERFACE(JOB_IN, PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_PURCHASE_ORDER
<b>Business Rule</b>	UPDATE PRIMARY VENDOR where Prime Vendor Rule = UPDATE_PRIME_VENDOR
<b>Sequence Number</b>	SA_PURCHASE_ORDER

<b>Interface Name</b>	<b>Receipt</b>
<b>Direction</b>	IN
<b>Description</b>	This interface allows Oracle Utilities Work and Asset Management to accept receipts from a third party receiving system. Receipts are processed as if created on-line (alerts sent, inventory updated PO updated, component IDs and lot IDs accepted).
<b>Syntax</b>	WIFP_RECEIPT_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_RECEIPT
<b>Business Rule</b>	NONE
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Requisition</b>
<b>Direction</b>	OUT
<b>Description</b>	This interface allows for requisitions to be sent to other systems.
<b>Syntax</b>	WIFP_REQUISITION_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_REQUISITION
<b>Business Rule</b>	NONE
<b>Sequence Number</b>	SA_REQUISITION

<b>Interface Name</b>	<b>Service Request</b>
<b>Direction</b>	IN

<b>Description</b>	This interface receives service requests from other systems so they can be worked through Oracle Utilities Work & Asset Management.
<b>Syntax</b>	WIFP_SERVICE_REQUEST_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_SERVICE_REQUEST
<b>Business Rule</b>	NONE
<b>Sequence Number</b>	SA_SERVICE_REQUEST

<b>Interface Name</b>	<b>Storeroom</b>
<b>Direction</b>	IN/OUT
<b>Description</b>	This interface sends or receives storeroom data from other systems. The OPTIONS parameter allows control of what type of data is processed.
<b>Syntax</b>	WIFP_STOREROOM_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_STOREROOM
<b>Business Rule</b>	STOREROOM INTERFACE
<b>Sequence Number</b>	N/A

<b>Interface Name</b>	<b>Timekeeping</b>
<b>Direction</b>	IN/OUT
<b>Description</b>	This interface allows Oracle Utilities Work and Asset Management to receive or send timesheets to an external payroll system.
<b>Syntax</b>	WIFP_TIMEKEEPING_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_TIMEKEEPING
<b>Business Rule</b>	NONE
<b>Sequence Number</b>	SA_TIMESHEET

<b>Interface Name</b>	<b>Vendor</b>
<b>Direction</b>	IN/OUT
<b>Description</b>	This interface sends or receives vendor data from other systems. The OPTIONS parameter allows control of what type of data is processed.

<b>Syntax</b>	WIFP_VENDOR_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
<b>Table Name</b>	WAIF_VENDOR
<b>Business Rule</b>	VENDOR INTERFACE
<b>Sequence Number</b>	SA_VENDOR

## Basic Batch Job Procedure Syntax

All of the interfaces have the following basic syntax:

```
PROCEDUREwifp_xxxxx_interface
(job_in   IN NUMBER
plant_in  IN VARCHAR2
direction_in IN VARCHAR2
purge_in  IN VARCHAR2
option_in IN VARCHAR2
pre_in    IN VARCHAR2
post_in   IN VARCHAR2)
```

## Parameters

**Job\_In** = The Job Number is generated by the Job Manager module in the application, using the "Create Job" Action.

**Plant\_In** = This is the plant number where the job is created. Only records with this plant code will be processed.

**Direction\_In** = Some interfaces can process either inbound or outbound records. Usually there will be a corresponding field in the interface table to indicate which records are inbound and which are outbound. Every interface has this parameter, even if it does not currently process records in both directions.

**Purge\_In** = This flag indicates whether records that have been processed should be deleted from the interface table, or if they should remain in the table but with the SENT\_TO\_INTERFACE\_IND flag set to "Y" (indicating that the record has already been processed and should not be processed again).

**Option\_In** = This is a generic parameter that allows each interface to have custom inputs that are particular to that interface. For example, the Option\_In parameter on the inbound Account interface indicates whether expense codes should also be processed, and on the outbound Account interface to indicate whether only active accounts should be interfaced.

**Pre\_In** = This parameter allows you to provide a pre-processing step before the interface code is run. For example, on an inbound interface, you may use the Pre\_In procedure to retrieve data from a file and insert it into the interface table, for inbound processing by the interface. Generally, this parameter will consist of a stored procedure call, in quotes, and including a job number and a plant code. The job number would be required if you want to log messages to the job manager log. The plant code allows you to ensure that only the data for the current plant is processed.

**Post\_In** = This is similar to the Pre\_In parameter above, except that it processes after the interface code. For example, you could use this to take the data from the interface table produced by an outbound interface and write it out to a file.



## Processing

All of the procedures work in the same general order:

1. Insert a row into the Job Manager Log to indicate the start of the job.
2. Execute the pre-processing procedure and check for errors (exceptions).
3. Perform the actions of the interface. For inbound interfaces, this usually includes looping through the contents of the interface table, processing each record as appropriate (usually by inserting data into the proper application tables), and marking the record as processed. For outbound interfaces, this usually includes looping through the unprocessed records in the proper application tables and inserting the data into the interface table.
4. Execute the post-processing procedure and check for errors (exceptions).
5. Write out any messages that indicate the status of the job, number of records processed, etc.
6. Mark the job status as Complete.

The purge action, if the purge flag is set to "Y", occurs at different locations in different interfaces, but usually occurs before the pre-procedure or before the post-procedure.

## Pre- and Post-Processing Routines

Each interface has parameters that you can use to perform processing before or after the interface runs. For example, you can use the pre-processing parameter move inbound data from a file into the interface table, and the post-processing parameter to move outbound data from the interface table to a file.

Pre- and post-processing parameters should consist of a call to a stored procedure or package procedure that will perform your processing. You must specify this parameter as a character string. If you need to use single quote characters around any parameters to your procedure, you will need to use two single quotes (not a double-quote) to surround these parameters. You will probably want a job number parameter (to log messages) and a plant parameter (to ensure that only data from the current plant is processed).

For example, if your procedure signature is this:

```
Create Procedure Cifp_Get_data_from_file(job_in number, plant_in varchar2);
```

Your pre- or post-processing job parameter should be specified like this:

```
'cifp_get_data_from_file(777, "01")'
```

where 777 is your job number and 01 is your plant code. Note the two single quotes around the plant code.

## Utility Functions

The following utility functions can be used by your custom pre- and post- procedures, to fully integrate your interface into the application.

In your custom pre- and post-procedures, you may want to log information in the job manager log about the progress of your procedure. For example, you may want to log stop and start times, or informational message. The following routines will allow you to insert a new job sequence number for your procedure and log any messages for users to review.

## wifp\_log\_job\_mgr\_job

This procedure is called at the top and bottom of your custom procedure, to create the job sequence in STARTED status, and then to update the status to indicate completion (for example, COMPLETED\_NO\_ERRORS). The routine does not commit changes.

```
wifp_log_job_mgr_job (
  job_in IN NUMBER,
  plant_in IN VARCHAR2,
  job_seq_in_out IN OUT NUMBER,
  status_in IN VARCHAR2,
  job_desc_msg_in IN VARCHAR2,
  error_no IN OUT NUMBER,
  error_msg IN OUT VARCHAR2
)
--
```

### Parameters

**job\_in** = Job Manager assigned job number. Your custom procedure should have Job\_In as a parameter, and you can pass it to this procedure. It will be the same as the interface procedure job.

**plant\_in** = Plant number. Your custom procedure should have Plant\_In as a parameter, and you can pass it in to this procedure.

**job\_seq\_in\_out** = This parameter corresponds to a particular run of the job. See the Status\_in parameter below.

**Status\_in** = Status code. If input status = STARTED, then sequence number is generated. Otherwise status is just updated.

**job\_desc\_msg\_in** = This should be a description of your procedure, for the initial log message. Usually it may include the name of your procedure, along with any custom parameters you may have. For example: "GET\_PURCHASE\_ORDERS\_FROM\_FILE()".

**error\_no** = The error message number. 0 = no error

**error\_msg** = Error message text

## wifp\_log\_job\_mgr\_log\_message

This procedure is called to log a message to the job manager log, for your job. This routine does not commit changes.

```
wifp_log_job_mgr_log_message (
  job_in IN NUMBER,
  plant_in IN VARCHAR2,
  job_seq_in IN NUMBER,
  message_in IN VARCHAR2,
  error_no IN OUT NUMBER,
  error_msg IN OUT VARCHAR2
)
```

### Parameters

**job\_in** = Job Manager assigned job number. Your custom procedure should have Job\_In as a parameter, and you can pass it to this procedure. It will be the same as the interface procedure job.

**plant\_in** = Plant number. Your custom procedure should have Plant\_In as a parameter, and you can pass it in to this procedure.

**job\_seq\_in** = This parameter corresponds to a particular run of the job. It will be generated and returned to you by the routine wifp\_log\_job\_mgr\_job (above).

**message\_in** = message text to be logged.

**error\_no** = The error message number. 0 = no error

**error\_msg** = Error message text

## wifp\_generate\_key\_interface

On inbound interfaces, when you want to create new entities in the application, you will often need to generate a key value for the item. For example, if you are inserting a new purchase order, your pre-procedure that is inserting records into the interface table may call this procedure to generate the new PO number. This procedure will generate the new number based on your sequence number configuration for purchase orders.

```
PROCEDURE wifp_generate_key_interface
(plant_in in varchar2,
 seq_no_name_in in varchar2,
 seq_no_len_in in number,
 seq_no_out in out varchar2,
 error_no in out number,
 error_msg in out varchar2)
```

### Parameters

**Plant\_In** = This is the plant to which you are inserting data and wish to generate a key for. Your custom procedure should have Plant\_In as a parameter, and you can pass it in to this procedure.

**Seq\_No\_Name\_In** = This is the name of the sequence that you need a new key for. Usually this is the application table name. Sequence names are listed in each interface chapter.

**Seq\_No\_Len\_in** = This is the maximum length of the key you wish to generate. It will generally correspond to the length of your varchar field seq\_no\_out, where the generated key will be written to. If your sequence number configuration generates a shorter number, then your output value will be shorter.

**Seq\_no\_out** = This output parameter will be set to the generated key value. When the sequence actually generates a number (for example, the SA\_Timesheet\_Seq\_No value), you will need to convert it to a number using TO\_NUMBER().

**Error\_No** = This output parameter indicates whether the key generation was successful or not. If the return value is 0, then no error occurred. If the return value is not 0, then you should report the error and not continue processing this record.

**Error\_Msg** = If an error occurs, this output parameter will contain text that describes what error occurs. You should write this information to the job manager log.

---

# Chapter 2

## Account

The Account interface provides a means to import and export the list of accounts that are used for recording costs in Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table:** WAIF\_ACCOUNT

**Stored Procedure:** WIFP\_ACCOUNT\_INTERFACE

**Business Rule:** ACCOUNT INTERFACE - controls the data elements updated by the stored procedure.

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management Job Manager according to the client's requirements.

```
WIFP_ACCOUNT_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

Only processed records are purged. The job number is used by the interface to determine which records to inactivate, if the Inactivate option is chosen.

On outbound, if you select the Update option, only those accounts updated after the last job run are interfaced. Unprocessed records are not purged and you may have more than one record for the account in the table.

If you do not select the Update option, purge "Y" will remove all existing outbound records and replace them with new ones. Unless you are sending only updates, you should always set the purge flag to "Y".

## Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management Job Manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to the Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from Oracle Utilities Work and Asset Management application to the interface table.

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y** = If DIRECTION = **I**, records will be deleted from the interface table as they are processed. If DIRECTION = **O**, records will be deleted from the interface table before writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_ACCOUNT\_INTERFACE runs again.

**OPTIONS\_IN = VARCHAR2** - (Optional) Character string that identifies the special processing options.

If DIRECTION\_IN = **I** and...

- **OPTION = I** - The interface procedure will 'Inactivate' all pre-existing account records in the Oracle Utilities Work and Asset Management application that were not created, or updated, by the current run of the interface procedure. This would be assuming that the interface table contains all the active accounts and that all other accounts are no longer active. If this option is combined with **E** (see below), all SA\_ACCOUNT\_EXPENSE\_CODES records that were not created by the current interface run will be removed. This has the effect of inactivating the given account/expense code combination.
- **OPTION = E** - The interface will create a SA\_ACCOUNT\_EXPENSE\_CODES record for each account number when EXPENSE\_CODE is present in the interface table. This allows the Oracle Utilities Work and Asset Management application to restrict account/expense code combinations.
- **OPTION = B** - After completing, the interface will make a call to WIFP\_BUILD\_EXPENSE\_CODES to backfill the EXPENSE CODES business rule based on data in the SA\_ACCOUNT\_EXPENSE\_CODES table.

Valid Combinations: space, **I**, **IE**, **IEB**, **E**, **EB**, **B**

If DIRECTION\_IN = **O** and...

- **OPTION = A** - The interface procedure will **only** write 'Active' account records to the interface table.
- **OPTION = U** - The interface procedure will **only** write account records that have been updated since the last run of the interface procedure to the interface table.

- **OPTION = E** - The interface procedure will include expense code data for each account. This data comes from SA\_ACCOUNT\_EXPENSE\_CODES.

Valid Combinations: **space, A, AU, AUE, U, UE, E**

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Business Rule Format

This business rule only applies to pre-existing inbound records.

**RULE ID** = ACCOUNT INTERFACE

**RULE TYPE** = INTERFACE

**RULE STYLE** = LIST

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound account record that already exists in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_ACCOUNT\_DATA column to be updated.

**UPDATE** = YES or NO. If this field is set to YES then the corresponding field will be updated.

## Interface Table Layout

### WAIF\_ACCOUNT

(same fields as SA\_ACCOUNT + SA\_ACCOUNT\_EXPENSE\_CODES)

Field Name	Data Type
PLANT*	VARCHAR2(3)
ACCOUNT_NO*	VARCHAR2(75)
ACCOUNT_SEGMENT1	VARCHAR2(20)
ACCOUNT_SEGMENT2	VARCHAR2(20)
ACCOUNT_SEGMENT3	VARCHAR2(20)
ACCOUNT_SEGMENT4	VARCHAR2(20)
ACCOUNT_SEGMENT5	VARCHAR2(20)
ACCOUNT_DESC	VARCHAR2(60)
LAST_UPDATE_DATE*	DATE
ACCOUNT_STATUS*	VARCHAR2(10)
DEPARTMENT	VARCHAR2(10)

Field Name	Data Type
AREA	VARCHAR2(10)
ACCOUNT_REFERENCE_ID	VARCHAR2(7)
ACCOUNT_SEGMENT6	VARCHAR2(20)
ACCOUNT_SEGMENT7	VARCHAR2(20)
ACCOUNT_SEGMENT8	VARCHAR2(20)
ACCOUNT_SEGMENT9	VARCHAR2(20)
ACCOUNT_SEGMENT10	VARCHAR2(20)
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_DATE*	DATE
CREATED_BY*	VARCHAR2(30)
ATTRIBUTE1	USER DEFINED
ATTRIBUTE2	USER DEFINED
ATTRIBUTE3	USER DEFINED
ATTRIBUTE4	USER DEFINED
ATTRIBUTE5	USER DEFINED
ATTRIBUTE6	USER DEFINED
ATTRIBUTE7	USER DEFINED
ATTRIBUTE8	USER DEFINED
ATTRIBUTE9	USER DEFINED
ATTRIBUTE10	USER DEFINED
DEFAULT_APPROVER	VARCHAR2(6)
BUDGET_TYPE	VARCHAR2(8)
ROUTING_LIST_ID	VARCHAR2(10)
SENT_TO_INTERFACE_IND	VARCHAR2(1)
JOB_SEQ_NO	NUMBER
EXPENSE_CODE	VARCHAR2(10)
EXPENSE_DESC	VARCHAR2(60)
EXPENSE_CATEGORY	VARCHAR2(8)
DIRECTION*	CHAR(1) I = Inbound, O = Outbound
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)

Field Name	Data Type
INT_INSTANCE_ID	NUMBER

\* required fields



---

# Chapter 3

## Accrual

The Accrual interface provides a means to export amounts owed for PO line items, where the items have been received, but not invoiced, in Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table:** WAIF\_ACCRUAL

**Stored Procedure:** WIFP\_ACCRUAL\_INTERFACE

**Business Rule:** ACCRUAL INTERFACE -controls the stored procedure.

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management Job Manager according to the client's requirements.

```
WIFP_ACCRUAL_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

### Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management Job Manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application. This option is not implemented for this interface.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table.

**OPTION\_IN = VARCHAR2** - Character string that identifies the PO line types to accrue.

Example: 'MSX' Null = All PO Line Types

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table. The Accrual interface should always be run with PURGE\_IN set to "Y". With this setting, records will be deleted from the interface table **before** writing out new records. Purge deletes all records in the Plant, regardless of the "Sent" flag. Only selects records where Received Quantity > Invoiced Quantity and Status = Issued, Invoiced, or Received.

**PRE\_IN = VARCHAR2** - Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN = VARCHAR2** - Character string that identifies the custom stored procedure to call after the interface procedure begins.

## Business Rule Format

**RULE ID** = ACCRUAL INTERFACE

**RULE TYPE** = INTERFACE

**RULE STYLE** = PARAMETER

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = This rule identifies the offsetting account and expense code to be used by the accrual interface.

**OFFSET NAME** = ACCRUAL

**EXPENSE CODE** = User entered expense code to be used for offsetting accruals (optional).

**ACCOUNT NO.** = User entered account number to be used for offsetting accruals (optional).

## Process Flow

1. If a PRE\_OUT procedure is specified, execute it.
2. If PURGE is equal to Y, delete ALL records in WAIF\_ACCRUAL.
3. Process all ISSUED, RECEIVED, FULLY, and INVOICED, FULLY purchase orders that have line items with RECEIVED\_NET\_QUANTITY greater than zero, RECEIVED\_NET\_QUANTITY greater than INVOICED\_QUANTITY, and PO\_ITEM\_TYPE as listed in OPTION\_IN.
4. For each qualified purchase order line item, write a record to WAIF\_ACCRUAL.
5. If the business rule ACCRUAL INTERFACE exists, create an offsetting record in WAIF\_ACCRUAL for each qualified purchase order line item, otherwise do not create an offsetting record.

If a POST\_OUT procedure is specified, execute it.

## Interface Table Layout

### WAIF\_ACCRUAL

Field Name	Data Type	Description
PLANT	VARCHAR2(3)	Plant Code
ACCRUAL_RUN_DATE	DATE	Accrual Run Date
PO_NO	VARCHAR2(10)	Purchase Order Number
PO_ITEM	VARCHAR2(3)	Purchase Order Item
ACCOUNT_NO	VARCHAR2(75)	Account Number
EXPENSE_CODE	VARCHAR2(10)	Expense Code
UNIT_PRICE	NUMBER(17,4)	Unit Price
RECEIVED_NET_QUANTITY	NUMBER(11,2)	Net Quantity Received
INVOICED_QUANTITY	NUMBER(11,2)	Invoiced Quantity
ACCRUAL_QUANTITY	NUMBER(11,2)	Accrued Quantity = (RECEIVED_NET_QUANTITY - INVOICED_QUANTITY) * (PERCENT_SPLIT / 100)
ACCRUAL_AMOUNT	NUMBER(15,2)	Accrued Amount = (ACCRUAL_QUANTITY * UNIT_PRICE) * (PERCENT_SPLIT / 100)
OFFSET_IND	CHAR(1)	Offset Indicator N=original record Y= offset record
SOURCE_SYSTEM	VARCHAR2(3)	
INT_BATCH_NUMBER	NUMBER	
JOB_STATUS	VARCHAR2(15)	
JOB_MESSAGE	VARCHAR2(4000)	
INT_INSTANCE_ID	NUMBER	
DIRECTION	CHAR(1)	

\* required fields



---

# Chapter 4

## Alert

The Alert interface allows external systems to send alerts to Oracle Utilities Work & Asset Management users. Once imported into the application, these interfaced alerts function like on-line alerts with comments and drill-down ability.

This interface requires:

**Interface Table:** WAIF\_ALERT

**Stored Procedure:** WIFP\_ALERT\_INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management Job Manager according to the client's requirements.

```
WIFP_ALERT_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

### Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management Job Manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not implemented for this interface.

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y** = Records will be deleted from the interface table **after** they are processed.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_ALERT\_INTERFACE runs again.

**OPTION\_IN** = Not used (pass null).

**PRE\_IN= VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Interface Table Layout

### WAIF\_ALERT

Field Name	Data Type	Description
ADDRESSEE*	VARCHAR2(30)	Username the alert is to be sent to as defined in User Profile.
DESCRIPTION*	VARCHAR2(2000)	Description (message) of the alert.
MODULE	VARCHAR2 (8)	Optional. Module to be opened upon drilling down on this alert.
WHERE_CLAUSE	VARCHAR2 (2000)	Optional. Where clause to use to filter records when drilling down. This has the same format as an Oracle SQL where clause without the WHERE keyword.
ORDER_BY	VARCHAR2 (2000)	
COMMENTS	VARCHAR2 (2000)	Optional. Additional comments for this alert.
SEQUENCE	NUMBER	Optional. Sequential number used for determining the order in which to create alerts.
SENT_TO_INTERFACE	CHAR(1)	
SOURCE_SYSTEM	VARCHAR2(3)	
INT_BATCH_NUMBER	NUMBER	
JOB_STATUS	VARCHAR2(15)	
JOB_MESSAGE	VARCHAR2(4000)	
INT_INSTANCE_ID	NUMBER	
DIRECTION	CHAR(1)	

\* required fields

# Asset

The Asset interface allows Oracle Utilities Work & Asset Management to interface asset data with other systems.

This interface requires:

**Interface Table:** WAIF\_ASSET

**Stored Procedure:** WIFP\_ASSET\_INTERFACE

**Business Rule:** ASSET INTERFACE - controls the data elements updated by the stored procedure.

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

## Batch Job Procedure

The stored procedure is scheduled and run in the Job Manager according to the client's requirements.

Updates will always set the last update date and the job sequence number.

```
WIFP_ASSET_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN  
PRE_IN,  
POST_IN  
)
```

## Keywords and Parameters

**JOB\_IN = Integer** - the job number assigned by the Oracle Utilities Work and Asset Management Job Manager.

**PLANT\_IN** = VARCHAR2 - identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not yet implemented for this interface.

**PURGE\_IN** = VARCHAR2 - Single character that identifies how records will be removed from the interface table.

**Y** = If DIRECTION = I, records will be deleted from the interface table as they are processed. If DIRECTION = O, previously processed records will be deleted from the interface table before writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_ASSET\_INTERFACE runs again.

**OPTION\_IN** = VARCHAR2 - (Optional) - Identifies any special processing options. This interface currently has no special options implemented.

**PRE\_IN** = VARCHAR2 (Optional) - Identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN** = VARCHAR2 (Optional) - Identifies the custom stored procedure to call after the interface code executes.

## Business Rule Format

This business rule only applies to pre-existing inbound records.

**Rule ID** = ASSET INTERFACE

**Type** = INTERFACE

**Style** = LIST

**Limit** = OFF

**Description** = Controls the fields that will be updated on an inbound asset record that already exists in Oracle Utilities Work and Asset Management.

**Column Name** = Name of SA\_ASSET column to be updated. List of values provided.

**Update** = YES or NO. List of values provided.

## Sequence Numbers

ASSET

## Interface Table Layout

### WAIF\_ASSET

PLANT* (cannot be null)	VARCHAR2(3)
ASSET_RECORD_TYPE* (prime key)	VARCHAR2(1)
ASSET_ID* (prime key)	VARCHAR2(15)
ASSET_TYPE	VARCHAR2(10)
ASSET_DESC	VARCHAR2(2000)
LAST_UPDATE_DATE*	DATE
CCB_SOURCE	VARCHAR2(15)
PARSE_ADDRESS	VARCHAR(1)
ASSET_STATUS*	VARCHAR2(20)
DEPARTMENT	VARCHAR2(10)



AREA	VARCHAR2(10)
PROCESS_NO	VARCHAR2(15)
PARENT_ASSET_ID	VARCHAR2(15)
BOM_ID	VARCHAR2(15)
ACCOUNT_NO	VARCHAR2(75)
FIXED_ASSET_NO	VARCHAR2(20)
CRITICALITY	CHAR
SAFETY_CRITICAL_IND	CHAR
SAFETY_NOTES_IND	CHAR
MAINTENANCE_MANAGER	VARCHAR2(6)
PLANNER	VARCHAR2(6)
MAINTENANCE_APPROVER	VARCHAR2(6)
PRODUCTION_APPROVER	VARCHAR2(6)
BACKLOG_GROUP	VARCHAR2(6)
SPECIFICATION_NO	VARCHAR2(10)
SPECIFICATION_TYPE	VARCHAR2(10)
SPECIFICATION_CATEGORY	VARCHAR2(15)
BREAKER_NO	VARCHAR2(12)
BUILDING	VARCHAR2(20)
LOCATION	VARCHAR2(30)
POSITION	VARCHAR2(20)
NOTES_IND	CHAR
FUTURE_RETIREMENT_IND	CHAR
FUTURE_RETIREMENT_DATE	DATE
PARENT_ASSET_RECORD_TYPE	VARCHAR2(1)
POINT_ID	VARCHAR2(30)
SERIAL_NO	VARCHAR2(30)
INSPECTION_DATE	DATE
TITLE_NO	VARCHAR2(20)
MODEL_YEAR	VARCHAR2(4)
MAKE	VARCHAR2(10)
MODEL	VARCHAR2(20)
ENGINE_SIZE	VARCHAR2(15)

ENGINE_DESCRIPTION	VARCHAR2(20)
TRANSMISSION_SIZE	VARCHAR2(15)
GROSS_VEHICLE_WEIGHT	NUMBER
FUEL_TYPE	VARCHAR2(10)
RADIO_IND	CHAR
DRIVE_TYPE	VARCHAR2(10)
AIR_CONDITION_IND	CHAR
CRUISE_CONTROL_IND	CHAR
COLOR	VARCHAR2(10)
TIRE_SIZE	VARCHAR2(15)
CLASS	VARCHAR2(20)
ASSET_SEGMENT1	VARCHAR2(20)
ASSET_SEGMENT2	VARCHAR2(20)
ASSET_SEGMENT3	VARCHAR2(20)
ASSET_SEGMENT4	VARCHAR2(20)
ASSET_SEGMENT5	VARCHAR2(20)
ASSET_SEGMENT6	VARCHAR2(20)
ASSET_SEGMENT7	VARCHAR2(20)
ASSET_SEGMENT8	VARCHAR2(20)
ASSET_SEGMENT9	VARCHAR2(20)
ASSET_SEGMENT10	VARCHAR2(20)
ASSET_KEY_SEGMENTS	VARCHAR2(75)
CAR_PHONE_IND	CHAR
STEREO_IND	CHAR
SHOP	VARCHAR2(10)
CREATION_DATE*	DATE
VEHICLE_LICENSE_TAG	VARCHAR2(20)
ORGANIZATION	VARCHAR2(10)
SUB_ORG	VARCHAR2(10)
EXTRAS	VARCHAR2(18)
FEDERAL_TAX_CODE	CHAR
STATE_PROVINCE_TAX_CODE	CHAR
DUTY_CODE	CHAR

ATTRIBUTE1	USER DEFINED
ATTRIBUTE2	USER DEFINED
ATTRIBUTE3	USER DEFINED
ATTRIBUTE4	USER DEFINED
ATTRIBUTE5	USER DEFINED
ATTRIBUTE6	USER DEFINED
ATTRIBUTE7	USER DEFINED
ATTRIBUTE8	USER DEFINED
ATTRIBUTE9	USER DEFINED
ATTRIBUTE10	USER DEFINED
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_BY*	VARCHAR2(30)
ROOM	VARCHAR2(20)
ISO_IND	VARCHAR2(1)
ENVIRONMENTAL_IND	VARCHAR2(1)
HEALTH_IND	VARCHAR2(1)
MANUFACTURER_CODE	VARCHAR2(20)
MANUFACTURER_DRAWING_NO	VARCHAR2(15)
MANUFACTURER_MODEL_NO	VARCHAR2(30)
MANUFACTURER_MODEL_REVISION_NO	NUMBER(3)
MANUFACTURER_ORDER_NO	VARCHAR2(15)
MANUFACTURER_PART_NO	VARCHAR2(50)
WARRANTY_EXPIRATION_DATE	DATE
WARRANTY_DESC	VARCHAR2(2000)
ORIGINAL_INSTALL_DATE	DATE
LAST_INSTALL_DATE	DATE
ROUTING_LIST_ID	VARCHAR2(10)
ASSET_CLASS	VARCHAR2(20)
LOCATION_BASIS	VARCHAR2(20)
NUMBER_PREFIX	VARCHAR2(3)
STREET_NUMBER	NUMBER
NUMBER_SUFFIX	VARCHAR2(5)
STREET_NAME	VARCHAR2(40)

STREET_DIRECTION	VARCHAR2(3)
CROSS_STREET	VARCHAR2(40)
CITY	VARCHAR2(40)
STATE_PROVINCE	VARCHAR2(4)
POSTAL_CODE	VARCHAR2(10)
OFFSET	VARCHAR2(200)
DIRECTION	VARCHAR2(10)
FROM_ASSET_RECORD_TYPE	VARCHAR2(1)
TO_ASSET_RECORD_TYPE	VARCHAR2(1)
FROM_ASSET_ID	VARCHAR2(15)
TO_ASSET_ID	VARCHAR2(15)
SENT_TO_INTERFACE_IND	VARCHAR2(1)
JOB_SEQ_NO	NUMBER
SUITE	VARCHAR2(10)
STREET_NUMBER_CHAR	VARCHAR2(10)
LOG_REVIEWER	VARCHAR2(30)
RIVA_ASSET_TYPE	VARCHAR2(20)
ASSET_MOBILE_1	VARCHAR2(30)
ASSET_MOBILE_2	VARCHAR2(30)
ASSET_MOBILE_3	VARCHAR2(30)
BREAKER_ASSET_RECORD_TYPE	VARCHAR2(1)
BREAKER_ASSET_ID	VARCHAR2(15)
GIS_GPS_LONGITUDE	VARCHAR2(15)
GIS_GPS_LATITUDE	VARCHAR2(15)
OMS_DEVICE_NUMBER	VARCHAR2
RUN_TO_FAILURE_IND	CHAR(1)
BREAKER_PANEL	VARCHAR2(15)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER
DIRECTION_IND	CHAR(1)

\* required fields

## Asset Web Services

This section describes the implementation of the web service that enables external applications to insert, update, delete and select Asset records in the Oracle Utilities Work and Asset Management application. This web service was created using Apache Axis.

### Web Service Methods

The Asset Tables web service exposes four methods to external applications:

- InsertAssetData - used for importing Asset records into ORACLE UTILITIES WORK AND ASSET MANAGEMENT
- UpdateAssetData - used for updating Asset records in ORACLE UTILITIES WORK AND ASSET MANAGEMENT
- DeleteAssetData - used for deleting Asset records from Oracle Utilities Work and Asset Management
- SelectAssetData - used for selecting Asset records in Oracle Utilities Work and Asset Management

Each of these methods accepts an array of data types related to asset records, including description, location and account attributes.

After the submitted entries are processed, the method returns a confirmation message. All imported entries are processed by the Oracle Utilities Work and Asset Management application from an interface table.

Please see the WSDL section of this document for a complete description of the web service elements and structure.

### Web Service Security

Oracle Utilities Work and Asset Management web services require no specific roles or responsibilities.

### Web Service URL

The web service is deployed to the following environment.

`http://server:port/synergen/services/AssetTables`  
for example:

`http://bali:1751/synergen/services/AssetTables`

### Web Service Definition Language (WSDL)

```
<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions
targetNamespace="http://bali:1751/synergen/services/AssetTables"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns1="http://webservice.synergen" xmlns:tns2="urn:Asset"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:intf="http://bali:1751/synergen/services/AssetTables"
xmlns:impl="http://bali:1751/synergen/services/AssetTables">
- <!--
```

```

WSDL created by Apache Axis version: 1.2RC3
Built on Feb 28, 2005 (10:15:14 EST)
-->
- <types>
- <schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://webservice.synergen"
elementFormDefault="qualified">
  <import namespace="urn:Asset" />
  - <element name="InsertAssetData">
  - <complexType>
  - <sequence>
    <element name="in0" type="tns2:AssetData" />
  </sequence>
  </complexType>
  </element>
  - <complexType abstract="true" name="WebServiceData">
    <sequence />
  </complexType>
  - <element name="InsertAssetDataResponse">
  - <complexType>
  - <sequence>
    <element name="InsertAssetDataReturn" type="tns2:AssetData" />
  </sequence>
  </complexType>
  </element>
  - <element name="UpdateAssetData">
  - <complexType>
  - <sequence>
    <element name="in0" type="tns2:AssetData" />
  </sequence>
  </complexType>
  </element>
  - <element name="UpdateAssetDataResponse">
  - <complexType>
  - <sequence>
    <element name="UpdateAssetDataReturn" type="xsd:int" />
  </sequence>
  </complexType>
  </element>
  - <element name="DeleteAssetData">
  - <complexType>
  - <sequence>
    <element name="in0" type="tns2:AssetData" />
  </sequence>
  </complexType>
  </element>
  - <element name="DeleteAssetDataResponse">
  - <complexType>
  - <sequence>
    <element name="DeleteAssetDataReturn" type="xsd:int" />
  </sequence>
  </complexType>
  </element>
  - <element name="SelectAssetData">
  - <complexType>
  - <sequence>
    <element name="in0" type="tns2:AssetData" />
  </sequence>
  </complexType>
  </element>
  - <element name="SelectAssetDataResponse">
  - <complexType>
  - <sequence>
    <element name="SelectAssetDataReturn" type="tns2:AssetData"
maxOccurs="unbounded" />
  </sequence>

```

```

        </complexType>
    </element>
</schema>
- <schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:Asset" elementFormDefault="qualified">
    <import namespace="http://webservice.synergen" />
    <complexType name="AssetData">
        <complexContent>
            <extension base="tnsl:WebServiceData">
                <sequence>
                    <element name="PLANT" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="ASSET_RECORD_TYPE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="ASSET_ID" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="ASSET_TYPE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="ASSET_DESC" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="LAST_UPDATE_DATE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:dateTime" />
                    <element name="ASSET_STATUS" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="DEPARTMENT" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="AREA" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="PROCESS_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="PARENT_ASSET_ID" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="BOM_ID" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="ACCOUNT_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="FIXED_ASSET_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="CRITICALITY" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="SAFETY_CRITICAL_IND" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
                    <element name="SAFETY_NOTES_IND" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="MAINTENANCE_MANAGER" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
                    <element name="PLANNER" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="MAINTENANCE_APPROVER" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
                    <element name="PRODUCTION_APPROVER" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
                    <element name="BACKLOG_GROUP" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="SPECIFICATION_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="SPECIFICATION_TYPE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
                    <element name="SPECIFICATION_CATEGORY" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
                    <element name="BREAKER_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="BUILDING" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                    <element name="LOCATION" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</schema>

```

```

        <element name="POSITION" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="NOTES_IND" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="FUTURE_RETIREMENT_IND" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="FUTURE_RETIREMENT_DATE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:dateTime" />
        <element name="PARENT_ASSET_RECORD_TYPE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="POINT_ID" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="SERIAL_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="INSPECTION_DATE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:dateTime" />
        <element name="TITLE_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="MODEL_YEAR" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="MAKE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="MODEL" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ENGINE_SIZE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ENGINE_DESCRIPTION" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="TRANSMISSION_SIZE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="GROSS_VEHICLE_WEIGHT" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:int" />
        <element name="FUEL_TYPE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="RADIO_IND" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="DRIVE_TYPE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="AIR_CONDITION_IND" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="CRUISE_CONTROL_IND" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="COLOR" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="TIRE_SIZE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="CLASS" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_SEGMENT1" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_SEGMENT2" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_SEGMENT3" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_SEGMENT4" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_SEGMENT5" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_SEGMENT6" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_SEGMENT7" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_SEGMENT8" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_SEGMENT9" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />

```



```

        <element name="ASSET_SEGMENT10" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_KEY_SEGMENTS" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="CAR_PHONE_IND" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="STEREO_IND" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="SHOP" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="CREATION_DATE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:dateTime" />
        <element name="VEHICLE_LICENSE_TAG" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="ORGANIZATION" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="SUB_ORG" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="EXTRAS" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="FEDERAL_TAX_CODE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="STATE_PROVINCE_TAX_CODE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="DUTY_CODE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE1" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE2" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE3" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE4" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE5" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE6" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE7" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE8" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE9" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE10" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="LAST_UPDATE_USER" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="CREATED_BY" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ROOM" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ISO_IND" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ENVIRONMENTAL_IND" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="HEALTH_IND" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="MANUFACTURER_CODE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="MANUFACTURER_DRAWING_NO" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="MANUFACTURER_MODEL_NO" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="MANUFACTURER_MODEL_REVISION_NO" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:int" />

```

```

        <element name="MANUFACTURER_ORDER_NO" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="MANUFACTURER_PART_NO" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="WARRANTY_EXPIRATION_DATE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:dateTime" />
        <element name="WARRANTY_DESC" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ORIGINAL_INSTALL_DATE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:dateTime" />
        <element name="LAST_INSTALL_DATE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:dateTime" />
        <element name="ROUTING_LIST_ID" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ASSET_CLASS" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="LOCATION_BASIS" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="NUMBER_PREFIX" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="STREET_NUMBER" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:int" />
        <element name="NUMBER_SUFFIX" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="STREET_NAME" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="STREET_DIRECTION" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="CROSS_STREET" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="CITY" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="STATE_PROVINCE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="POSTAL_CODE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="OFFSET" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="DIRECTION" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="FROM_ASSET_RECORD_TYPE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="TO_ASSET_RECORD_TYPE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="FROM_ASSET_ID" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="TO_ASSET_ID" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="SENT_TO_INTERFACE_IND" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="JOB_SEQ_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:int" />
        <element name="SUITE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="STREET_NUMBER_CHAR" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
        <element name="LOG_REVIEWER" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="RIVA_ASSET_TYPE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    </sequence>
</extension>
</complexType>
</schema>
</types>

```

```

- <wsdl:message name="SelectAssetDataResponse">
  <wsdl:part name="parameters"
element="tnsl:SelectAssetDataResponse" />
</wsdl:message>
- <wsdl:message name="InsertAssetDataRequest">
  <wsdl:part name="parameters" element="tnsl:InsertAssetData" />
</wsdl:message>
- <wsdl:message name="UpdateAssetDataRequest">
  <wsdl:part name="parameters" element="tnsl:UpdateAssetData" />
</wsdl:message>
- <wsdl:message name="SelectAssetDataRequest">
  <wsdl:part name="parameters" element="tnsl:SelectAssetData" />
</wsdl:message>
- <wsdl:message name="DeleteAssetDataResponse">
  <wsdl:part name="parameters"
element="tnsl>DeleteAssetDataResponse" />
</wsdl:message>
- <wsdl:message name="InsertAssetDataResponse">
  <wsdl:part name="parameters"
element="tnsl:InsertAssetDataResponse" />
</wsdl:message>
- <wsdl:message name="UpdateAssetDataResponse">
  <wsdl:part name="parameters"
element="tnsl:UpdateAssetDataResponse" />
</wsdl:message>
- <wsdl:message name="DeleteAssetDataRequest">
  <wsdl:part name="parameters" element="tnsl>DeleteAssetData" />
</wsdl:message>
- <wsdl:portType name="AssetTables">
- <wsdl:operation name="InsertAssetData">
  <wsdl:input name="InsertAssetDataRequest"
message="impl:InsertAssetDataRequest" />
  <wsdl:output name="InsertAssetDataResponse"
message="impl:InsertAssetDataResponse" />
</wsdl:operation>
- <wsdl:operation name="UpdateAssetData">
  <wsdl:input name="UpdateAssetDataRequest"
message="impl:UpdateAssetDataRequest" />
  <wsdl:output name="UpdateAssetDataResponse"
message="impl:UpdateAssetDataResponse" />
</wsdl:operation>
- <wsdl:operation name="DeleteAssetData">
  <wsdl:input name="DeleteAssetDataRequest"
message="impl>DeleteAssetDataRequest" />
  <wsdl:output name="DeleteAssetDataResponse"
message="impl>DeleteAssetDataResponse" />
</wsdl:operation>
- <wsdl:operation name="SelectAssetData">
  <wsdl:input name="SelectAssetDataRequest"
message="impl>SelectAssetDataRequest" />
  <wsdl:output name="SelectAssetDataResponse"
message="impl>SelectAssetDataResponse" />
</wsdl:operation>
</wsdl:portType>
- <wsdl:binding name="AssetTablesSoapBinding"
type="impl:AssetTables">
  <wsdlsoap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="InsertAssetData">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="InsertAssetDataRequest">
  <wsdlsoap:body use="literal" />
</wsdl:input>
- <wsdl:output name="InsertAssetDataResponse">
  <wsdlsoap:body use="literal" />
</wsdl:output>

```

```

        </wsdl:operation>
    - <wsdl:operation name="UpdateAssetData">
      <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="UpdateAssetDataRequest">
      <wsdlsoap:body use="literal" />
    - </wsdl:input>
    - <wsdl:output name="UpdateAssetDataResponse">
      <wsdlsoap:body use="literal" />
    - </wsdl:output>
    - </wsdl:operation>
    - <wsdl:operation name="DeleteAssetData">
      <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="DeleteAssetDataRequest">
      <wsdlsoap:body use="literal" />
    - </wsdl:input>
    - <wsdl:output name="DeleteAssetDataResponse">
      <wsdlsoap:body use="literal" />
    - </wsdl:output>
    - </wsdl:operation>
    - <wsdl:operation name="SelectAssetData">
      <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="SelectAssetDataRequest">
      <wsdlsoap:body use="literal" />
    - </wsdl:input>
    - <wsdl:output name="SelectAssetDataResponse">
      <wsdlsoap:body use="literal" />
    - </wsdl:output>
    - </wsdl:operation>
    - </wsdl:binding>
    - <wsdl:service name="AssetTablesService">
    - <wsdl:port name="AssetTables"
binding="impl:AssetTablesSoapBinding">
      <wsdlsoap:address
location="http://bali:1751/synergen/services/AssetTables" />
    - </wsdl:port>
    - </wsdl:service>
    - </wsdl:definitions>

```

## Asset Specification Web Services

This section describes the implementation of the web service that enables external applications to insert, select, update, and delete Specification records in the Oracle Utilities Work and Asset Management application. This web service was created using Apache Axis.

### Web Services Methods

Oracle Utilities Work and Asset Management uses separate header and details blocks describe a Specification record, where a single header can be associated with any number of details.

The Specifications web service exposes eight methods to external applications:

- InsertSpecificationData - used for importing specification header information into Oracle Utilities Work and Asset Management
- UpdateSpecificationData - used for updating specification header information in Oracle Utilities Work and Asset Management

- DeleteSpecificationData - used for deleting specification header information from Oracle Utilities Work and Asset Management
- SelectSpecificationData - used for selecting specification header information in Oracle Utilities Work and Asset Management
- InsertSpecificationDetailsData - used for importing specification detail information into Oracle Utilities Work and Asset Management
- UpdateSpecificationDetailsData - used for updating specification detail information in Oracle Utilities Work and Asset Management
- DeleteSpecificationDetailsData - used for deleting specification detail information from Oracle Utilities Work and Asset Management
- SelectSpecificationDetailsData - used for selecting specification detail information in Oracle Utilities Work and Asset Management

Each of these methods accepts an array of data types related to specification records.

After the submitted entries are processed, the method returns a confirmation message. All imported entries will be processed by the Oracle Utilities Work and Asset Management application from an interface table.

Please see the WSDL section of this document for a complete description of the web service elements and structure.

## Web Service Security

Oracle Utilities Work and Asset Management web services require no specific roles or responsibilities.

## Web Service URL

The web service is deployed to the following environment.

`http://server:port/synergen/services/SpecTables`  
for example:

`http://bali:1751/synergen/services/SpecTables`

## Specification Tables Data Type

Attribute	Description	Sys Req?	Prime?	Null?
PLANT	VARCHAR2 (3)			N
SPECIFICATION_NO	VARCHAR2 (10)		Y	N
SPECIFICATION_TYPE	VARCHAR2 (10)	Y		N
SPECIFICATION_CATEGORY	VARCHAR2 (15)	Y		N
SPECIFICATION_DESC	VARCHAR2 (2000)			
LAST_UPDATE_DATE	DATE			
LAST_UPDATE_USER	VARCHAR2 (30)			
CREATED_DATE	DATE			
CREATED_BY	VARCHAR2 (30)			
ATTRIBUTE1	User Defined VARCHAR2 (1)			
ATTRIBUTE2	User Defined VARCHAR2 (5)			
ATTRIBUTE3	User Defined VARCHAR2 (15)			
ATTRIBUTE4	User Defined VARCHAR2 (1)			
ATTRIBUTE5	User Defined VARCHAR2 (1)			
ATTRIBUTE6	User Defined VARCHAR2 (1)			
ATTRIBUTE7	User Defined VARCHAR2 (1)			
ATTRIBUTE8	User Defined VARCHAR2 (1)			
ATTRIBUTE9	User Defined VARCHAR2 (1)			
ATTRIBUTE10	User Defined VARCHAR2 (1)			

## Specification Tables Detail Data Type

Attribute	Description	Sys Req?	Prime?	Null?
PLANT	VARCHAR2 (3)			N
SPECIFICATION_NO	VARCHAR2 (10)			N
SPECIFICATION_SEQUENCE_NO	NUMBER (4)	Y		N
ATTRIBUTE_DESC	VARCHAR2 (60)			
ATTRIBUTE_VALUE	VARCHAR2 (50)			
LAST_UPDATE_DATE	DATE			
LAST_UPDATE_USER	VARCHAR2 (30)			
CREATED_DATE	DATE			
CREATED_BY	VARCHAR2 (30)			
SOURCE_INFORMATION	VARCHAR2 (50)			
SPECIFICATION_LOV	VARCHAR2 (30)			
SPECIFICATION_CODE_TABLE	NUMBER (4)			
SPECIFICATION_QUERY	VARCHAR2 (2000)			

## Web Service Definition Language (WSDL)

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions
targetNamespace="http://bali:1751/synergen/services/SpecTables"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns1="http://webservice.synergen" xmlns:tns2="urn:Spec"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:intf="http://bali:1751/synergen/services/SpecTables"
xmlns:impl="http://bali:1751/synergen/services/SpecTables">
- <!--
WSDL created by Apache Axis version: 1.2RC3
Built on Feb 28, 2005 (10:15:14 EST)
-->
- <types>
- <schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://webservice.synergen"
elementFormDefault="qualified">
<import namespace="urn:Spec" />
- <element name="InsertSpecificationDataData">
- <complexType>
- <sequence>
<element name="in0" type="tns2:SpecificationDataData" />
</sequence>
</complexType>
</element>
- <complexType abstract="true" name="WebServiceData">
<sequence />
</complexType>
- <element name="InsertSpecificationDataDataResponse">
- <complexType>
- <sequence>
<element name="InsertSpecificationDataDataReturn"
type="tns2:SpecificationDataData" />
</sequence>
</complexType>
</element>
- <element name="UpdateSpecificationDataData">
- <complexType>
- <sequence>
<element name="in0" type="tns2:SpecificationDataData" />
</sequence>
</complexType>
</element>
- <element name="UpdateSpecificationDataDataResponse">
- <complexType>
- <sequence>

```

```

        <element name="UpdateSpecificationDataDataReturn"
type="xsd:int" />
    </sequence>
</complexType>
</element>
- <element name="DeleteSpecificationDataData">
- <complexType>
- <sequence>
    <element name="in0" type="tns2:SpecificationDataData" />
</sequence>
</complexType>
</element>
- <element name="DeleteSpecificationDataDataResponse">
- <complexType>
- <sequence>
    <element name="DeleteSpecificationDataDataReturn"
type="xsd:int" />
    </sequence>
</complexType>
</element>
- <element name="SelectSpecificationDataData">
- <complexType>
- <sequence>
    <element name="in0" type="tns2:SpecificationDataData" />
</sequence>
</complexType>
</element>
- <element name="SelectSpecificationDataDataResponse">
- <complexType>
- <sequence>
    <element name="SelectSpecificationDataDataReturn"
type="tns2:SpecificationDataData" maxOccurs="unbounded" />
    </sequence>
</complexType>
</element>
- <element name="InsertSpecificationDetailsData">
- <complexType>
- <sequence>
    <element name="in0" type="tns2:SpecificationDetailsData" />
</sequence>
</complexType>
</element>
- <element name="InsertSpecificationDetailsDataResponse">
- <complexType>
- <sequence>
    <element name="InsertSpecificationDetailsDataReturn"
type="tns2:SpecificationDetailsData" />
    </sequence>
</complexType>
</element>
- <element name="UpdateSpecificationDetailsData">
- <complexType>
- <sequence>
    <element name="in0" type="tns2:SpecificationDetailsData" />
</sequence>
</complexType>
</element>
- <element name="UpdateSpecificationDetailsDataResponse">
- <complexType>
- <sequence>
    <element name="UpdateSpecificationDetailsDataReturn"
type="xsd:int" />
    </sequence>
</complexType>
</element>
- <element name="DeleteSpecificationDetailsData">

```

```

- <complexType>
- <sequence>
  <element name="in0" type="tns2:SpecificationDetailsData" />
</sequence>
</complexType>
</element>
- <element name="DeleteSpecificationDetailsDataResponse">
- <complexType>
- <sequence>
  <element name="DeleteSpecificationDetailsDataReturn"
type="xsd:int" />
</sequence>
</complexType>
</element>
- <element name="SelectSpecificationDetailsData">
- <complexType>
- <sequence>
  <element name="in0" type="tns2:SpecificationDetailsData" />
</sequence>
</complexType>
</element>
- <element name="SelectSpecificationDetailsDataResponse">
- <complexType>
- <sequence>
  <element name="SelectSpecificationDetailsDataReturn"
type="tns2:SpecificationDetailsData" maxOccurs="unbounded" />
</sequence>
</complexType>
</element>
</schema>
- <schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:Spec" elementFormDefault="qualified">
  <import namespace="http://webservice.synergen" />
  <complexType name="SpecificationDataData">
  <complexContent>
  <extension base="tns1:WebServiceData">
  <sequence>
    <element name="PLANT" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="SPECIFICATION_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="SPECIFICATION_TYPE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
    <element name="SPECIFICATION_CATEGORY" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
    <element name="SPECIFICATION_DESC" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
    <element name="LAST_UPDATE_DATE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:dateTime" />
    <element name="LAST_UPDATE_USER" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="CREATED_DATE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:dateTime" />
    <element name="CREATED_BY" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="ATTRIBUTE1" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="ATTRIBUTE2" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="ATTRIBUTE3" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="ATTRIBUTE4" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="ATTRIBUTE5" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />

```



```

        <element name="ATTRIBUTE6" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE7" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE8" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE9" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
        <element name="ATTRIBUTE10" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    </sequence>
</extension>
</complexContent>
</complexType>
- <complexType name="SpecificationDetailsData">
- <complexContent>
- <extension base="tnsl:WebServiceData">
- <sequence>
    <element name="PLANT" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="SPECIFICATION_NO" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="SPECIFICATION_SEQUENCE_NO" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:int" />
    <element name="ATTRIBUTE_DESC" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="ATTRIBUTE_VALUE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="LAST_UPDATE_DATE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:dateTime" />
    <element name="LAST_UPDATE_USER" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="CREATED_DATE" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:dateTime" />
    <element name="CREATED_BY" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="SOURCE_INFORMATION" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
    <element name="SPECIFICATION_LOV" nillable="true" minOccurs="0"
maxOccurs="1" type="xsd:string" />
    <element name="SPECIFICATION_CODE_TABLE" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:int" />
    <element name="SPECIFICATION_QUERY" nillable="true"
minOccurs="0" maxOccurs="1" type="xsd:string" />
    </sequence>
</extension>
</complexContent>
</complexType>
</schema>
</types>
- <wsdl:message name="UpdateSpecificationDataDataRequest">
    <wsdl:part name="parameters"
element="tnsl:UpdateSpecificationDataData" />
</wsdl:message>
- <wsdl:message name="InsertSpecificationDetailsDataRequest">
    <wsdl:part name="parameters"
element="tnsl:InsertSpecificationDetailsData" />
</wsdl:message>
- <wsdl:message name="InsertSpecificationDataDataRequest">
    <wsdl:part name="parameters"
element="tnsl:InsertSpecificationDataData" />
</wsdl:message>
- <wsdl:message name="DeleteSpecificationDetailsDataResponse">
    <wsdl:part name="parameters"
element="tnsl:DeleteSpecificationDetailsDataResponse" />
</wsdl:message>

```

```

- <wsdl:message name="InsertSpecificationDataDataResponse">
  <wsdl:part name="parameters"
element="tnsl:InsertSpecificationDataDataResponse" />
</wsdl:message>
- <wsdl:message name="InsertSpecificationDetailsDataResponse">
  <wsdl:part name="parameters"
element="tnsl:InsertSpecificationDetailsDataResponse" />
</wsdl:message>
- <wsdl:message name="DeleteSpecificationDataDataResponse">
  <wsdl:part name="parameters"
element="tnsl:DeleteSpecificationDataDataResponse" />
</wsdl:message>
- <wsdl:message name="SelectSpecificationDataDataRequest">
  <wsdl:part name="parameters"
element="tnsl:SelectSpecificationDataData" />
</wsdl:message>
- <wsdl:message name="SelectSpecificationDetailsDataRequest">
  <wsdl:part name="parameters"
element="tnsl:SelectSpecificationDetailsData" />
</wsdl:message>
- <wsdl:message name="UpdateSpecificationDetailsDataRequest">
  <wsdl:part name="parameters"
element="tnsl:UpdateSpecificationDetailsData" />
</wsdl:message>
- <wsdl:message name="DeleteSpecificationDataDataRequest">
  <wsdl:part name="parameters"
element="tnsl:DeleteSpecificationDataData" />
</wsdl:message>
- <wsdl:message name="UpdateSpecificationDataDataResponse">
  <wsdl:part name="parameters"
element="tnsl:UpdateSpecificationDataDataResponse" />
</wsdl:message>
- <wsdl:message name="DeleteSpecificationDetailsDataRequest">
  <wsdl:part name="parameters"
element="tnsl:DeleteSpecificationDetailsData" />
</wsdl:message>
- <wsdl:message name="SelectSpecificationDetailsDataResponse">
  <wsdl:part name="parameters"
element="tnsl:SelectSpecificationDetailsDataResponse" />
</wsdl:message>
- <wsdl:message name="SelectSpecificationDataDataResponse">
  <wsdl:part name="parameters"
element="tnsl:SelectSpecificationDataDataResponse" />
</wsdl:message>
- <wsdl:message name="UpdateSpecificationDetailsDataResponse">
  <wsdl:part name="parameters"
element="tnsl:UpdateSpecificationDetailsDataResponse" />
</wsdl:message>
- <wsdl:portType name="SpecTables">
- <wsdl:operation name="InsertSpecificationDataData">
  <wsdl:input name="InsertSpecificationDataDataRequest"
message="impl:InsertSpecificationDataDataRequest" />
  <wsdl:output name="InsertSpecificationDataDataResponse"
message="impl:InsertSpecificationDataDataResponse" />
</wsdl:operation>
- <wsdl:operation name="UpdateSpecificationDataData">
  <wsdl:input name="UpdateSpecificationDataDataRequest"
message="impl:UpdateSpecificationDataDataRequest" />
  <wsdl:output name="UpdateSpecificationDataDataResponse"
message="impl:UpdateSpecificationDataDataResponse" />
</wsdl:operation>
- <wsdl:operation name="DeleteSpecificationDataData">
  <wsdl:input name="DeleteSpecificationDataDataRequest"
message="impl:DeleteSpecificationDataDataRequest" />
  <wsdl:output name="DeleteSpecificationDataDataResponse"
message="impl:DeleteSpecificationDataDataResponse" />

```

```

        </wsdl:operation>
        - <wsdl:operation name="SelectSpecificationDataData">
            <wsdl:input name="SelectSpecificationDataDataRequest"
message="impl:SelectSpecificationDataDataRequest" />
            <wsdl:output name="SelectSpecificationDataDataResponse"
message="impl:SelectSpecificationDataDataResponse" />
        </wsdl:operation>
        - <wsdl:operation name="InsertSpecificationDetailsData">
            <wsdl:input name="InsertSpecificationDetailsDataRequest"
message="impl:InsertSpecificationDetailsDataRequest" />
            <wsdl:output name="InsertSpecificationDetailsDataResponse"
message="impl:InsertSpecificationDetailsDataResponse" />
        </wsdl:operation>
        - <wsdl:operation name="UpdateSpecificationDetailsData">
            <wsdl:input name="UpdateSpecificationDetailsDataRequest"
message="impl:UpdateSpecificationDetailsDataRequest" />
            <wsdl:output name="UpdateSpecificationDetailsDataResponse"
message="impl:UpdateSpecificationDetailsDataResponse" />
        </wsdl:operation>
        - <wsdl:operation name="DeleteSpecificationDetailsData">
            <wsdl:input name="DeleteSpecificationDetailsDataRequest"
message="impl:DeleteSpecificationDetailsDataRequest" />
            <wsdl:output name="DeleteSpecificationDetailsDataResponse"
message="impl:DeleteSpecificationDetailsDataResponse" />
        </wsdl:operation>
        - <wsdl:operation name="SelectSpecificationDetailsData">
            <wsdl:input name="SelectSpecificationDetailsDataRequest"
message="impl:SelectSpecificationDetailsDataRequest" />
            <wsdl:output name="SelectSpecificationDetailsDataResponse"
message="impl:SelectSpecificationDetailsDataResponse" />
        </wsdl:operation>
    </wsdl:portType>
    - <wsdl:binding name="SpecTablesSoapBinding"
type="impl:SpecTables">
        <wsdlsoap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
        - <wsdl:operation name="InsertSpecificationDataData">
            <wsdlsoap:operation soapAction="" />
            - <wsdl:input name="InsertSpecificationDataDataRequest">
                <wsdlsoap:body use="literal" />
            </wsdl:input>
            - <wsdl:output name="InsertSpecificationDataDataResponse">
                <wsdlsoap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        - <wsdl:operation name="UpdateSpecificationDataData">
            <wsdlsoap:operation soapAction="" />
            - <wsdl:input name="UpdateSpecificationDataDataRequest">
                <wsdlsoap:body use="literal" />
            </wsdl:input>
            - <wsdl:output name="UpdateSpecificationDataDataResponse">
                <wsdlsoap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        - <wsdl:operation name="DeleteSpecificationDataData">
            <wsdlsoap:operation soapAction="" />
            - <wsdl:input name="DeleteSpecificationDataDataRequest">
                <wsdlsoap:body use="literal" />
            </wsdl:input>
            - <wsdl:output name="DeleteSpecificationDataDataResponse">
                <wsdlsoap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        - <wsdl:operation name="SelectSpecificationDataData">
            <wsdlsoap:operation soapAction="" />
            - <wsdl:input name="SelectSpecificationDataDataRequest">

```

```

        <wsdlsoap:body use="literal" />
    </wsdl:input>
    - <wsdl:output name="SelectSpecificationDataDataResponse">
        <wsdlsoap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    - <wsdl:operation name="InsertSpecificationDetailsData">
        <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="InsertSpecificationDetailsDataRequest">
        <wsdlsoap:body use="literal" />
    </wsdl:input>
    - <wsdl:output name="InsertSpecificationDetailsDataResponse">
        <wsdlsoap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    - <wsdl:operation name="UpdateSpecificationDetailsData">
        <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="UpdateSpecificationDetailsDataRequest">
        <wsdlsoap:body use="literal" />
    </wsdl:input>
    - <wsdl:output name="UpdateSpecificationDetailsDataResponse">
        <wsdlsoap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    - <wsdl:operation name="DeleteSpecificationDetailsData">
        <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="DeleteSpecificationDetailsDataRequest">
        <wsdlsoap:body use="literal" />
    </wsdl:input>
    - <wsdl:output name="DeleteSpecificationDetailsDataResponse">
        <wsdlsoap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    - <wsdl:operation name="SelectSpecificationDetailsData">
        <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="SelectSpecificationDetailsDataRequest">
        <wsdlsoap:body use="literal" />
    </wsdl:input>
    - <wsdl:output name="SelectSpecificationDetailsDataResponse">
        <wsdlsoap:body use="literal" />
    </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
    - <wsdl:service name="SpecTablesService">
    - <wsdl:port name="SpecTables"
binding="impl:SpecTablesSoapBinding">
        <wsdlsoap:address
location="http://bali:1751/synergen/services/SpecTables" />
    </wsdl:port>
    </wsdl:service>
</wsdl:definitions>

```

## GIS Interfaces for Assets and Specifications

The GIS specification interface is called WIFP\_GIS\_ASSET\_SPECIFICATION. It uses the GIS Specification Interface business rule and the SAIF\_GIS\_SPECIFICATION\_XREF table. The interface updates SA\_SPECIFICATION\_DATA and SA\_SPECIFICATION\_DETAILS.

No example view is delivered for this interface. Processing is as follows:

1. Read the GIS Specification Interface business rule for the view name.
2. Open a cursor to the view outer-joined with saif\_gis\_specification\_xref.

3. If no specification number associated with the SYNERGEN\_ID for a record, then try to create specification for it. Otherwise, it will use the specification number associated to update the spec details.
4. If a specification is created, insert a new row into saif\_gis\_specification\_xref table to create the association. If the specification is created, assume/use the SYNERGEN\_ID as the asset to create an attachment on the asset (insert into sa\_asset\_attachment) and update the asset header with the specification number.
5. The batch job, WIFP\_GIS\_ASSET\_SPECIFICATION, moves GIS Specification records from the views of GIS tables listed in the GIS Specification Interface.

### GIS Specification View

Each GIS Specification view must contain:

- SYNERGENID - This contains: asset\_record\_type || '-' || asset\_id from GIS Table

It can be an alias

- field1 - Spec data (FORMAT IS 'sequence no-' || GIS\_FIELD)

Example: '1-' || pipe\_length

- field2
- fieldN

The GIS Specification view may also use LAST\_RUN\_DATE from the business rule to only pull records that have changed since the last run. In this case a timestamp would have to be added to the GIS source table.

### GIS Specification Interface Business Rule

This rule shows the list of GIS Views to be processed by the GIS specification interface. The GIS View must match SV\_GIS\_SPECIFICATION\_LAYOUT. GIS View records will be copied directly to SA\_SPECIFICATION. Then the GIS Table will be updated with the newly created Specification ID (SPECIFICATION\_RECORD\_TYPE - SPECIFICATION\_ID).

SAIF\_GIS\_SPECIFICATION\_XREF uses this rule to execute the mapping between GIS data and Oracle Utilities Work and Asset Management Specifications.

- KEY\_NAME = GIS\_SPEC\_VIEW\_NAME
- KEY\_VALUE = SPEC\_TYPE
- KEY\_VALUE2 = SPEC\_CAT
- KEY\_VALUE3 = LAST\_RUN\_DATE (FORMAT IS 'DD-MON-YYYY HH:MI:SS')

**GIS Asset Interface Business Rule**

This rule shows the list of GIS Views to be processed by the GIS Asset Interface. The GIS View must match SV\_GIS\_ASSET\_LAYOUT. GIS View records will be copied directly to SA\_ASSET. GIS Table will be updated with the newly created Asset ID (ASSET\_RECORD\_TYPE - ASSET\_ID).

WIFP\_GIS\_ASSET uses this rule to execute the mapping between GIS data and Oracle Utilities Work and Asset Management Assets.

---

# Chapter 6

## Blanket Contract

The Blanket Contract interface provides means by which Oracle Utilities Work & Asset Management can interface blanket contract data with other systems.

This interface requires:

**Interface Table:** WAIF\_BLANKET\_CONTRACT

**Stored Procedure:** WIFP\_BLANKET\_CONTRACT\_INTERFAC

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Job Manager according to the client's requirements.

```
WIFP_BLANKET_CONTRACT_INTERFAC(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN  
PRE_IN,  
POST_IN  
)
```

Cancelled lines are not created in the application. Inbound information must be a new revision, not an update. For a new revision of an existing blanket contract, default accounts, notes, access list and attachments are all copied to the new revision. A processing option allows you to bring forward the current “used” amount. The new revision is activated after creation.

When running the interface outbound, only ACTIVE blanket contracts will be processed. ALL Active blanket contracts will be written to the interface table. Therefore, when using the outbound direction, you should always use Purge = 'Y'.

### Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management Job Manager.

**PLANT\_IN** = VARCHAR2 - identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to the Oracle Utilities Work & Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work & Asset Management application to the interface table.

**PURGE\_IN** = VARCHAR2 - identifies how records will be removed from the interface table.

**Y** = Records that have already been processed will be deleted from the interface table before processing begins.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, until some other process deletes the records or until this interface is run with PURGE\_IN set to Y.

**OPTION\_IN** = VARCHAR2 - (Optional) - In this interface, the option\_in parameter indicates whether the current Blanket Used Amount should be retrieved and stored from the application table instead of using the value from the interface table.

**Y** = the value should be retrieved from the application table. Anything else (including null) indicates the value in the interface table should be used.

**PRE\_IN** = VARCHAR2 (Optional) - Identifies the custom stored procedure to call before the interface code executes.

**POST\_IN** = VARCHAR2 (Optional) - Identifies the custom stored procedure to call after the interface code executes.

## Sequence Numbers

SA\_BLANKET\_CONTRACT

## Interface Table Layout

WAIF\_BLANKET\_CONTRACT

(same fields as SA\_BLANKET\_CONTRACT, SA\_BLANKET\_CONTRACT\_ITEM, and SA\_BLANKET\_CONTRACT\_ITEM\_ACCTS)

The following fields are available for ALL records:	
PLANT*	VARCHAR2(3)
BLANKET_CONTRACT_NO*	VARCHAR2(10)
BLANKET_REVISION_NO*	VARCHAR2(3)
DIRECTION*	CHAR(1)
RECORD_TYPE*	CHAR(1)
JOB_SEQ_NO	NUMBER
SENT_TO_INTERFACE_IND	CHAR(1)
The following fields are available for Header records	



BLANKET_RELEASE_NO	VARCHAR2(4)
BLANKET_STATUS*	VARCHAR2(20)
BLANKET_STATUS_DATE*	DATE
BLANKET_CATEGORY	VARCHAR2(10)
QUOTE_NO	VARCHAR2(8)
VENDOR_CODE	VARCHAR2(30)
BLANKET_EXPIRY_DATE	DATE
BLANKET_EXTENDED_DATE	DATE
BLANKET_USED_AMOUNT	NUMBER(15,2)
BLANKET_LIMIT_AMOUNT	NUMBER(15,2)
CARRIER	VARCHAR2(30)
BLANKET_DESC	VARCHAR2(2000)
BUYER	VARCHAR2(3)
ENFORCE_LIMIT	CHAR(1)
STANDARD_TEXT	CHAR(1)
FOB	VARCHAR2(20)
TERMS	VARCHAR2(2)
ATTRIBUTE1	USER DEFINED
ATTRIBUTE2	USER DEFINED
ATTRIBUTE3	USER DEFINED
ATTRIBUTE4	USER DEFINED
ATTRIBUTE5	USER DEFINED
ATTRIBUTE6	USER DEFINED
ATTRIBUTE7	USER DEFINED
ATTRIBUTE8	USER DEFINED
ATTRIBUTE9	USER DEFINED
ATTRIBUTE10	USER DEFINED
WORK_ORDER_RELEASE_IND	CHAR(1)
DELIVERY_LEAD_TIME	NUMBER(4)
DELIVERY_LEAD_TIME_UNITS	VARCHAR2(6)
EDI_IND	VARCHAR2(1)
REQUISITION_NO	VARCHAR2(7)
AUTOFAX_IND	VARCHAR2(1)

BLANKET_INITIATION_DATE	DATE
AUTO_PAY_IND	CHAR(1)
VENDOR_CLASS	VARCHAR2(5)
TEMPLATE_IND	VARCHAR2(1)
DEFAULT_ACCOUNTS_IND	VARCHAR2(1)
DISCOUNT_CONTRACT_IND	VARCHAR2(1)
USE_MASTER_ACCESS_LIST	CHAR(1)
REVISION_WITH_HISTORY_IND	VARCHAR2(1)
The following fields are available for Header and Line records	
LAST_UPDATE_DATE*	DATE
CREATION_DATE*	DATE
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_BY*	VARCHAR2(30)
The following fields are available for Line and Account records	
BLANKET_ITEM*	VARCHAR2(5)
The following fields are available on Line records	
BLANKET_ITEM_STATUS*	VARCHAR2(20)
STOCK_CODE	VARCHAR2(15)
STOCK_TYPE	VARCHAR2(15)
ITEM_QUANTITY	NUMBER(15,5)
UNIT_PRICE	NUMBER(17,4)
ITEM_TOTAL_AMOUNT	NUMBER(15,2)
FEDERAL_TAX_CODE	CHAR(1)
STATE_PROVINCE_TAX_CODE	CHAR(1)
DUTY_CODE	CHAR(1)
ITEM_DESC	VARCHAR2(2000)
UNIT_OF_PURCHASE	VARCHAR2(4)
MANUFACTURER_CODE	VARCHAR2(20)
MANUFACTURER_MODEL_NO	VARCHAR2(30)
MANUFACTURER_PART_NO	VARCHAR2(50)
MANUFACTURER_DRAWING_NO	VARCHAR2(15)
VENDOR_MODEL_NO	VARCHAR2(30)
VENDOR_PART_NO	VARCHAR2(50)

LINE_ATTRIBUTE1	USER-DEFINED
LINE_ATTRIBUTE2	USER-DEFINED
LINE_ATTRIBUTE3	USER-DEFINED
LINE_ATTRIBUTE4	USER-DEFINED
LINE_ATTRIBUTE5	USER-DEFINED
LINE_ATTRIBUTE6	USER-DEFINED
LINE_ATTRIBUTE7	USER-DEFINED
LINE_ATTRIBUTE8	USER-DEFINED
LINE_ATTRIBUTE9	USER-DEFINED
LINE_ATTRIBUTE10	USER-DEFINED
QUOTE_ITEM	VARCHAR2(5)
PROCUREMENT_LEVEL	VARCHAR2(5)
QUALITY_CLASS	VARCHAR2(5)
QUALITY_IND	VARCHAR2(1)
PI_RATIO	NUMBER
LEAD_TIME	NUMBER(3)
ITEM_SEQUENCE_NUMBER	NUMBER
DISCOUNT	NUMBER(7,4)
ASSET_ID	VARCHAR2(15)
ASSET_RECORD_TYPE	VARCHAR2(1)
COMPONENT_ID	VARCHAR2(15)
The following columns are available on Account records	
ACCOUNT_NO*	VARCHAR2(75)
EXPENSE_CODE*	VARCHAR2(10)
PERCENT_SPLIT	NUMBER(8,5)
SPLIT_AMOUNT	NUMBER(15,2)
UNITS	NUMBER(12,5)
DIRECTION	CHAR(1)
RECORD_TYPE	CHAR(1)
JOB_SEQ_NO	NUMBER
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)

JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

\*Required fields

---

# Chapter 7

## Budget

The Budget interface provides a means to import and export changes to the budgetary amounts recorded in Oracle Utilities Work & Asset Management.

This interface requires:

**Interface Table:** WAIF\_ACCOUNT\_COST

**Stored Procedure:** WIFP\_BUDGET\_INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management Job Manager according to the client's requirements.

```
WIFP_BUDGET_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

On outbound, purge deletes all outbound records from the plant, regardless of the sent flag, and sends all new records. For this reason, you should always use purge on outbound. On inbound, purge only removes processed records.

On inbound only the budget amount is used when updating, and it's a delta value. It should contain the amount (positive or negative) to change the current budget amount.

### Keywords and Parameters

**JOB\_IN = INTEGER-** The job number assigned by the Oracle Utilities Work and Asset Management Job Manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to the Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table.

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y** = If **DIRECTION = I**, records will be deleted from the interface table as they are processed. If **DIRECTION = O**, records will be deleted from the interface table before writing out new records. When **DIRECTION = O**, **PURGE\_IN** should always be set to **Y**.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before **WIFP\_BUDGET\_INTERFACE** runs again.

**OPTION\_IN = VARCHAR2**. Not used for this interface.

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins. See Custom Stored Procedure Note below for requirements.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Interface Table Layout

WAIF\_ACCOUNT\_COST (same fields as SA\_ACCOUNT\_COST)

Field Name	Data Type	Description
PLANT*	VARCHAR2(3)	
ACCOUNT_NO*	VARCHAR2(75)	
EXPENSE_CODE*	VARCHAR2(10)	
PERIOD_YEAR*	CHAR(4)	
PERIOD_MONTH*	CHAR(2)	
BUDGET_AMOUNT	NUMBER(15,2)	This is the only field affected on update. It is always <i>added</i> to the existing budget amount. (value should be a delta).
ACTUAL_AMOUNT	NUMBER(15,2)	Only used on insert.
COMMITTED_AMOUNT	NUMBER(15,2)	Only used on insert.
SENT_TO_INTERFACE_IND	CHAR(1)	
JOB_SEQ_NO	NUMBER	
DIRECTION*	CHAR(1)	
SOURCE_SYSTEM	VARCHAR2(3)	
INT_BATCH_NUMBER	NUMBER	
JOB_STATUS	VARCHAR2(15)	
JOB_MESSAGE	VARCHAR2(4000)	

Field Name	Data Type	Description
INT_INSTANCE_ID	NUMBER	

\* required fields

---

# Chapter 8

## Catalog

The Catalog Interface provides a means by which Oracle Utilities Work and Asset Management can interface master catalog data with other systems.

This interface requires:

**Interface Table:** WAIF\_CATALOG

**Stored Procedure:** WIFP\_CATALOG\_INTERFACE

**Business Rule:** CATALOG INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_CATALOG_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

Only processed records are purged.

The job number is used to determine which records to inactivate, if the Inactivate option is chosen.

On outbound, if you select the Update option only those records updated after the last job run are interfaced. Unprocessed records are not purged, so you may have more than one unprocessed record for the stock code in the table.

If you do not select the Update option, purge "Y" will remove all existing outbound records and replace them with new ones. Unless you are sending only updates, you should always set the purge flag to "Y".



## Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to the Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table.

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y** = If DIRECTION = **I**, records will be deleted from the interface table as they are processed. If DIRECTION = **O**, records will be deleted from the interface table before writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_CATALOG\_INTERFACE runs again.

See above section for more on using Purge.

**OPTION\_IN = VARCHAR2** - (Optional) The OPTION\_IN parameter allows control of what type of data is processed.

**If DIRECTION = I and...**

- **OPTIONS = I** - The interface procedure will 'Inactivate' all pre-existing catalog records in the Oracle Utilities Work and Asset Management application that were not created, or updated, by the current run of the interface procedure. This would be assuming that the interface table contains all the active catalog items and that all other catalog items are no longer active.

**If DIRECTION = O and...**

- **OPTIONS = A** – The interface procedure will **only** write 'Active' catalog records to the interface table.
- **OPTIONS = U** – The interface procedure will **only** write catalog records that have been updated since the last run of the interface procedure to the interface table.
- **OPTIONS = AU** – Both of the above options apply.

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_OUT = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Business Rule Format

This business rule only applies to pre-existing inbound records.

**RULE ID = CATALOG INTERFACE**

**RULE TYPE = INTERFACE**

**RULE STYLE** = LIST

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound catalog record that already exists in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_CATALOG column to be updated.

**UPDATE** = YES or NO. If this value is set to YES then the corresponding field will be updated.

## Sequence Numbers

SA\_CATALOG

## Interface Table Layout

### WAIF\_CATALOG (same fields as SA\_CATALOG)

Field Name	Data Type
PLANT*	VARCHAR2(3)
STOCK_CODE*	VARCHAR2(15)
STOCK_TYPE*	VARCHAR2(15)
LAST_UPDATE_DATE*	DATE
STOCK_DESC	VARCHAR2(2000)
SHELF_LIFE	NUMBER(4)
CAPITAL_IND	CHAR(1)
DO_NOT_SUBSTITUTE_IND	CHAR(1)
ALTERNATE_STOCK_CODE	VARCHAR2(15)
PI_RATIO	NUMBER
UNIT_OF_ISSUE	VARCHAR2(4)
LEAD_TIME	NUMBER(3)
HAZARD_IND	CHAR(1)
SUB_ASSEMBLY_IND	CHAR(1)
COMMODITY_CATEGORY	VARCHAR2(5)
COMMODITY_NAME	VARCHAR2(5)
COMMODITY_TYPE	VARCHAR2(5)
COMMODITY_COMPOSITION	VARCHAR2(10)
COMMODITY_SIZE	VARCHAR2(40)
PRIMARY_VENDOR_CODE	VARCHAR2(30)
UNIT_OF_PURCHASE	VARCHAR2(4)

Field Name	Data Type
STOCK_CLASS	VARCHAR2(10)
CONSIGNMENT_IND	CHAR(1)
MSDS_NO	VARCHAR2(15)
CREATED_DATE*	DATE
HAZARD_TYPE	VARCHAR2(10)
COMMODITY	VARCHAR2(20)
FEDERAL_TAX_CODE	CHAR(1)
STATE_PROVINCE_TAX_CODE	CHAR(1)
DUTY_CODE	CHAR(1)
BOM_IND	CHAR(1)
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_BY*	VARCHAR2(30)
ATTRIBUTE1	USER-DEFINED
ATTRIBUTE2	USER-DEFINED
ATTRIBUTE3	USER-DEFINED
ATTRIBUTE4	USER-DEFINED
ATTRIBUTE5	USER-DEFINED
ATTRIBUTE6	USER-DEFINED
ATTRIBUTE7	USER-DEFINED
ATTRIBUTE8	USER-DEFINED
ATTRIBUTE9	USER-DEFINED
ATTRIBUTE10	USER-DEFINED
BUYER	VARCHAR2(3)
RESTRICTED_ISSUE_IND	VARCHAR2(1)
CATALOG_STATUS*	VARCHAR2(20)
QUALITY_ITEM_IND	VARCHAR2(1)
INSPECTION_TYPE	VARCHAR2(20)
PROCUREMENT_LEVEL	VARCHAR2(5)
QUALITY_CLASS	VARCHAR2(5)
SHELF_LIFE_UNITS	VARCHAR2(6)
SHELF_LIFE_CLASS	VARCHAR2(10)
STORAGE_CODE	VARCHAR2(5)
SPECIAL_REQUIREMENT	VARCHAR2(5)

Field Name	Data Type
TEMPORARY_STOCK_IND	VARCHAR2(1)
SENT_TO_INTERFACE_IND	VARCHAR2(1)
JOB_SEQ_NO	NUMBER
CU_RECONCILE_IND	VARCHAR2(1)
TRUCK_STOCK_IND	VARCHAR2(1)
DIRECTION*	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

\* required fields

---

# Chapter 9

## Cost Adjustments

The Cost Adjustments interface allows external systems to send indirect charges and journal entries to the Oracle Utilities Work & Asset Management application. This interface also can send “POSTED” cost adjustments out to an external system.

This interface requires:

**Interface Table:** WAIF\_COST\_ADJUSTMENT

**Stored Procedure:** WIFP\_COST\_ADJUSTMENT\_INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client’s requirements.

```
WIFP_COST_ADJUSTMENT_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

The stored procedure processes both IN data, data being put into the Oracle Utilities Work and Asset Management Cost Adjustment module, and OUT data, data being sent from the Oracle Utilities Work and Asset Management Cost Adjustment module to an external application.

During outbound processing, only processed records are purged and only posted records are sent. During inbound processing, only processed are purged. The entire adjustment must be processed successfully, or it will be skipped.

### Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies which record type(s) to process.

**I** = Indicates that IN records in the interface table are to be processed.

All records in the interface table with DIRECTION = 'I' will be moved to the appropriate Oracle Utilities Work and Asset Management table as is.

**O** = Indicates that OUT records are to be written to the interface table.

Only adjustments in Posted status are sent.

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y** = Records will be deleted from the interface table after they are processed for IN records and before they are processed for OUT records.

**N** = Records will not be deleted from the interface table.

Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_COST\_ADJUSTMENT\_INTERFACE runs again.

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface process runs.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface process runs.

## Sequence Numbers

SA\_ADJUSTMENT

## Interface Table Layout

### WAIF\_COST\_ADJUSTMENT

(same fields as SA\_ADJUSTMENT and SA\_ADJUSTMENT\_ITEM)

Field Name	Data Type	Description
The following fields are required for both Header and Line records.		
DIRECTION	VARCHAR2(1)	Indicates an <b>I</b> N or <b>O</b> UT record
RECORD_TYPE	VARCHAR2(1)	Indicates a <b>H</b> header or <b>L</b> ine record
PLANT	VARCHAR2 (3)	Plant Code
ADJUSTMENT_ID	VARCHAR2 (20)	Adjustment Number
TRANSACTION_DATE	DATE	Transaction Date
The following fields are available for Header records		
LAST_UPDATE_DATE*	DATE	Last data the adjustment was updated
LAST_UPDATE_USER*	VARCHAR2 (30)	Last person who updated the adjustment

Field Name	Data Type	Description
ADJUSTMENT_STATUS	VARCHAR2 (20)	Status of the adjustment; generally, inbound adjustments should be APPROVED
ADJUSTMENT_STATUS_DATE	DATE	Date adjustment status was last changed
SEND_TO_GL_IND	CHAR(1)	Indicates that adjustment should be sent to the General Ledger
NEXT_APPROVER	VARCHAR2 (6)	Next Approver
INITIATOR	VARCHAR2 (30)	Initiator
CREATION_DATE	DATE	Date adjustment was created
ADJUSTMENT_COMMENTS	VARCHAR2 (2000)	Comments
SENT_TO_GL	VARCHAR2(1)	Not used
SENT_TO_INTERFACE_IND	VARCHAR2(1)	'Y' means a record has been processed by the interface, otherwise, 'N'
JOB_SEQ_NO	NUMBER	Current job's sequence number assigned by Job Manager
The following fields are available for Line records		
SEQUENCE_NO*	VARCHAR2(5)	Unique ID for adjustment lines
WORK_ORDER_NO	VARCHAR2(7)	Work Order Number
WORK_ORDER_TASK_NO	VARCHAR2(2)	Work Order Task Number
ACCOUNT_NO	VARCHAR2(75)	Account Number
EXPENSE_CODE	VARCHAR2(10)	Expense Code
DEPARTMENT	VARCHAR2(10)	Department
AREA	VARCHAR2(10)	Area
ADJUSTMENT_AMOUNT	NUMBER(15,2)	Adjustment Amount
ASSET_RECORD_TYPE	CHAR(1)	Asset Record Type
ASSET_ID	VARCHAR2(15)	Asset ID
ITEM_REFERENCE_ID	VARCHAR2(30)	Item Reference ID
ADJUSTMENT_QUANTITY	NUMBER(15,5)	Adjustment Quantity
DESCRIPTION	VARCHAR2(200)	Description
CRAFT	VARCHAR2(5)	Craft
SERVICE_REQUEST_NO	VARCHAR2(7)	Service Request Number
COMPONENT_ID	VARCHAR2(15)	Component ID
MULTI_ASSET_GENERATED	VARCHAR2(1)	
DIRECTION	CHAR(1)	

Field Name	Data Type	Description
SOURCE_SYSTEM	VARCHAR2(3)	
INT_BATCH_NUMBER	NUMBER	
JOB_STATUS	VARCHAR2(15)	
JOB_MESSAGE	VARCHAR2(4000 )	
INT_INSTANCE_ID	NUMBER	

\* required fields



---

# Chapter 10

## Customer

The Customer Interface provides a means by which customer information from other applications can be imported into Oracle Utilities Work & Asset Management.

This interface requires:

**Interface Table:** WAIF\_CUSTOMER

**Stored Procedure:** WIFP\_CUSTOMER\_INTERFACE

**Business Rule:** CUSTOMER INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_CUSTOMER_INTERFACE(  
JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN  
)
```

Customer records are never updated if they already exist in the application. The business rule applies to updates to the Customer Address records. Header records are necessary whenever address information is given. All headers and address information for the customer is considered one transaction. Either all information for the customer will be interfaced, or all the information will be skipped.

### Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application. Customer information is only inserted. Address information can be inserted or updated.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not yet implemented for this interface.

**PURGE\_IN** = VARCHAR2 - Single character that identifies how records will be removed from the interface table.

**Y** = records will be deleted from the interface table as they are processed.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_CUSTOMER\_INTERFACE runs again.

**OPTION\_IN** = VARCHAR1 - Not used.

**PRE\_IN** = VARCHAR2 - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN** = VARCHAR2(200) - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Customer Interface Rule

This business rule only applies to pre-existing inbound records.

**RULE ID** = CUSTOMER INTERFACE

**RULE TYPE** = INTERFACE

**RULE STYLE** = LIST

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound customer record that already exist in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_CUSTOMER column to be updated.

**UPDATE** = YES or NO. If this value is set to YES then the corresponding field will be updated.

## Customer Address Interface Rule

**RULE ID** = CUSTOMER ADDRESS INTERFACE

**RULE TYPE** = INTERFACE

**RULE STYLE** = LIST

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound customer address records that already exists in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_CUSTOMER\_ADDRESS column to be updated.

**UPDATE** = YES or NO. If this value is set to YES then the corresponding field will be updated.

## Customer SA Interface Rule

**RULE ID** = CUSTOMER SA INTERFACE

**RULE TYPE** = INTERFACE

**RULE STYLE** = LIST

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound customer address records that already exists in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_CUSTOMER\_ADDRESS\_SA column to be updated.

**UPDATE** = YES or NO. If this value is set to YES then the corresponding field will be updated.

## Sequence Numbers

SA\_CUSTOMER

## Interface Table Layout

### WAIF\_CUSTOMER

(same fields as SA\_CUSTOMER & SA\_CUSTOMER\_ADDRESS)

Field Name	Data Type
The following fields are used for both Customer (Record type = H) and Customer Address (Record type = L), and CCB SA (Record Type = A)	
PLANT*	VARCHAR2(3)
CUSTOMER_ID*	VARCHAR2(20)
LAST_UPDATE_USER*	VARCHAR2(30)
LAST_UPDATE_DATE*	DATE
CREATED_BY*	VARCHAR2(30)
CREATED_DATE*	DATE
JOB_SEQ_NO	NUMBER
SENT_TO_INTERFACE_IND	CHAR(1)
RECORD_TYPE*	CHAR(1)
The following fields are used only for Customer records (Record type = H)	
CUSTOMER_STATUS*	VARCHAR2(20)
COMPANY	VARCHAR2(60)

Field Name	Data Type
CUSTOMER_LAST_NAME*	VARCHAR2(30)
CUSTOMER_FIRST_NAME*	VARCHAR2(30)
CUSTOMER_MIDDLE_NAME	VARCHAR2(30)
ATTRIBUTE1	USER-DEFINED
ATTRIBUTE2	USER-DEFINED
ATTRIBUTE3	USER-DEFINED
ATTRIBUTE4	USER-DEFINED
ATTRIBUTE5	USER-DEFINED
ATTRIBUTE6	USER-DEFINED
ATTRIBUTE7	USER-DEFINED
ATTRIBUTE8	USER-DEFINED
ATTRIBUTE9	USER-DEFINED
ATTRIBUTE10	USER-DEFINED
The following fields are used only for Customer Address records (record type = L)	
CUSTOMER_ADDRESS_ID*	NUMBER
CONTACT_INFO_IND*	CHAR(1)
STREET_NUMBER*	NUMBER(10)
NUMBER_SUFFIX	VARCHAR2(5)
STREET_NAME*	VARCHAR2(40)
STREET_DIRECTION	VARCHAR2(30)
SUITE	VARCHAR2(10)
POST_OFFICE_BOX	VARCHAR2(10)
CITY	VARCHAR2(40)
STATE_PROVINCE	VARCHAR2(4)
POSTAL_CODE	VARCHAR2(15)
COUNTRY_CODE	VARCHAR2(2)
PROPERTY_TYPE	VARCHAR2(20)
PHONE_NO_HOME	VARCHAR2(30)
PHONE_NO_WORK	VARCHAR2(30)
PHONE_NO_WORK_EXT	VARCHAR2(5)
FAX_NO	VARCHAR2(30)
EMAIL_ADDRESS	VARCHAR2(100)
WEBSITE_ADDRESS	VARCHAR2(100)

Field Name	Data Type
TAX_ID	VARCHAR2(20)
LINE_ATTRIBUTE1	USER-DEFINED
LINE_ATTRIBUTE2	USER-DEFINED
LINE_ATTRIBUTE3	USER-DEFINED
LINE_ATTRIBUTE4	USER-DEFINED
LINE_ATTRIBUTE5	USER-DEFINED
LINE_ATTRIBUTE6	USER-DEFINED
LINE_ATTRIBUTE7	USER-DEFINED
LINE_ATTRIBUTE8	USER-DEFINED
LINE_ATTRIBUTE9	USER-DEFINED
LINE_ATTRIBUTE10	USER-DEFINED
NUMBER_PREFIX	VARCHAR2(3)
STREET_NUMBER_CHAR	VARCHAR2(10)
CCB_ACCT_ADDRESS_SOURCE	VARCHAR2(60)
CCB_ACCT_SETUP_DATE	DATE
CCB_ACCT_BILL_CYCLE	VARCHAR2(60)
CCB_ACCT_MAIN_PERSON_ID	VARCHAR2(10)
CCB_ACCT_PREMISE_ID	VARCHAR2(10)
CROSS_STREET	VARCHAR2(40)
CCB_SA_ID	VARCHAR2(10)
CCB_SA_STATUS	VARCHAR2(70)
CCB_SA_TYPE_DESC	VARCHAR2(60)
CCB_SA_START_DATE	DATE
PARSE_ADDRESS	VARCHAR(1)
CROSS_STREET	VARCHAR2(40)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER
DIRECTION	CHAR(1)

\* required fields

---

# Chapter 11

---

## Direct Charges

The Direct Charge interface provides a means of importing P-Card and other purchase transactions that occur outside Oracle Utilities Work and Asset Management into the application in order for the work order to properly reflect all relevant charges against an asset. Direct charges can also be interfaced out to an external system, after they are posted.

This interface requires:

**Interface Table:** WAIF\_DIRECT\_CHARGE

**Stored Procedure:** WIFP\_DIRECT\_CHARGE\_INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

WIFP\_DIRECT\_CHARGE\_INTERFACE(JOB\_IN,  
PLANT\_IN,  
DIRECTION\_IN,  
PURGE\_IN,  
OPTION\_IN,  
PRE\_IN,  
POST\_IN)

### Keywords and Parameters

**JOB\_IN** = will be assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN** = identifies the PLANT\_CODE.

**DIRECTION\_IN** = **VARCHAR2** - Single character that identifies which record type(s) to process.

**I** = Indicates that IN records in the interface table are to be processed.

All records in the interface table with **DIRECTION** = 'I' will be moved to the appropriate Oracle Utilities Work and Asset Management table as is.

**O** = Indicates that OUT records are to be written to the interface table.

Only "posted" adjustments are sent.

**PURGE\_IN** = identifies how records will be removed from the interface table.

**Y = If DIRECTION\_IN = I**, records will be deleted from the interface table as they are processed.

**If DIRECTION\_IN = O**, records will be deleted from the interface table before writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_DIRECT\_CHARGE\_INTERFACE runs again.

**OPTION\_IN** = For inbound only - If set to “Y”, the interface will ignore Item (Line) record errors and move onto next Item; otherwise it will ignore the whole header/detail set. To clarify further, when the option is set to “Y”, and there's an error on one line of an inbound direct charge, just that line will be skipped but the rest of the direct charge will be created. If the option is set to “N” and there's an error on one line of the charge, the whole charge will be skipped.

**PRE\_IN** = (Optional) Identifies the custom stored procedure to call before the interface code executes.

**POST\_IN** = (Optional) Identifies the custom stored procedure to call after the interface code executes.

## Sequence Numbers

SA\_DIRECT\_CHARGES

## Interface Table Layout

### WAIF\_DIRECT\_CHARGE

(same fields as SA\_DIRECT\_CHARGES and SA\_DIRECT\_CHARGES\_ITEMS)

Field Name	Data Type
The following columns are used by both 'Header and 'Line records.	
PLANT*	VARCHAR2(3)
ODC_NO*	VARCHAR2(10)
LAST_UPDATE_USER*	VARCHAR2(30)
LAST_UPDATE_DATE*	DATE
CREATED_BY*	VARCHAR2(30)
CREATED_DATE*	DATE
WEB_SERVICE_INBOUND_ID	NUMBER
JOB_SEQ_NO	NUMBER
RECORD_TYPE*	CHAR(1) “H” = Header, “L” = Line
DIRECTION*	CHAR(1) “I” = Inbound, “O” = Outbound
SENT_TO_INTERFACE_IND	CHAR(1)

Field Name	Data Type
The following columns are used by 'Header records.	
EMPLOYEE_NO*	VARCHAR2(6)
ODC_DATE*	DATE
ODC_DESC	VARCHAR2(2000)
EMPLOYEE_NAME	VARCHAR2(45)
ODC_STATUS*	VARCHAR2(20)
NEXT_APPROVER	VARCHAR2(6)
ATTRIBUTE1	User-defined
ATTRIBUTE2	User-defined
ATTRIBUTE3	User-defined
ATTRIBUTE4	User-defined
ATTRIBUTE5	User-defined
ATTRIBUTE6	User-defined
ATTRIBUTE7	User-defined
ATTRIBUTE8	User-defined
ATTRIBUTE9	User-defined
ATTRIBUTE10	User-defined
ODC_CATEGORY	VARCHAR2(10)
ODC_OWNER	VARCHAR2(30)
ROUTING_LIST_ID	VARCHAR2(10)
The following columns are used by 'Line records.	
ODC_ITEM_NO*	VARCHAR2(3)
ODC_TYPE*	VARCHAR2(10)
ODC_UNITS*	VARCHAR2(10)
STANDARD_PRICE*	NUMBER(15,4)
QUANTITY*	NUMBER(12,4)
TOTAL_AMOUNT*	NUMBER(15,4)
CHARGE_TYPE*	CHAR(1)
CHARGE_NO*	VARCHAR2(15)
WORK_ORDER_TASK_NO	VARCHAR2(2)
ASSET_RECORD_TYPE	VARCHAR2(1)
COMMENTS	VARCHAR2(2000)
ACCOUNT_NO*	VARCHAR2(75)



Field Name	Data Type
EXPENSE_CODE*	VARCHAR2(10)
RENTED_ASSET_RECORD_TYPE	VARCHAR2(1)
RENTED_ASSET_ID	VARCHAR2(15)
BEGINNING_READING	NUMBER(10,2)
ENDING_READING	NUMBER(10,2)
USAGE_DATE	DATE
ODC_TOTAL_AMOUNT	NUMBER
VENDOR_CODE	VARCHAR2(30)
REFERENCE_NO	VARCHAR2(30)
RECORD_TYPE	CHAR(1)
DIRECTION	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER
WEB_SERVICE_INBOUND_ID	NUMBER

\* required fields

---

# Chapter 12

## Employee

The Employee interface provides a means to import and export employee information from Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table:** WAIF\_EMPLOYEE

**Stored Procedure:** WIFP\_EMPLOYEE\_INTERFACE

**Business Rule:** EMPLOYEE INTERFACE -controls the stored procedure.

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_EMPLOYEE_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

Only processed records are purged. The job number is used to determine which records to inactivate, if the Inactivate option is chosen.

During outbound processing, if you select the Update option, only those records updated after the last job run are interfaced. Unprocessed records are not purged and you may have more than one record for the employee in record the table. If you do not select the Update option, Purge "Y" will remove all existing outbound records and replace them with new ones. Unless you are sending only updates, you should always set the purge flag to "Y".

### Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to the Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the interface table to the Oracle Utilities Work and Asset Management application.

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y** = If **DIRECTION = I**, records will be deleted from the interface table as they are processed.

**If DIRECTION = O**, records will be deleted from the interface table before writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_EMPLOYEE\_INTERFACE runs again.

**OPTION\_IN = VARCHAR2** - (Optional) Character string that identifies the special processing options.

**If DIRECTION = I and...**

- **OPTIONS = I** – The interface procedure will ‘Inactivate’ all pre-existing employee records in the Oracle Utilities Work and Asset Management application that were not created, or updated, by the current run of the interface procedure. This would be assuming that the interface table contains all the active employees and that all other employees are no longer active.

**If DIRECTION = O and...**

- **OPTIONS = A** -The interface procedure will only write ‘Active’ employee records to the interface table.
- **OPTIONS = U** - The interface procedure will only write employee records that have been updated since the last run of the interface procedure to the interface table.
- **OPTIONS = AU** - Both of the above options apply.

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Business Rule Format

This business rule only applies to pre-existing inbound records.

**RULE ID** = EMPLOYEE INTERFACE

**RULE TYPE** = INTERFACE

**RULE STYLE** = LIST

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound employee record that already exists in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_EMPLOYEE column to be updated.

**UPDATE** = YES or NO. If this value is set to YES then the corresponding field will be updated.

## Sequence Numbers

SA\_EMPLOYEE

## Interface Table Layout

### WAIF\_EMPLOYEE

(Same fields as SA\_EMPLOYEE)

Field Name	Data Type
PLANT*	VARCHAR2(3)
EMPLOYEE_NO*	VARCHAR2(6)
EMPLOYEE_STATUS*	VARCHAR2(20)
LAST_UPDATE_DATE*	DATE
NAME_LAST	VARCHAR2(20)
NAME_FIRST	VARCHAR2(20)
NAME_MIDDLE	VARCHAR2(20)
ADDRESS	VARCHAR2(40)
CITY	VARCHAR2(40)
STATE_PROVINCE	VARCHAR2(4)
POSTAL_CODE	VARCHAR2(15)
PHONE_NO_HOME	VARCHAR2(30)
PHONE_NO_WORK	VARCHAR2(30)
DEPARTMENT	VARCHAR2(10)
AREA	VARCHAR2(10)
CREW	VARCHAR2(5)
CRAFT	VARCHAR2(5)
LOCATION	VARCHAR2(10)
USERNAME	VARCHAR2(30)
HIRED_DATE	DATE
JOB_NO	VARCHAR2(6)
TERMINATION_DATE	DATE
RESOURCE_TYPE	VARCHAR2(6)
SUPERVISOR_TITLE	VARCHAR2(6)

Field Name	Data Type
JOB_TITLE	VARCHAR2(6)
TITLE1	VARCHAR2(6)
TITLE2	VARCHAR2(6)
TITLE3	VARCHAR2(6)
TITLE4	VARCHAR2(6)
TITLE5	VARCHAR2(6)
TITLE6	VARCHAR2(6)
TITLE7	VARCHAR2(6)
TITLE8	VARCHAR2(6)
TITLE9	VARCHAR2(6)
ADMINISTRATIVE_DEPARTMENT	VARCHAR2(10)
CLASSIFICATION	VARCHAR2(10)
SOCIAL_SECURITY_NO	VARCHAR2(9)
PAYROLL_GROUP	VARCHAR2(20)
OVERTIME_CREW	VARCHAR2(5)
OVERTIME_ZONE	VARCHAR2(5)
SENIORITY	NUMBER(3)
YTD_OT_HOURS	NUMBER(10,2)
LAST_YEAR_OT_HOURS	NUMBER(10,2)
CREW_OVERTIME_IND	CHAR(1)
ZONE_OVERTIME_IND	CHAR(1)
LAST_OT_ADJUSTMENT_DATE	DATE
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_DATE*	DATE
CREATED_BY*	VARCHAR2(30)
ATTRIBUTE1	USER-DEFINED
ATTRIBUTE2	USER-DEFINED
ATTRIBUTE3	USER-DEFINED
ATTRIBUTE4	USER-DEFINED
ATTRIBUTE5	USER-DEFINED
ATTRIBUTE6	USER-DEFINED
ATTRIBUTE7	USER-DEFINED
ATTRIBUTE8	USER-DEFINED

Field Name	Data Type
ATTRIBUTE9	USER-DEFINED
ATTRIBUTE10	USER-DEFINED
SENT_TO_INTERFACE_IND	VARCHAR2(1)
JOB_SEQ_NO	NUMBER
AUTO_CREATE_TIMESHEET_IND*	CHAR(1)
WORK_START_TIME	VARCHAR2(5)
WORK_STOP_TIME	VARCHAR2(5)
ROUTING_LIST_ID	VARCHAR2(10)
START_DATE	DATE
DEFAULT_NEXT_APPROVER	VARCHAR2(6)
SUNDAY_IND	VARCHAR2(1)
MONDAY_IND	VARCHAR2(1)
TUESDAY_IND	VARCHAR2(1)
WEDNESDAY_IND	VARCHAR2(1)
THURSDAY_IND	VARCHAR2(1)
FRIDAY_IND	VARCHAR2(1)
SATURDAY_IND	VARCHAR2(1)
DIRECTION*	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

\* required fields

---

# Chapter 13

---

## Employee Wage Rate

The Employee Wage Rate interface provides a means to import wage rate data for use in Oracle Utilities Work & Asset Management.

This interface requires:

**Interface Table:** WAIF\_EMPLOYEE\_WAGE\_RATE

**Stored Procedure:** WIFP\_EMP\_WAGERATE\_INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_EMP_WAGERATE_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

If a record already exists for the employee number and effective date, the record is updated with the new wage rate, transaction date and last updated by information only. No user defined fields are updated.

### Keywords and Parameters

**JOB\_IN**= Integer – The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN** = VARCHAR2 – Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not implemented for this interface.

**PURGE\_IN** = VARCHAR2 – Single character that identifies how records will be removed from the interface table.

**Y** = records will be deleted from the interface table as they are processed.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_EMPLOYEE\_INTERFACE runs again.

**PRE\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Interface Table Layout

### WAIF\_EMPLOYEE\_WAGE\_RATE

(same fields as SA\_EMPLOYEE\_WAGE\_RATE)

Field Name	Data Type
PLANT *	VARCHAR2(3)
EMPLOYEE_NO*	VARCHAR2(6)
TRANSACTION_DATE*	DATE
EFFECTIVE_DATE*	DATE
WAGE_RATE	NUMBER(8,4)
WAGE_RATE_COMMENTS	VARCHAR2(2000)
LAST_UPDATE_DATE*	DATE
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_DATE*	DATE
CREATED_BY*	VARCHAR2(30)
ATTRIBUTE1	USER-DEFINED
ATTRIBUTE2	USER-DEFINED
ATTRIBUTE3	USER-DEFINED
ATTRIBUTE4	USER-DEFINED
ATTRIBUTE5	USER-DEFINED
ATTRIBUTE6	USER-DEFINED
ATTRIBUTE7	USER-DEFINED
ATTRIBUTE8	USER-DEFINED
ATTRIBUTE9	USER-DEFINED
ATTRIBUTE10	USER-DEFINED
SENT_TO_INTERFACE_IND	VARCHAR2(1)



JOB_SEQ_NO	NUMBER
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

\* required fields

---

# Chapter 14

## Fuel

The Fuel interface provides a means for fuel transactions occurring outside of Oracle Utilities Work & Asset Management to be properly reflected against the asset.

This interface requires:

**Interface Table:** WAIF\_FUEL\_SYSTEM\_DATA

**Stored Procedure:** WIFP\_FUEL\_INTERFACE

**Package Body:** SIFP\_FUEL\_INTERFACE

If ASSET\_RECORD\_TYPE = V, consumable types beginning with G (Gas) are logged in Vehicle Cost under Fuel. Consumable types beginning with O (Oil) are logged under Lubricants/Oils. The interface inserts new runtime values with Reason = INTERFACE

To determine if a reading is a rollover reading, the interface compares current INTERFACE reading with the previous INTERFACE reading. If there is no previous reading with an INTERFACE reason code, the reading is assumed not to be a rollover reading.

PMs will cycle as appropriate based on the updated runtime values.

This interface inserts into both Asset Consumables and Asset Cost.

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Job Manager according to the client's requirements.

```
WIFP_FUEL_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPRION_IN,  
PRE_IN,  
POST_IN  
)
```

Only processed records are purged.

## Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management Job Manager.

**PLANT\_IN** = VARCHAR2 - identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not implemented for this interface.

**PURGE\_IN** = VARCHAR2 - identifies how records will be removed from the interface table.

**Y** = Records that have already been processed will be deleted from the interface table before processing begins.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, until some other process deletes the records or until this interface is run with PURGE\_IN set to Y.

**OPTION\_IN** = VARCHAR2 - This parameter is not used in this interface.

**PRE\_IN** = VARCHAR2(200) (Optional) - Identifies the custom stored procedure to call before the interface code executes.

**POST\_IN** = VARCHAR2(200) (Optional) - Identifies the custom stored procedure to call after the interface code executes.

## Business Rule Format

There must be exactly one row in the EXPENSE CODES Business Rule where the Category is CONSUMABLES.

## Interface Table Layout

### WAIF\_FUEL\_SYSTEM\_DATA

		Description
PLANT*	VARCHAR2(3)	Plant code for the transaction.
ASSET_RECORD_TYPE*	VARCHAR2(1)	Asset record type of the asset that used the fuel.
ASSET_ID*	VARCHAR2(15)	Asset ID to that used the fuel
TRANSACTION_DATE*	DATE	Date the transaction occurred.
TRANSACTION_QUANTITY*	NUMBER	Number of units dispensed.

		<b>Description</b>
UNIT_OF_ISSUE*	VARCHAR2(4)	Unit type of issued quantity (for example, GL = Gallons), from code table 23.
CONSUMABLE_CATEGORY*	CHAR(1)	Category of the consumable for this transaction, from code table 36. For example, F = fuel.
CONSUMABLE_TYPE*	VARCHAR2(10)	Type of consumable, from code table 31. For example UL for unleaded.
CONSUMABLE_SOURCE	VARCHAR2(8)	Source of consumables, from code table 32.
READING	NUMBER	The current meter reading for the asset.
READING_UNITS	VARCHAR2(15)	The units of the meter reading (MILES or HOURS).
TRANSACTION_USER*	VARCHAR2(30)	Who to record as the created user, on the Consumables record.
ATTRIBUTE1	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
ATTRIBUTE2	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
ATTRIBUTE3	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
ATTRIBUTE4	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
ATTRIBUTE5	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
ATTRIBUTE6	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
ATTRIBUTE7	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
ATTRIBUTE8	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
ATTRIBUTE9	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
ATTRIBUTE10	VARCHAR2(1)	Consumables module user-defined field, based on system configuration.
SENT_TO_INTERFACE_IND	CHAR	Has the record been processed yet? (Y/N)
TRANSACTION_NUMBER	NUMBER	Not used. For auditing purposes only.
TOTAL_COST	NUMBER(15,2)	The total cost of the fuel purchase. If null, cost will be calculated based on the consumables price.

		Description
METER_NUMBER_OF_DIALS	NUMBER(2)	The number of dials on the asset's meter; used to calculate rolled-over meter reading. A 5-dial meter rolls over to 100,000.
SOURCE_SYSTEM	VARCHAR2(3)	
INT_BATCH_NUMBER	NUMBER	
JOB_STATUS	VARCHAR2(15)	
JOB_MESSAGE	VARCHAR2(4000)	
INT_INSTANCE_ID	NUMBER	
DIRECTION	CHAR(1)	

\* required fields

---

# Chapter 15

## General Ledger

The General Ledger interface allows Oracle Utilities Work & Asset Management transactions to be exported to an external accounting system. The interface can also create the correct offset transactions to satisfy an accounting system's two-sided general ledger.

This interface requires:

**Interface Table:** WAIF\_GL\_TRANSACTION

**Stored Procedure:** WIFP\_GL\_TRANS\_INTERFACE

**Business Rule:** GL TRANSACTION INTERFACE

The transactions being interfaced can come from the Storeroom Log or the Account Log, depending on the transaction type and the business rule.

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Transaction Types](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_GL_TRANS_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

Since general ledger data is often time sensitive, this interface will only commit the data as a batch. If any record fails to process, the whole job fails and the problem must be resolved before running the job again.

The OPTION\_IN parameter allows you to target different transaction outputs for different post-processing. If you do not supply a value for this parameter, the default value GL TRANSACTION INTERFACE is used.

## Keywords and Parameters

**JOB\_IN** = Integer – The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN** = VARCHAR2 – Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application. This option is not implemented for this interface.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table.

**PURGE\_IN** = VARCHAR2 – Single character that identifies how records will be removed from the interface table.

**Y** = Records will be deleted from the interface table **before** writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_GL\_TRANS\_INTERFACE runs again.

**OPTION\_IN** – VARCHAR2 – Character string that identifies the user defined business rule to use for control. See rule format below for details. The default Rule ID is GL TRANSACTION INTERFACE.

**PRE\_IN** = VARCHAR2 – Character string that identifies the custom stored procedure to call before OUT records are written to the interface table.

**POST\_IN** = VARCHAR2 – Character string that identifies the custom stored procedure to call after OUT records are written to the interface table.

## Business Rule Format

**RULE ID** = User Defined. The default is GL TRANSACTION INTERFACE.

**RULE TYPE** = INTERFACE

**RULE STYLE** = LIST

**RULE LIMIT** = ON

**RULE DESCRIPTION** = Identifies the GL transaction types to process.

**TRANSACTION TYPE** = Oracle Utilities Work and Asset Management Transaction Type Codes that are to be processed (See list Below)

**INCLUDE** = Turns on or off the processing of this transaction type (YES, NO).

**LOG TABLE** = Indicates which log table to pull the transaction from (ACCOUNT, STOREROOM). Some transactions can only come from one location. Some can come from either. See online help for a detailed list.

**OFFSET** = Indicates if a transaction is to have a corresponding offset transaction in the interface table (YES, NO).

**OFFSET WITH** = Indicates where the offset account will come from (CREDIT ACCOUNT, RULE).

**OFFSET ACCOUNT** = Account Number to use for offsets on this transaction type if **OFFSET WITH** = **RULE**. Using **CREDIT ACCOUNT** here allows you to use the same expense codes but different credit accounts.

**OFFSET EXPENSE CODE** = Expense Code to use for offsets on this transaction type If **OFFSET WITH** = **RULE**.

## Transaction Types

The following shows some of the transaction types available in the Oracle Utilities Work and Asset Management application. See online help for a complete listing of transaction types.

Type	Description
AC	Asset Cost Change
AG	Asset Gain/Loss
AF	Accrued Invoice Federal Tax
AI	Stock Issue to an Account
AJ	Cost Adjustment
AN	Asset Cost for New Asset
AR	Stock Return from an Account
AS	Accrued Invoice State Tax
AW	Asset Write Off
CI	Checkout Request Stock Issue
DA	Accumulated Depreciation
DC	Direct Charges
DP	Asset Depreciation
IN	Invoice Transaction
LP	Premium Labor
LR	Regular Labor
M1	Labor Markup 1
M2	Labor Markup 2
M3	Labor Markup 3
PB	Premium Labor Burden
RB	Regular Labor Burden
RI	Service Request Stock Issue
RR	Service Request Stock Return
WR	Work Order Stock Return
WU	Work Order Unused Stock Return



## Interface Table Layout

### WAIF\_GL\_TRANSACTION

(same fields as SA\_ACCOUNT\_LOG and SA\_STOREROOM\_LOG)

Field Name	Data Type
The following apply to both Account Log and Storeroom Log records:	
GL_TRANS_RUN_DATE	DATE – the date the data was extracted
SOURCE_LOG	VARCHAR2(10) – ACCOUNT or STOREROOM
SOURCE_ROWID	This is the ROWID of the source record, in either the SA_ACCOUNT_LOG or SA_STOREROOM_LOG table
RULE_ID	Business rule ID used to generate the data.
OFFSET_IND	CHAR(1) – N = the row contains the original transaction. Y = the row contains the offset transaction.
SENT_TO_INTERFACE_IND	CHAR(1)
JOB_SEQ_NO	NUMBER
PLANT	VARCHAR2(3)
STOREROOM	VARCHAR2(3)
STOCK_CODE	VARCHAR2(15)
ACCOUNT_NO	VARCHAR2(75)
EXPENSE_CODE	VARCHAR2(10)
TRANSACTION_TYPE	VARCHAR2(2)
TRANSACTION_DATE	DATE
TRANSACTION_QUANTITY	NUMBER(15,5)
WORK_ORDER_NO	VARCHAR2(7)
WORK_ORDER_TASK_NO	VARCHAR2(2)
PO_NO	VARCHAR2(10)
PO_ITEM	VARCHAR2(10)
ASSET_ID	VARCHAR2(15)
ASSET_RECORD_TYPE	VARCHAR2(1)
SENT_TO_GL	CHAR(1)
INVOICE_NO	VARCHAR2(20)
TRANSACTION_ID	NUMBER(10)
VENDOR_CODE	VARCHAR2(30)

Field Name	Data Type
PERIOD_YEAR	VARCHAR2(4)
PERIOD_MONTH	VARCHAR2(2)
TRANSACTION_AMOUNT	NUMBER(15,2)
The following apply only to Account Log records.	
SEQUENCE_NUMBER	NUMBER
POST_DATE	DATE
WORK_ORDER_CHARGE_IND	CHAR(1)
INVOICE_CHARGE_IND	CHAR(1)
EMPLOYEE_NO	VARCHAR2(6)
TRANSACTION_HOURS	NUMBER(10,2)
GL_BATCH	VARCHAR2(10)
GL_BATCH_DATE	DATE
PROJECT_ID	VARCHAR2(10)
SUBPROJECT_ID	NUMBER
ADJUSTMENT_ID	VARCHAR2(20)
ADJUSTMENT_SEQUENCE_NO	VARCHAR2(5)
SERVICE_CONTRACT_NO	VARCHAR2(10)
CONTRACT_REVISION_NO	NUMBER(5,0)
INVOICE_ITEM	VARCHAR2(3)
SERVICE_REQUEST_NO	VARCHAR2(7)
BILLING_MARKUP_COMP	VARCHAR2(1)
ODC_NO	VARCHAR2(10)
ODC_ITEM_NO	VARCHAR2(3)
COMMITTED_AMOUNT	NUMBER(15,2)
ACCT_COMMITTED_AMOUNT	NUMBER(15,2)
EXP_COMMITTED_AMOUNT	NUMBER(15,2)
ACCT_BUDGET_AMOUNT	NUMBER(15,2)
EXP_BUDGET_AMOUNT	NUMBER(15,2)
ACCOUNT_OVERRUN_IND	CHAR(1)
EXPENSE_OVERRUN_IND	CHAR(1)
REG_ACCT_DIST_COMPLETE_IND	VARCHAR2(1)
INTERFACE_DOCUMENT_TYPE	VARCHAR2(10)
INTERFACE_DOCUMENT_ID	VARCHAR2(20)

Field Name	Data Type
INTERFACE_DOCUMENT_DESC	VARCHAR2(2000)
ASSET_CHANGE_EFFECTIVE_DATE	DATE
ASSET_CHANGE_REQUEST_NO	VARCHAR2(7)
ASSET_CHANGE_REQUEST_SEQ_NO	NUMBER
WRITTEN_DOWN_VALUE	NUMBER(15,2)
ACCUMULATED_DEPRECIATION	NUMBER(15,2)
ASSET_CHANGE_TRANSACTION_TYPE	VARCHAR2(20)
SENT_TO_INTERFACE2_IND	VARCHAR2(1)
SENT_TO_INTERFACE3_IND	VARCHAR2(1)
TIMESHEET_SEQ_NO	NUMBER
EXT_ACTUAL_AMOUNT	NUMBER(15,2)
EXT_ACTUAL_QUANTITY	NUMBER(15,2)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER
DIRECTION_IND	CHAR(1)
The following apply only to Storeroom Log records.	
AVERAGE_UNIT_PRICE_BEFORE	NUMBER(15,4)
MARKUP_RATE	NUMBER
NET_ADJUSTMENT_AMOUNT	NUMBER(15,2)
ISSUE_TICKET_NO	NUMBER
COMPONENT_ID	VARCHAR2(15)
ADJUSTMENT_REASON	VARCHAR2(20)
PO_PREFIX	VARCHAR2(5)
RECEIVING_EMPLOYEE	VARCHAR2(30)
ISSUING_EMPLOYEE	VARCHAR2(30)
TOOL_ID	VARCHAR2(6)
TOOL_LOCATION	VARCHAR2(10)
TOOL_ISSUE_DATE	DATE
LOT_ID	VARCHAR2(20)

Field Name	Data Type
RETURN_REQUEST_NO	NUMBER
RETURN_REQUEST_ITEM	VARCHAR2(3)
BIN	VARCHAR2(20)
NEW_INVENTORY_QUANTITY	NUMBER(15,5)
COMMENTS	VARCHAR2(2000)
DELIVERY_ID	VARCHAR2(7)
STOCK_OUT_IND	VARCHAR2(1)
AVERAGE_UNIT_PRICE_AFTER	NUMBER(15,4)
TOTAL_VALUE_BEFORE	NUMBER(15,2)
TOTAL_VALUE_AFTER	NUMBER(15,2)
INVENTORY_QUANTITY_BEFORE	NUMBER(15,5)
INVENTORY_QUANTITY_AFTER	NUMBER(15,5)
PHYSICAL_INVENTORY_NO	NUMBER
REFERENCE_NO	VARCHAR2(30)
TRANSFER_NO	NUMBER

# GIS Interfaces

Regardless of which GIS viewer you use, implementation of the Oracle Utilities Work and Asset Management interface to GIS requires that you map GIS features to the Oracle Utilities Work and Asset Management sa\_asset table then complete additional configuration steps. Please refer to the [GIS Implementation](#) chapter of the GIS Overview guide for details.









---

# Chapter 17

## Inventory Log

The Inventory Log interface provides the ability to import inventory log (issues & returns) data from other systems into Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table:** WAIF\_INVENTORY\_LOG

**Stored Procedure:** WIFP\_INVENTORY\_LOG\_INTERFACE

**Business Rule:** INTERFACE PARAMETERS

[Batch Job Procedure](#)

[Business Rule Format](#)

[Keywords and Parameters](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_INVENTORY_LOG_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

Set the transaction status to "1" if you want the system to post the charges for the transaction account, work order, asset, etc. Otherwise, set the transaction status to "2". In either case, storeroom inventory counts will not be affected.

### Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not yet implemented for this interface.

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y = Records** will be deleted from the interface table as they are processed.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_INVENTORY\_LOG\_INTERFACE runs again.

**OPTION\_IN = VARCHAR2** - This parameter is not used by the Inventory Log interface.

**PRE\_IN = VARCHAR2**- (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes. Note below for requirements.

## Business Rule Format

**RULE ID** = INTERFACE PARAMETERS

**RULE TYPE** = INTERFACE

**KEY NAME** = INVENTORY INTERFACE HANDLING

**PARAMETER VALUE** = The default is OLD. The rule key determines account and expense code handling for the WIFP\_INVENTORY\_LOG\_INTERFACE. If the Param Value is set to OLD, the system uses the account from the referenced work order task and the expense code from the storeroom. If the Param Value is set to NEW, the system uses the values from the interface table. When NEW processing is used, the action type must also be set to 'P'. This setting can be accessed in WAIF\_INVENTORY\_LOG.

## Sequence Numbers

SA\_INVENTORY\_LOG

## Interface Table Layout

### WAIF\_INVENTORY\_LOG

(Same fields as SA\_INVENTORY\_LOG)

Field Name	Data Type
PLANT*	VARCHAR2(3)
ISSUE_TICKET_NO*	NUMBER
TRANSACTION_TYPE*	VARCHAR2(2)

Field Name	Data Type
TRANSACTION_STATUS*	CHAR(1) 1=Cost needs to be posted; 2=Already posted
TRANSACTION_DATE*	DATE
REQUEST_NO	NUMBER
ISSUING_EMPLOYEE	VARCHAR2(30)
TRANSACTION_AMOUNT	NUMBER(10,2)
TRANSACTION_QUANTITY	NUMBER(15,5)
STOREROOM	VARCHAR2(3)
STOCK_CODE	VARCHAR2(15)
COMPONENT_ID	VARCHAR2(15)
PRIMARY_BIN	VARCHAR2(20)
INVENTORY_QUANTITY	NUMBER(15,5)
AVERAGE_UNIT_PRICE	NUMBER(15,4)
MARKUP_RATE	NUMBER(5,3)
RECEIVING_EMPLOYEE	VARCHAR2(30)
ASSET_ID	VARCHAR2(15)
ASSET_RECORD_TYPE	VARCHAR2(1)
WORK_ORDER_NO	VARCHAR2(7)
WORK_ORDER_TASK_NO	VARCHAR2(2)
PROJECT_ID	VARCHAR2(10)
ACCOUNT_NO	VARCHAR2(75)
EXPENSE_CODE	VARCHAR2(10)
SUBPROJECT_ID	NUMBER
PO_NO	VARCHAR2(10)
ISSUE_PRICE	NUMBER(15,4)
LOT_ID	VARCHAR2(20)
RETURN_REQUEST_NO	NUMBER
RETURN_REQUEST_ITEM	VARCHAR2(3)
BOM_PROCESSED_IND	VARCHAR2(1)
SERVICE_REQUEST_NO	VARCHAR2(7)
STOCK_OUT_IND	VARCHAR2(1)
COMMENTS	VARCHAR2(2000)
SENT_TO_INTERFACE_IND	VARCHAR2(1)
JOB_SEQ_NO	NUMBER

Field Name	Data Type
WEB_SERVICE_INBOUND_ID	NUMBER
UNIT_OF_ISSUE	VARCHAR2(4)
UNIT_OF_PURCHASE	VARCHAR2(4)
PI_RATIO	NUMBER(22,0)
ACTION_TYPE	CHAR(1)
UPDATE_STOREROOM	CHAR(1)
DIRECTION*	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER
WEB_SERVICE_INBOUND_ID	NUMBER

\* required fields

---

# Chapter 18

## Invoice AP Data

The Invoice AP Data interface receives AP data (check number, AP amount, etc.) from an external AP system and updates the proper invoice in Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table:** WAIF\_INVOICE\_AP\_DATA

**Stored Procedure:** WIFP\_INVOICE\_AP\_DATA\_INTERFACE

All fields in the application table (except the Sent to Interface indicator) are updated with the values from the interface table, even if the value is null. Be sure to populate all values in the export of the pre-processing procedure. The vendor code should be the Order From vendor, which is part of the invoice primary key, and not the Pay To vendor.

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_INVOICE_AP_DATA_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTIONS_IN,  
PRE_IN,  
POST_IN)
```

### Keywords and Parameters

**JOB\_NUMBER** = Integer – The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT** = VARCHAR2 – Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not implemented for this interface.

**PURGE\_IN** = VARCHAR2 – Single character that identifies how records will be removed from the interface table.

**Y** = Records will be deleted from the interface table as they are processed.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_INVOICE\_AP\_DATA\_INTERFACE runs again.

**OPTION\_IN** = Not used in this interface.

**PRE\_IN** = VARCHAR2 – Optional Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Interface Table Layout

### WAIF\_INVOICE\_AP\_DATA

Field Name	Data Type
PLANT (required)	VARCHAR2(3)
INVOICE_NO (required)	VARCHAR2(20)
VENDOR_CODE (required)	VARCHAR2(20)
AP_AMOUNT	NUMBER(15,2)
AP_BATCH_VOUCHER	VARCHAR2(10)
AP_BATCH_NO	VARCHAR2(10)
AP_BATCH_DATE	DATE
MANUAL_CHECK_NO	VARCHAR2(20)
INVOICE_CHECK_CODE	CHAR(1)
SENT_TO_INTERFACE_IND	VARCHAR2(1)
JOB_SEQ_NO	NUMBER
PAYMENT_DATE	DATE
PAID_IN	CHAR(1)
DIRECTION (required)	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

---

# Chapter 19

## Invoice

This interface allows invoices from Oracle Utilities Work and Asset Management to be exported to a client's AP system. The interface will also load and post invoices from the AP system to Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table** - WAIF\_INVOICE

**Stored Procedure** - WIFP\_INVOICE\_INTERFACE (processes the interface table)

(optional) **Business Rule** - INTERFACE PARAMETERS

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_INVOICE_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

The stored procedure processes both IN data, data being put into the Oracle Utilities Work and Asset Management Invoice module, and OUT data, data being sent from the Oracle Utilities Work and Asset Management Invoice module to an external system, such as Accounts Payable.

For inbound invoices, the split amount should be left null on the "A" account records. After inserting the invoice in the application tables, the invoice is automatically approved.

### Keywords and Parameters

**JOB\_IN** = Integer – The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN** = VARCHAR2 – Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 – (Optional) Single character that identifies which record type(s) to process.

**I** = Indicates that IN records in the interface table are to be processed. All records in the interface table with **DIRECTION** = 'I' will be moved to the appropriate Oracle Utilities Work and Asset Management table as is.

**O** = Indicates that OUT records are to be written to the interface table. All invoice records that have status equal to the value from the business rule (default is POSTED) and Send to AP equal to NULL or 'N', will be written to the interface table with **DIRECTION** = 'O'. After writing all records to the interface table, the invoice Sent to AP indicator is set to 'Y'.

**PURGE\_IN** = VARCHAR2 – (Optional) Single character that identifies how records will be removed from the interface table.

**Y** = Records will be deleted from the interface table after they are processed for IN records and before that are processed for OUT records.

**N** = Records will not be deleted from the interface table.

Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_INVOICE\_INTERFACE runs again.

**PRE\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call before IN records in the interface table are processed. See Custom Stored Procedure Note below for requirements.

**POST\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call after IN records in the interface table are processed. See Custom Stored Procedure Note below for requirements.

## Business Rule Format

**RULE ID** = INTERFACE PARAMETERS

**RULE TYPE** = INTERFACE

**KEY NAME** = INVOICE STATUS

**PARAMETER VALUE** = The default is POSTED. If you send unposted invoices they may be changed later but will not be re-sent by the interface.

## Interface Table Layout

### WAIF\_INVOICE

(same fields as SA\_INVOICE, SA\_INVOICE\_ITEM, AND SA\_INVOICE\_ITEM\_ACCOUNT)

Field Name	Data Type	Comments
The following columns are shared by multiple record types (Header, Line and Account).		
PLANT*	VARCHAR2(3)	
VENDOR_CODE*	VARCHAR2(30)	
INVOICE_NO*	VARCHAR2(20)	



Field Name	Data Type	Comments
INVOICE_ITEM*	VARCHAR2(3)	Line and Account records only
FEDERAL_TAX_AMOUNT	NUMBER(15,2)	Header and Line records only
STATE_PROVINCE_TAX_AMOUNT	NUMBER(15,2)	Header and Line records only
DUTY_AMOUNT	NUMBER(15,2)	Header and Line records only
PO_NO	VARCHAR2(10)	Header and Line records only
LAST_UPDATE_DATE*	DATE	Header and Line records only
LAST_UPDATE_USER*	VARCHAR2(30)	Header and Line records only
CREATED_DATE*	DATE	Header and Line records only
CREATED_BY*	VARCHAR2(30)	Header and Line records only
DIRECTION*	CHAR(1)	“I” inbound or “O” outbound
RECORD_TYPE*	CHAR(1)	“H” = header, “L” = line, “A” = account
SENT_TO_INTERFACE_IND	CHAR(1)	
DIRECTION*	CHAR(1)	
SOURCE_SYSTEM	VARCHAR2(3)	
INT_BATCH_NUMBER	NUMBER	
JOB_STATUS	VARCHAR2(15)	
JOB_MESSAGE	VARCHAR2(4000)	
INT_INSTANCE_ID	NUMBER	
The following columns apply only to the header record.		
PO_TYPE	CHAR(1)	
INVOICE_STATUS*	VARCHAR2(20)	
PAY_TO_VENDOR_CODE	VARCHAR2(30)	
CARRIER	VARCHAR2(30)	
ALTERNATE_INVOICE_NO	VARCHAR2(20)	
INVOICE_ENTERED_DATE	DATE	
INVOICE_RECEIVED_DATE	DATE	
INVOICE_STATUS_DATE*	DATE	
INVOICE_VENDOR_DATE	DATE	
INVOICE_DUE_DATE	DATE	
INVOICE_PAYMENT_DAYS	NUMBER(4)	
INVOICE_ITEM_TOTAL_AMOUNT	NUMBER(15,2)	
DISCOUNT_AMOUNT	NUMBER(15,2)	

Field Name	Data Type	Comments
DISCOUNT_RATE	NUMBER(7,4)	
INVOICE_FREIGHT_AMOUNT	NUMBER(15,2)	
INVOICE_EXTRA_AMOUNT	NUMBER(15,2)	
ITEMS_FEDERAL_TAX_AMOUNT	NUMBER(15,2)	
ITEMS_STATE_PROV_TAX_AMOUNT	NUMBER(15,2)	
INVOICE_TOTAL_AMOUNT	NUMBER(15,2)	
CURRENCY_CODE	VARCHAR2(3)	
EXCHANGE_RATE	NUMBER(15,7)	
FOB	VARCHAR2(20)	
BLANKET_CONTRACT_NO	VARCHAR2(10)	
BLANKET_RELEASE_NO	VARCHAR2(4)	
DEFAULT_FEDERAL_TAX_CODE	CHAR(1)	
DEFAULT_STATE_PROV_TAX_CODE	CHAR(1)	
NEXT_APPROVER	VARCHAR2(6)	
AP_BATCH_NO	VARCHAR2(10)	
AP_AMOUNT	NUMBER(15,2)	
AP_BATCH_DATE	DATE	
AP_BATCH_VOUCHER	VARCHAR2(10)	
INVOICE_NOTE	VARCHAR2(200)	
BUYER	VARCHAR2(3)	
TERMS	VARCHAR2(2)	
OVERRIDE_STATE_PROV_TAX	CHAR(1)	
OVERRIDE_FEDERAL_TAX	CHAR(1)	
ITEMS_DUTY_AMOUNT	NUMBER(15,2)	
OVERRIDE_DUTY_IND	CHAR(1)	
PRORATE_DISCOUNT_IND	CHAR(1)	
PRORATE_EXTRA_IND	CHAR(1)	
PRORATE_FREIGHT_IND	CHAR(1)	
PRORATE_FEDERAL_TAX_IND	CHAR(1)	
PRORATE_STATE_PROV_TAX_IND	CHAR(1)	
PRORATE_DUTY_IND	CHAR(1)	
MANUAL_CHECK_NO	VARCHAR2(20)	
INVOICE_CHECK_CODE	CHAR(1)	

Field Name	Data Type	Comments
SENT_TO_AP	VARCHAR2(1)	
FEDERAL_TAX_TO_VENDOR_IND	CHAR(1)	
STATE_PROV_TAX_TO_VENDOR_IND	CHAR(1)	
DUTY_TO_VENDOR_IND	CHAR(1)	
ATTRIBUTE1	User-Defined	
ATTRIBUTE2	User-Defined	
ATTRIBUTE3	User-Defined	
ATTRIBUTE4	User-Defined	
ATTRIBUTE5	User-Defined	
ATTRIBUTE6	User-Defined	
ATTRIBUTE7	User-Defined	
ATTRIBUTE8	User-Defined	
ATTRIBUTE9	User-Defined	
ATTRIBUTE10	User-Defined	
DUTY_TO_VENDOR_AMOUNT	NUMBER(15,2)	
FEDERAL_TO_VENDOR_AMOUNT	NUMBER(15,2)	
STATE_PROV_TO_VENDOR_AMOUNT	NUMBER(15,2)	
ACCRUED_DUTY_TAX_AMOUNT	NUMBER(15,2)	
ACCRUED_FEDERAL_TAX_AMOUNT	NUMBER(15,2)	
ACCRUED_STATE_PROV_TAX_AMOUNT	NUMBER(15,2)	
ACCRUAL_DUTY_ACCOUNT	VARCHAR2(75)	
ACCRUAL_FEDERAL_ACCOUNT	VARCHAR2(75)	
ACCRUAL_STATE_PROV_ACCOUNT	VARCHAR2(75)	
PRORATE_DUTY_TO_NONTAX_IND	CHAR(1)	
PRORATE_FEDERAL_TO_NONTAX_IND	CHAR(1)	
PRORATE_STATE_TO_NONTAX_IND	CHAR(1)	
ACCRUAL_DUTY_EXPENSE	VARCHAR2(10)	
ACCRUAL_FEDERAL_EXPENSE	VARCHAR2(10)	
ACCRUAL_STATE_PROV_EXPENSE	VARCHAR2(10)	
SENT_ALERT_IND	CHAR(1)	

Field Name	Data Type	Comments
AUTO_PAY_IND	CHAR(1)	
PAY_IMMEDIATE_IND	VARCHAR2(1)	
PAYMENT_DATE	DATE	
REPORTING_DATE	DATE	
POST_DATE	DATE	
BATCH_ID	VARCHAR2(10)	
VOUCHER_NO	VARCHAR2(20)	
SEND_TO_AP_IND	VARCHAR2(1)	
PROCESS_LEVEL	VARCHAR2(10)	
SEPARATE_CHECK_IND	VARCHAR2(1)	
MANUAL_INVOICE_TOTAL	NUMBER(15,4)	
PAID_IND	VARCHAR2(1)	
PRORATE_IND	VARCHAR2(1)	
JOB_SEQ_NO	NUMBER	
ROUTING_LIST_ID	VARCHAR2(10)	
MULTIPLE_PO_IND	CHAR(1)	
USE_DISCOUNT_AMOUNT	CHAR(1)	
The following columns apply only to Line records.		
INVOICE_ITEM_TYPE	CHAR(1)	
INVOICE_QUANTITY	NUMBER(15,5)	
INVOICE_UNIT_PRICE	NUMBER(15,4)	
INVOICE_ITEM_AMOUNT	NUMBER(15,2)	
FEDERAL_TAX_CODE	CHAR(1)	
STATE_PROVINCE_TAX_CODE	CHAR(1)	
FEDERAL_TAX_RATE	NUMBER(7,4)	
STATE_PROVINCE_TAX_RATE	NUMBER(7,4)	
STOREROOM	VARCHAR2(3)	
STOCK_CODE	VARCHAR2(15)	
PO_ITEM	VARCHAR2(3)	
WAYBILL_NO	VARCHAR2(15)	
WORK_ORDER_NO	VARCHAR2(7)	
WORK_ORDER_TASK_NO	VARCHAR2(2)	
INVOICE_ITEM_NOTE	VARCHAR2(200)	

Field Name	Data Type	Comments
REVERSAL_INVOICE_NO	VARCHAR2(20)	
ITEM_DESC	VARCHAR2(2000)	
PRORATED_DISCOUNT_AMOUNT	NUMBER(15,2)	
PRORATED_ITEM_AMOUNT	NUMBER(15,2)	
DUTY_CODE	CHAR(1)	
DUTY_RATE	NUMBER(7,4)	
PRORATED_EXTRA_AMOUNT	NUMBER(15,2)	
PRORATED_FREIGHT_AMOUNT	NUMBER(15,2)	
TOTAL_ITEM_CHARGE_AMOUNT	NUMBER(15,2)	
INVOICE_ITEM_TAXABLE_AMOUNT	NUMBER(15,2)	
PRORATED_FREIGHT_EXTRA_AMOUNT	NUMBER(15,2)	
LINE_ATTRIBUTE1	User-Defined	
LINE_ATTRIBUTE2	User-Defined	
LINE_ATTRIBUTE3	User-Defined	
LINE_ATTRIBUTE4	User-Defined	
LINE_ATTRIBUTE5	User-Defined	
LINE_ATTRIBUTE6	User-Defined	
LINE_ATTRIBUTE7	User-Defined	
LINE_ATTRIBUTE8	User-Defined	
LINE_ATTRIBUTE9	User-Defined	
LINE_ATTRIBUTE10	User-Defined	
FEDERAL_TAX_REBATE_RATE	NUMBER(10,4)	
FEDERAL_TAX_REBATE_AMOUNT	NUMBER(15,2)	
DISCREPANCY_IND	CHAR(1)	
The following columns apply only to the Line Account records.		
ACCOUNT_NO*	VARCHAR2(75)	
EXPENSE_CODE*	VARCHAR2(10)	
PERCENT_SPLIT	NUMBER(8,5)	
SPLIT_AMOUNT	NUMBER(15,2)	Leave null on inbound.
UNITS	NUMBER(12,5)	

\* required fields

---

# Chapter 20

---

## Manufacturer Vendor

The Manufacturer Vendor interface provides a means to import and export Master Catalog manufacturer vendor data used in Oracle Utilities Work and Asset Management. The OPTIONS parameter allows control of what type of data is processed.

This interface requires:

**Interface Table:** WAIF\_CATALOG\_MFR\_VENDOR

**Stored Procedure:** WIFP\_MFR\_VENDOR\_INTERFACE

**Business Rule:** MANUFACTURER\_VENDOR\_INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_MFR_VENDOR_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTIONS_IN,  
PRE_IN,  
POST_IN)
```

On outbound, if the Update flag is set, only records updated after the last run of the job are interfaced. Unprocessed records are not purged so more than one record for the same stock code may result. If you do not select the Update option, you should set the purge flag to "Y". The interface removes all previous process or unprocessed output before loading new data.

### Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies inbound or outbound processing.

**I** = Indicates data will be loaded from the interface table to the Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from Oracle Utilities Work and Asset Management application to the interface table.

**PURGE\_IN = VARCHAR2** - Single character identifies how records will be removed from the interface table.

**Y = If DIRECTION = I**, records will be deleted from the interface table as they are processed.

**If DIRECTION = O**, records will be deleted from the interface table before writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_MFR\_VENDOR\_INTERFACE runs again.

**OPTION\_IN = VARCHAR2** – (Optional) Character string identifies the special processing options.

**If DIRECTION = O and...**

- **OPTIONS = U** – The interface procedure will **only** write manufacturer vendor records that have been updated since the last run of the interface procedure to the interface table.

**PRE-PROCESSING = VARCHAR2** – (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST-PROCESSING = VARCHAR2** – (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Business Rule Format

This business rule only applies to pre-existing inbound records.

**RULE ID** = MANUFACTURER VENDOR INTERFACE

**RULE TYPE** = INTERFACE

**RULE STYLE** = LIST

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound manufacturer vendor record that already exists in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_CATALOG\_MFR\_VENDOR column to be updated.

**UPDATE** = YES or NO. If this value is set to YES then the corresponding field will be updated.

## Interface Table Layout

### WAIF\_CATALOG\_MFR\_VENDOR

(same fields as SA\_CATALOG\_MFR\_VENDOR)

Field Name	Data Type
PLANT*	VARCHAR2(3)
STOCK_CODE*	VARCHAR2(15)
SEQUENCE_NO*	VARCHAR2(3)
MANUFACTURER_CODE	VARCHAR2(20)
MANUFACTURER_MODEL_NO	VARCHAR2(30)
MANUFACTURER_PART_NO	VARCHAR2(50)
MANUFACTURER_DRAWING_NO	VARCHAR2(15)
PREVIOUS_MANUFACTURER_CODE	VARCHAR2(20)
VENDOR_CODE	VARCHAR2(30)
VENDOR_MODEL_NO	VARCHAR2(30)
VENDOR_PART_NO	VARCHAR2(50)
PRIMARY_VENDOR_IND	CHAR(1)
FOB	VARCHAR2(20)
SHIPPING_POINT	VARCHAR2(30)
TERMS	VARCHAR2(2)
OEM_IND	CHAR(1)
LAST_INVOICE_PRICE	NUMBER(15,4)
LAST_INVOICE_DATE	DATE
LAST_UPDATE_DATE*	DATE
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_DATE*	DATE
CREATED_BY*	VARCHAR2(30)
ATTRIBUTE1	USER-DEFINED
ATTRIBUTE2	USER-DEFINED
ATTRIBUTE3	USER-DEFINED
ATTRIBUTE4	USER-DEFINED
ATTRIBUTE5	USER-DEFINED
ATTRIBUTE6	USER-DEFINED
ATTRIBUTE7	USER-DEFINED



Field Name	Data Type
ATTRIBUTE8	USER-DEFINED
ATTRIBUTE9	USER-DEFINED
ATTRIBUTE10	USER-DEFINED
COPY_MFR_TO_RFQ_IND	VARCHAR2(1)
COPY_MFR_TO_PO_IND	VARCHAR2(1)
RFQ_VENDOR_IND	VARCHAR2(1)
SENT_TO_INTERFACE_IND	CHAR(1)
DIRECTION*	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

\* required fields

---

# Chapter 21

## Meter Reading

The Meter Reading provide a means by which data from meter readings can be passed to the Oracle Utilities Work and Asset Management for certain assets to initiate routine maintenance activities automatically.

Meter reading data for the identified assets will be passed on a routine basis to populate the Oracle Utilities Work and Asset Management Run-Time (meter) Reading log. While the Run-Time Entry module is available for the manual entry of run-time values for assets, this interface automates the process of data population. The run-time data is available to be viewed from this module by searching on a given asset. Alternatively, you can view asset run-time data as a detail on the Asset record (called Runtime Log). One benefit of this latter option is that the runtime Log is exportable to a file should you wish to conduct trending analysis on the recorded data in Excel. The Runtime Log data is the source used by Oracle Utilities Work and Asset Management in internal batch processing to compare run-time readings against preset values in the PM Master module to trigger the creation of Work Orders. Based on the set up of the Oracle Utilities Work and Asset Management PM Master for the affected assets, Work Orders can be created and placed into the backlog for planning and execution. The interface developed here is both flexible and extendible so that it will remain functional in future Oracle Utilities Work and Asset Management releases.

This interface requires:

**Interface Table:** WAIF\_METER\_READING

**Stored Procedure:** WIFP\_METER\_READING\_INTERFACE

**Package Body:** SIFP\_READINGS\_INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Interface Table Layout](#)

## Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management Job Manager according to the client's requirements.

```
WIFP_METER_READING_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTIONS_IN,  
PRE_IN,  
POST_IN)  
SOURCE_SYSTEM_IN  
INT_BATCH_NUMBER_IN
```

## Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not implemented for this interface.

**PURGE\_IN = VARCHAR2(1)** - Single character that identifies how records will be removed from the interface table.

**Y** = If **DIRECTION = I**, records will be deleted from the interface table as they are processed.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_METER\_READING runs again.

**OPTION\_IN = VARCHAR2** - Not used.

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

**SOURCE\_SYSTEM\_IN = IN VARCHAR2 DEFAULT NULL**

**INT\_BATCH\_NUMBER\_IN = IN NUMBER DEFAULT NULL**

## Interface Table Layout

### WAIF\_METER\_READINGS\_INTERFACE

Field Name	Data Type
PLANT	VARCHAR2 (3)
POINT_ID*	VARCHAR2 (20)
ASSET_RECORD_TYPE*	VARCHAR2 (1)
ASSET_ID*	VARCHAR2 (15)
READING_DATE	DATE
METER_READING	NUMBER
METER_UNITS	VARCHAR2 (15)
SENT_TO_INTERFACE_IND	VARCHAR2(1)
METER_NUMBER_OF_DIALS	NUMBER (2)

---

DIRECTION*	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

\* Either Point ID or Asset Record ID/Type must be specified.

---

# Chapter 22

## Purchase Order

The Purchase Order interface allows Oracle Utilities Work and Asset Management to accept purchase orders from an external purchasing system. The interfaced POs are processed as if created on-line (inventory updated, change orders created).

This interface requires:

**Interface Table:** WAIF\_PURCHASE\_ORDER

**Stored Procedure:** WIFP\_PURCHASE\_ORDER\_INTERFACE

Change Orders allow only changes for limited number of Header and Line Item fields. Please contact Oracle Utilities Work and Asset Management for updated fields. The following fields are not included in the input: status values and dates (they will be set to ISSUED with today's date) and Next Approver Title.

Generally, only changes need to be brought in for Change Orders, since all the other fields (if set to NULL) will be copied from the original Purchase Order.

Oracle Utilities Work and Asset Management does not support change of Account Numbers and Expense Codes for the Change Orders. This logic is supported with this interface. Consequently, if these fields are changed this change will be ignored. If this might create problems this concern should be handled with PRE\_IN procedure.

This interface validates PO Type (Code Table 155) and currency code (Code Table 210).

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_PURCHASE_ORDER_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

## Keywords and Parameters

**JOB\_IN** = Integer – The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN** = VARCHAR2 – Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not yet implemented for this interface.

**PURGE\_IN** = VARCHAR2 – (Optional) Single character that identifies how records will be removed from the interface table.

**Y** = Records will be deleted from the interface table after they are.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_PURCHASE\_ORDER\_INTERFACE.PO\_INTERFACE runs again or at some other regularly scheduled interval.

**PRE\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call before IN records in the interface table are processed. See Custom Stored Procedure Note below for requirements.

**POST\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call after IN records in the interface table are processed. See Custom Stored Procedure Note below for requirements.

**OPTION\_IN** = VARCHAR2 – One or two characters that specify processing options.

**I** = Inactivate (sets status to cancelled) any Line Item that is not in the interface table.

**B** = Blanket contract revisions are maintained in Oracle Utilities Work and Asset Management.

Valid combinations are space, I, B, IB

## Business Rule Format

**RULE ID** = UPDATE PRIMARY VENDOR

**RULE TYPE** = BUSINESS

**KEYNAME** = UPDATE\_PRIME\_VENDOR

## Sequence Numbers

SA\_PURCHASE ORDER

## Interface Table Layout

WAIF\_PURCHASE\_ORDER (same fields as SA\_PURCHASE\_ORDER, SA\_PURCHASE\_ORDER\_ITEM AND SA\_PURCHASE\_ORDER\_ITEM\_ACCOUNT)

Field Name	Data Type	Description
The following fields are available for ALL records.		
PLANT*	VARCHAR2(3)	Plant code
PO_NO*	VARCHAR2(10)	Purchase Order Number
PO_REVISION_NO*	VARCHAR2(3)	Purchase order revision number <sup>1</sup>
RECORD_TYPE*	VARCHAR2(1)	Indicates <u>H</u> header, <u>L</u> ine, <u>D</u> efault Account Or Line <u>A</u> ccount
SENT_TO_INTERFACE_IND	VARCHAR2(1)	If PO Was Successfully Processed
ACTION_TYPE*	VARCHAR2(1)	Indicates PO <u>I</u> nsert, <u>U</u> pdate, or <u>C</u> ancellation
JOB_SEQ_NO	NUMBER	Current Job's Job Sequence Number
The following fields are available for Header and Line records.		
REQUIRED_DATE	DATE	Required date for delivery
FEDERAL_TAX_CODE	CHAR(1)	Federal tax rate - Code Table 159
STATE_PROVINCE_TAX_CODE	CHAR(1)	State or province tax rate - Code Table 160
DUTY_CODE	CHAR(1)	Local tax rate - Code Table 161
CREATED_BY*	VARCHAR2(30)	The user who created this PO
LAST_UPDATE_DATE*	DATE	Date the PO was last updated
LAST_UPDATE_USER*	VARCHAR2(30)	User who last updated this PO
REQUESTOR	VARCHAR2(30)	Requestor of the PO
PROMISE_DATE	DATE	Vendor's promised delivery date
REQUESTOR_DEPARTMENT	VARCHAR2(10)	Requestor's department
DELIVER_TO_DEPARTMENT	VARCHAR2(10)	Department where purchases are to be delivered
QUALITY_IND	VARCHAR2(1)	Quality Indicator
CREATION_DATE*	DATE	Date the record was created.
The following fields are available for Header records.		
PO_TYPE*	CHAR(1)	Type of purchase order – Code Table 155
ISSUE_DATE*	DATE	Date the PO was issued
PO_ISSUER	VARCHAR2(30)	Person who issued the PO
PO_REVISION_DATE	DATE	Date the PO was last revised

Field Name	Data Type	Description
PO_REVISION_ISSUER	VARCHAR2(30)	Person who issued the PO revision
BLANKET_CONTRACT_NO	VARCHAR2(10)	Blanket contract number for this PO
BLANKET_RELEASE_NO	VARCHAR2(4)	Blanket contract release number
REQUESTOR_PHONE_NO	VARCHAR2(30)	Requestor's phone number
REQUEST_DATE	DATE	The requestor's desired delivery date for the PO
PAY_TO_VENDOR_CODE	VARCHAR2(30)	Code no of the vendor to whom payment is made
VENDOR_CODE*	VARCHAR2(30)	Vendor's code for the purchase order <sup>2</sup>
DIVISION	VARCHAR2(40)	Division name of the vendor address <sup>2</sup>
ADDRESS	VARCHAR2(200) )	Street address of the vendor <sup>2</sup>
CITY	VARCHAR2(40)	City of the vendor's address <sup>2</sup>
STATE_PROVINCE	VARCHAR2(4)	State or province of the vendor's address <sup>2</sup>
POSTAL_CODE	VARCHAR2(10)	Zip/Postal code of the vendor's address <sup>2</sup>
VENDOR_CONTACT	VARCHAR2(20)	Contact name for the vendor <sup>2</sup>
VENDOR_PHONE_NO	VARCHAR2(30)	Contact phone number for the vendor <sup>2</sup>
TOTAL_AMOUNT	NUMBER(15,2)	Total amount of PO line items <sup>3</sup>
TERMS	VARCHAR2(2)	The code for the vendor's payment terms
CONFIRMATION_IND	CHAR(1)	Y if this PO has been confirmed by the vendor
CONFIRMATION_CONTACT	VARCHAR2(30)	The vendor's contact name for confirmation
CONFIRMATION_DATE	DATE	The date of PO confirmation by vendor
CARRIER	VARCHAR2(30)	Code for proposed carrier for delivery
FOB	VARCHAR2(20)	Location to which freight charges will be vendor paid
DELIVER_TO	VARCHAR2(20)	Deliver to person or location
SHIP_TO_CODE	VARCHAR2(3)	The storeroom that should receive the PO items
INITIATOR	VARCHAR2(30)	PO initiator's username
CREATION_DATE	DATE	The date the PO was created
BUYER	VARCHAR2(3)	3-place buyer's code used to identify the buyer
EXPEDITE_STATUS	VARCHAR2(10)	Status of the expedite process
EXPEDITE_DATE	DATE	Date the PO was expedited



Field Name	Data Type	Description
DEFAULT_ACCOUNTS	CHAR(1)	Y indicates a default account(s) for line items <sup>4</sup>
BUYER_PHONE_NO	VARCHAR2(30)	Buyer's phone number
CURRENCY_CODE	CHAR(3)	Currency code for the vendor - Code Table 210 <sup>5</sup>
EXCHANGE_RATE*	NUMBER(15,7)	Exchange rate for the currency code <sup>5</sup>
ONLY_SUPPLIER_IND	CHAR(1)	Y indicates that the vendor is only supplier for the item
RFQ_IND	CHAR(1)	Y indicates that a vendor request for quotes is desired
PRICE_CODE	CHAR(1)	Price code
END_USER_CODE	CHAR(1)	End user code specifying the usage - Code Table 26
STANDARD_TEXT	CHAR(1)	Standard text
CREDIT_CARD_PURCHASE_IND	CHAR(1)	Y indicates it is the credit card purchase.
CREDIT_CARD_HOLDER_NAME	VARCHAR2(30)	Credit cardholder's name
BLANKET_CATEGORY	VARCHAR2(10)	Blanket contract category – Code Table 35
BLANKET_REVISION_NO	VARCHAR2(3)	Revision number of the blanket contract
ATTRIBUTE1	USER-DEFINED	This is a user-defined field.
ATTRIBUTE2	USER-DEFINED	This is a user-defined field.
ATTRIBUTE3	USER-DEFINED	This is a user-defined field.
ATTRIBUTE4	USER-DEFINED	This is a user-defined field.
ATTRIBUTE5	USER-DEFINED	This is a user-defined field.
ATTRIBUTE6	USER-DEFINED	This is a user-defined field.
ATTRIBUTE7	USER-DEFINED	This is a user-defined field.
ATTRIBUTE8	USER-DEFINED	This is a user-defined field.
ATTRIBUTE9	USER-DEFINED	This is a user-defined field.
ATTRIBUTE10	USER-DEFINED	This is a user-defined field.

Field Name	Data Type	Description
EDI_IND	VARCHAR2(1)	EDI Indicator
AUTOFAX_IND	VARCHAR2(1)	Auto Fax Indicator
FAXED_DATE	DATE	Date Faxed
FAX_NO	VARCHAR2(30)	Fax Number
SHIP_MEMO_NO	VARCHAR2(10)	Ship Memo Number
CONFIRMATION_TYPE	VARCHAR2(10)	Confirmation Type
AUTO_PAY_IND	CHAR(1)	Auto-Pay Indicator
ROUTING_LIST_ID	VARCHAR2(10)	Routing List ID
PO_DESC	VARCHAR2(2000)	PO Description
VENDOR_CLASS	VARCHAR2(5)	Vendor Class
ERS_INVOICE_SEQ	NUMBER	
DIRECTION*	CHAR(1)	
SOURCE_SYSTEM	VARCHAR2(3)	
INT_BATCH_NUMBER	NUMBER	
JOB_STATUS	VARCHAR2(15)	
JOB_MESSAGE	VARCHAR2(4000)	
INT_INSTANCE_ID	NUMBER	
The following fields are available for Line records.		
PO_ITEM*	VARCHAR2(3)	PO line item number
PO_ITEM_TYPE	CHAR(1)	PO item type - Code Table 155
STOREROOM	VARCHAR2(3)	Storeroom <sup>6</sup>
STOCK_TYPE	VARCHAR2(15)	Stock type code <sup>6</sup>
STOCK_CODE	VARCHAR2(20)	Stock code <sup>6</sup>
ITEM_DESC*	VARCHAR2(2000)	The description of the item to be purchased <sup>6</sup>
WORK_ORDER_NO	VARCHAR2(7)	Work order number
WORK_ORDER_TASK_NO	VARCHAR2(2)	Work order task number
UNIT_OF_PURCHASE*	VARCHAR2(4)	Unit of purchase
PI_RATIO*	NUMBER	Purchase-to-Issue ratio <sup>6</sup>
REQUISITION_NO	VARCHAR2(7)	Requisition number
REQUISITION_ITEM	VARCHAR2(3)	Requisition item number
PRINT_PRICE_IND	CHAR(1)	Y indicates this price is to be printed on the PO

Field Name	Data Type	Description
PRINT_ITEM_IND	CHAR(1)	Y indicates this line item is to be printed on the PO
ZERO_DOLLAR_IND	CHAR(1)	Y indicates that the unit price can be zero
PO_QUANTITY*	NUMBER(15,5)	The PO quantity <sup>7</sup>
UNIT_PRICE*	NUMBER(17,4)	Unit price <sup>7</sup>
ITEM_TOTAL_AMOUNT*	NUMBER(15,2)	Total amount of PO item <sup>7, 8</sup>
FIRST_RECEIVED_DATE	DATE	
LAST_RECEIVED_DATE	DATE	
RECEIVED_QUANTITY	NUMBER(15,5)	
RETURNED_QUANTITY	NUMBER(15,5)	
RETURNED_CREDIT_QUANTITY	NUMBER(15,5)	
RECEIVED_NET_QUANTITY	NUMBER(15,5)	
INVOICED_QUANTITY	NUMBER(15,5)	
INVOICED_AMOUNT	NUMBER(15,2)	
COMPLETE_IND	CHAR(1)	
INVOICE_MATCHED_QUANTITY	NUMBER(15,5)	
FROM_WORK_ORDER_IND	CHAR(1)	From work order indication
CONTRACT_TYPE	CHAR(1)	The type of contract applicable to this item
DO_NOT_SUBSTITUTE_IND	CHAR(1)	Y indicates a substitute cannot be used
STATE_PROVINCE_TAX_ITEM_AMOUNT	NUMBER(15,2)	Total amount of the state or province tax
FEDERAL_TAX_ITEM_AMOUNT	NUMBER(15,2)	Total amount of federal tax
DUTY_ITEM_AMOUNT	NUMBER(15,2)	Total amount of duty
QUOTE_NO	VARCHAR2(8)	Quote number on which item was sent for bid
QUOTE_VENDOR_CODE	VARCHAR2(30)	Quote vendor code
FEDERAL_TAX_RATE	NUMBER(7,4)	Federal tax rate
STATE_PROVINCE_TAX_RATE	NUMBER(7,4)	The state or province tax rate
DUTY_RATE	NUMBER(7,4)	Import duty rate
BLANKET_ITEM	VARCHAR2(5)	Blanket contract item number <sup>9</sup>
MANUFACTURER_CODE	VARCHAR2(20)	The code for the manufacturer of the item

Field Name	Data Type	Description
MANUFACTURER_PART_NO	VARCHAR2(50)	The manufacturer's part number
VENDOR_PART_NO	VARCHAR2(50)	Vendor part number
LINE_ATTRIBUTE1	VARCHAR2(1)	This is a user-defined field.
LINE_ATTRIBUTE2	VARCHAR2(1)	This is a user-defined field.
LINE_ATTRIBUTE3	VARCHAR2(1)	This is a user-defined field.
LINE_ATTRIBUTE4	VARCHAR2(1)	This is a user-defined field.
LINE_ATTRIBUTE5	VARCHAR2(1)	This is a user-defined field.
LINE_ATTRIBUTE6	VARCHAR2(1)	This is a user-defined field.
LINE_ATTRIBUTE7	VARCHAR2(1)	This is a user-defined field.
LINE_ATTRIBUTE8	VARCHAR2(1)	This is a user-defined field.
LINE_ATTRIBUTE9	VARCHAR2(1)	This is a user-defined field.
LINE_ATTRIBUTE10	VARCHAR2(1)	This is a user-defined field.
QUOTE_ITEM	VARCHAR2(5)	Quote item on which item was sent for bid
INSPECTION_TYPE	VARCHAR2(20)	
IN_RECEIPT_QUANTITY	NUMBER(15,5)	
PROCUREMENT_LEVEL	VARCHAR2(5)	
QUALITY_IND	VARCHAR2(1)	
EXPEDITING_IND	VARCHAR2(1)	
COURTESY_STORES_IND	VARCHAR2(1)	
TRANSFER_PO_NO	VARCHAR2(10)	
TRANSFER_PO_ITEM	VARCHAR2(3)	
FULLY_RECEIVED_DATE	DATE	
DISCREPANT_IND	VARCHAR2(1)	
SENT_ALERT_IND	VARCHAR2(1)	
MAXIMUM_SCORE	NUMBER	
ITEM_SCORE	NUMBER	
ON_TIME_IND	VARCHAR2(1)	
ASSET_ID	VARCHAR2(15)	
ASSET_RECORD_TYPE	VARCHAR2(1)	
COMPONENT_ID	VARCHAR2(15)	
The following fields are available for Default Account and Line Account Splits.		
ACCOUNT_NO*	VARCHAR2(75)	Account Number
EXPENSE_CODE*	VARCHAR2(10)	Expense Code for Account Number - Business Rule "EXPENSE CODES".

Field Name	Data Type	Description
PERCENT_SPLIT	NUMBER(8,5)	Percentage of the Line Item to be Charged to this Account No/Expense Code <sup>10</sup>
UNITS	NUMBER(12,5)	

\* required fields

<sup>1</sup> This number needs to be passed so the appropriate matching of POs, PO lines, and PO line account splits can be done if there is more than one revision of the same PO between two interface runs. This number may or may not be the same as the revision numbers generated by Oracle Utilities Work and Asset Management. Within the application, new POs are created with PO Revision No = '000', and each subsequent revision sent over the interface is generated with a PO Revision No incremented by one.

<sup>2</sup> Division, Address, City, State or Province, Postal Code, Vendor Contact, Vendor Phone Number are all going to be recreated from the appropriate Oracle Utilities Work & Asset Management vendor record if they are not brought in.

<sup>3</sup> This entry will be recalculated after all the items for a specific PO are entered. If this amount is different than originally entered warning is logged into the job manager log. This will not stop the processing, though.

<sup>4</sup> If all the items on a PO are assigned the same account split only default account entries are required with the specified PO Number and Record Type set to 'D'. This could be triggered directly by setting this field to 'Y', but it is not necessary since presence of the default record type will trigger it anyway.

<sup>5</sup> Currency Code can be provided instead of Exchange Rate, since the appropriate value will be pulled from the appropriate Code Table.

<sup>6</sup> It might be possible to supply only stock code. The attempt will be made to fill in other required information. The information on Storeroom, Item Description, Unit of Purchase, and P/I Ratio will be pulled from the appropriate tables if present in Oracle Utilities Work & Asset Management. This is required field for 'INVENTORY' and 'EXPENSE' Stock Types.

<sup>7</sup> Between PO Quantity, Unit Price, and Total Amount, only two out of three fields are required.

<sup>8</sup> This entry will also be recalculated based on PO Quantity and Unit Price. If this amount is different than originally entered warning is logged into the job manager log. This will not stop the processing, though.

<sup>9</sup> Field is NOT available for Change Orders.

<sup>10</sup> Must sum to 100%.

---

# Chapter 23

## Receipt

The Receipt interface allows Oracle Utilities Work and Asset Management to accept receipts from an external receiving system. Receipts are processed as if created on-line (alerts sent, inventory updated PO updated, component IDs and lot IDs accepted).

This interface requires:

**Interface Table:** WAIF\_RECEIPT

**Stored Procedure:** WIFP\_RECEIPT\_INTERFACE

Quantities and transactions types are validated. Receipts are logged against PO items. If an alert username is given, an alert is generated for that user.

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_RECEIPT_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

### Keywords and Parameters

**JOB\_NUMBER** = Integer – The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN** = VARCHAR2 – Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not implemented for this interface.

**PURGE\_IN** = VARCHAR2 – Single character that identifies how records will be removed from the interface table.

**Y** = Records will be deleted from the interface table after they are processed.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_RECEIPT\_INTERFACE runs again.

**OPTION\_IN** = VARCHAR2 - Not used in this interface.

**PRE\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call before IN records in the interface table are processed.

**POST\_IN** = VARCHAR2(200) – (Optional) Character string that identifies the custom stored procedure to call after IN records in the interface table are processed.

## Sequence Numbers

N/A

## Interface Table Layout

### WAIF\_RECEIPT

Field Name	Data Type	Description
PLANT*	VARCHAR2(3)	Plant Code
PO_NO*	VARCHAR2(10)	Purchase Order Number
PO_ITEM*	VARCHAR2(3)	Purchase Order Item Number
RECEIPT_DATE*	DATE	Receipt Date
RECEIPT_QUANTITY*	NUMBER(15,5)	Receipt Quantity – Positive for 'RE', Negative for 'RT','RC'
RECEIPT_TYPE*	VARCHAR2(2)	Receipt Type 'RE','RT','RC'
RECEIPT_USERNAME*	VARCHAR2(30)	Received by Username
COMPONENT_ID	VARCHAR2(15)	Component ID – If not NULL then Quantity must be 1 or -1, stock item must be marked as trackable, and if component ID exists its status must be INACTIVE, if it doesn't exists then a new component ID is created.
WAYBILL_NO	VARCHAR2(20)	Waybill Number
PACKING_SLIP	VARCHAR2(20)	Packing Slip
LOT_ID	VARCHAR2(20)	Lot ID – if not NULL then new lot will be created, existing lot will be updated
ALERT_USERNAME	VARCHAR2(30)	Username to send alert to – This is in addition to normal Oracle Utilities Work and Asset Management Alerts upon receipts.

Field Name	Data Type	Description
ALERT_COMMENTS	VARCHAR2(2000)	Additional comments to be added to the alert.
UPDATE_INVENTORY_IND	VARCHAR2(1)	'Y' – Inventory quantities will be updated 'N' – Inventory will not be updated DEFAULT IS 'Y'
UPDATE_PO_IND	VARCHAR2(1)	'Y' – PO quantities will be updated 'N' – PO quantities will not be updated DEFAULT IS 'Y'
SEQUENCE	NUMBER	Receipts will be processed in this order. Ascending order (after Receipt Date).
SENT_TO_INTERFACE_IND	VARCHAR2(1)	Y If PO Was Successfully Processed
DIRECTION*	CHAR(1)	
SOURCE_SYSTEM	VARCHAR2(3)	
INT_BATCH_NUMBER	NUMBER	
JOB_STATUS	VARCHAR2(15)	
JOB_MESSAGE	VARCHAR2(4000)	
INT_INSTANCE_ID	NUMBER	
JOB_SEQ_NO	NUMBER	

\* required fields



---

# Chapter 24

## Requisition

The Requisition interface provides a means to export requisition information maintained in Oracle Utilities Work and Asset Management. Only Approved requisitions are sent.

This interface requires:

**Interface Table:** WAIF\_REQUISITION

**Stored Procedure:** WIFP\_REQUISITION\_INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_REQUISITION_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

The stored procedure processes OUT data, data being sent from the Oracle Utilities Work and Asset Management Requisition module to a client's Purchasing System.

### Keywords and Parameters

**JOB\_IN** = Integer – The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN** = VARCHAR2 – Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN** = VARCHAR2 - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application. This option is not yet implemented for this interface.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table.

**PURGE\_IN** = VARCHAR2 – (Optional) Single character that identifies how records will be removed from the interface table.

**OPTION\_IN** = VARCHAR2 - There are no options for this interface.

**Y** = Records will be deleted from the interface table after they are processed for IN records and before that are processed for OUT records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_REQUISITION\_INTERFACE runs again.

**PRE\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call before records in the interface table are processed.

(Note: This option currently is **NOT** implemented.)

**POST\_IN** = VARCHAR2 – (Optional) Character string that identifies the custom stored procedure to call after IN records in the interface table are processed.

## Sequence Numbers

SA\_REQUISITION

## Interface Table Layout

### WAIF\_REQUISITION

(same fields as SA\_REQUISITION, SA\_REQUISITION\_ITEM and SA\_REQUISITION\_ITEM\_ACCOUNT)

Field Name	Data Type	Description
The following columns are shared by multiple record types ('Header', 'Line' and 'Account').		
PLANT*	VARCHAR2(3)	
REQUISITION_NO*	VARCHAR2(7)	
REQUISITION_ITEM*	VARCHAR2(3)	Line and Account records only
FEDERAL_TAX_CODE	CHAR(1)	Header and Line records only
STATE_PROVINCE_TAX_CODE	CHAR(1)	Header and Line records only
DUTY_CODE	CHAR(1)	Header and Line records only
LAST_UPDATE_DATE	DATE	Header and Line records only
LAST_UPDATE_USER	VARCHAR2(30)	Header and Line records only
CREATION_DATE	DATE	Header and Line records only

Field Name	Data Type	Description
CREATED_BY	VARCHAR2(30)	Header and Line records only
REQUIRED_DATE	DATE	Header and Line records only
PROMISE_DATE	DATE	Header and Line records only
DIRECTION*	CHAR(1)	“I” inbound or “O” outbound
RECORD_TYPE*	CHAR(1)	“H” = header, “L” = line, “A” = account
SENT_TO_INTERFACE_IND	CHAR(1)	
DIRECTION	CHAR	
SOURCE_SYSTEM	VARCHAR2	
INT_BATCH_NUMBER	NUMBER	
JOB_STATUS	VARCHAR2	
JOB_MESSAGE	VARCHAR2	
INT_INSTANCE_ID	NUMBER	
COMMODITY_CATEGORY	VARCHAR2	
COMMODITY_NAME	VARCHAR2	
COMMODITY_TYPE	VARCHAR2	
COMMODITY_COMPOSITION	VARCHAR2	
COMMODITY_SIZE	VARCHAR2	
The following columns apply only to the header record.		
REQUISITION_TYPE	CHAR(1)	
BLANKET_CONTRACT_NO	VARCHAR2(10)	
BLANKET_RELEASE_NO	VARCHAR2(4)	
REQUISITION_STATUS	VARCHAR2(20)	
REQUISITION_STATUS_DATE	DATE	
REQUESTOR	VARCHAR2(30)	
REQUESTOR_PHONE_NO	VARCHAR2(30)	
REQUEST_DATE	DATE	
REQUESTOR_DEPARTMENT	VARCHAR2(10)	
VENDOR_CODE	VARCHAR2(30)	
DIVISION	VARCHAR2(40)	
ADDRESS	VARCHAR2(200)	
CITY	VARCHAR2(40)	

Field Name	Data Type	Description
STATE_PROVINCE	VARCHAR2(4)	
POSTAL_CODE	VARCHAR2(10)	
VENDOR_CONTACT	VARCHAR2(20)	
VENDOR_PHONE_NO	VARCHAR2(30)	
TOTAL_AMOUNT	NUMBER(15,2)	
TERMS	VARCHAR2(2)	
CONFIRMATION_IND	CHAR(1)	
CONFIRMATION_CONTACT	VARCHAR2(30)	
CONFIRMATION_DATE	DATE	
CARRIER	VARCHAR2(30)	
FOB	VARCHAR2(20)	
DELIVER_TO	VARCHAR2(20)	
DELIVER_TO_DEPARTMENT	VARCHAR2(10)	
SHIP_TO_CODE	VARCHAR2(3)	
INITIATOR	VARCHAR2(30)	
BUYER	VARCHAR2(3)	
DEFAULT_ACCOUNTS	CHAR(1)	
BUYER_PHONE_NO	VARCHAR2(30)	
NEXT_APPROVER_TITLE	VARCHAR2(6)	
CURRENCY_CODE	VARCHAR2(3)	
EXCHANGE_RATE	NUMBER(15,7)	
ONLY_SUPPLIER_IND	CHAR(1)	
RFQ_IND	CHAR(1)	
PRICE_CODE	CHAR(1)	
END_USER_CODE	CHAR(1)	
STANDARD_TEXT	CHAR(1)	
CREDIT_CARD_HOLDER_NAME	VARCHAR2(30)	
CREDIT_CARD_PURCHASE_IND	CHAR(1)	
NEXT_APPROVER	VARCHAR2(6)	
BLANKET_REVISION_NO	VARCHAR2(3)	
ATTRIBUTE1	User-Defined	
ATTRIBUTE2	User-Defined	
ATTRIBUTE3	User-Defined	

Field Name	Data Type	Description
ATTRIBUTE4	User-Defined	
ATTRIBUTE5	User-Defined	
ATTRIBUTE6	User-Defined	
ATTRIBUTE7	User-Defined	
ATTRIBUTE8	User-Defined	
ATTRIBUTE9	User-Defined	
ATTRIBUTE10	User-Defined	
REQUESTED_DOCUMENT_TYPE	VARCHAR2(15)	
COMMENTS	VARCHAR2(200)	
EDI_IND	VARCHAR2(1)	
AUTOFAX_IND	VARCHAR2(1)	
FAX_NO	VARCHAR2(30)	
SHIP_MEMO_NO	VARCHAR2(10)	
SHIP_MEMO_IND	VARCHAR2(1)	
AUTO_PAY_IND	CHAR(1)	
TEMPLATE_NAME	VARCHAR2(40)	
TEMPLATE_IND	CHAR(1)	
TEMPLATE_STATUS	VARCHAR2(20)	
TEMPLATE_START_DATE	DATE	
TEMPLATE_END_DATE	DATE	
ROUTING_LIST_ID	VARCHAR2(10)	
REQUISITION_DESC	VARCHAR2(2000)	
VENDOR_CLASS	VARCHAR2(5)	
VENDOR_UPDATED_IND	VARCHAR2(1)	
SENT_TO_INTERFACE_IND	VARCHAR2(1)	
JOB_SEQ_NO	NUMBER	
These columns only apply to the Line records.		
REQUISITION_ITEM_STATUS	VARCHAR2(20)	
PO_ITEM_TYPE	CHAR(1)	
STOREROOM	VARCHAR2(3)	
STOCK_TYPE	VARCHAR2(15)	
STOCK_CODE	VARCHAR2(15)	
ITEM_DESC	VARCHAR2(2000)	

Field Name	Data Type	Description
WORK_ORDER_NO	VARCHAR2(7)	
WORK_ORDER_TASK_NO	VARCHAR2(2)	
UNIT_OF_PURCHASE	VARCHAR2(4)	
PI_RATIO	NUMBER	
PRINT_PRICE_IND	CHAR(1)	
PO_NO	VARCHAR2(10)	
PO_ITEM	VARCHAR2(3)	
ZERO_DOLLAR_IND	CHAR(1)	
REQUISITION_QUANTITY	NUMBER(15,5)	
UNIT_PRICE	NUMBER(17,4)	
ITEM_TOTAL_AMOUNT	NUMBER(15,2)	
FROM_WORK_ORDER_IND	CHAR(1)	
CONTRACT_TYPE	CHAR(1)	
DO_NOT_SUBSTITUTE_IND	CHAR(1)	
STATE_PROVINCE_TAX_ITEM_AMOUNT	NUMBER(15,2)	
FEDERAL_TAX_ITEM_AMOUNT	NUMBER(15,2)	
DUTY_ITEM_AMOUNT	NUMBER(15,2)	
QUOTE_NO	VARCHAR2(8)	
QUOTE_VENDOR_CODE	VARCHAR2(30)	
MANUFACTURER_CODE	VARCHAR2(20)	
MANUFACTURER_PART_NO	VARCHAR2(50)	
VENDOR_PART_NO	VARCHAR2(50)	
LINE_ATTRIBUTE1	User-Defined	
LINE_ATTRIBUTE2	User-Defined	
LINE_ATTRIBUTE3	User-Defined	
LINE_ATTRIBUTE4	User-Defined	
LINE_ATTRIBUTE5	User-Defined	
LINE_ATTRIBUTE6	User-Defined	
LINE_ATTRIBUTE7	User-Defined	
LINE_ATTRIBUTE8	User-Defined	
LINE_ATTRIBUTE9	User-Defined	
LINE_ATTRIBUTE10	User-Defined	
QUOTE_ITEM	VARCHAR2(5)	

Field Name	Data Type	Description
PROCUREMENT_LEVEL	VARCHAR2(5)	
QUALITY_CLASS	VARCHAR2(5)	
QUALITY_IND	VARCHAR2(1)	
EXPEDITING_IND	VARCHAR2(1)	
COURTESY_STORES_IND	VARCHAR2(1)	
ASSET_ID	VARCHAR2(15)	
ASSET_RECORD_TYPE	VARCHAR2(1)	
COMPONENT_ID	VARCHAR2(15)	
The following columns apply to the Line Account records.		
ACCOUNT_NO*	VARCHAR2(75)	
EXPENSE_CODE*	VARCHAR2(10)	
PERCENT_SPLIT*	NUMBER(8,5)	
CHARGE_ACCOUNT_ID	NUMBER	
UNITS	NUMBER(12,5)	

\* required fields

---

# Chapter 25

## Service Request

The Service Request interface provides a means to import information used for recording and processing service request records in Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table:** WAIF\_SERVICE\_REQUEST

**Stored Procedure:** WIFP\_SERVICE\_REQUEST\_INTERFACE

**Business Rule:** SERVICE\_REQUEST\_INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

WIFP\_SERVICE\_REQUEST\_INTERFACE(JOB\_IN,  
PLANT\_IN,  
DIRECTION\_IN,  
PURGE\_IN,  
OPTIONS\_IN,  
PRE\_IN,  
POST\_IN)

### Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN =VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table. This option is not yet implemented for this interface.



**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y = If DIRECTION = I**, records will be deleted from the interface table as they are processed.

**If DIRECTION = O**, records will be deleted from the interface table before writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely; this assumes some other process will actually purge the interface table before WIFP\_SERVICE REQUEST\_INTERFACE runs again.

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Business Rule Format

This business rule only applies to pre-existing inbound records.

**RULE ID** = SERVICE REQUEST INTERFACE

**RULE TYPE** = INTERFACE

**RULE STYLE** = LIST

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound service request record that already exists in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_SERVICE REQUEST column to be updated.

**UPDATE** = YES or NO. If this value is set to YES then the corresponding field will be updated.

## Sequence Numbers

SA\_SERVICE\_REQUEST

## Interface Table Layout

### WAIF\_SERVICE REQUEST

(same fields as SA\_SERVICE\_REQUEST)

PLANT*	VARCHAR2(3)
SERVICE_REQUEST_NO*	VARCHAR2(7)
SERVICE_REQUEST_STATUS*	VARCHAR2(20)
SERVICE_REQUEST_STATUS_DATE*	DATE

SERVICE_REQUEST_TYPE	VARCHAR2(15)
CALL_BACK_READY	VARCHAR2(1)
CUSTOMER_CALL_BACK_COMPLETE	VARCHAR2(1)
DISPATCHER	VARCHAR2(30)
NEXT_APPROVER	VARCHAR2(6)
CREATED_DATE*	DATE
REQUESTED_DATE	DATE
IN_PROGRESS_DATE	DATE
FINISHED_DATE	DATE
CREATED_BY*	VARCHAR2(30)
LAST_UPDATE_DATE*	DATE
LAST_UPDATE_USER*	VARCHAR2(30)
PROBLEM_CODE	VARCHAR2(10)
PROBLEM_DESCRIPTION	VARCHAR2(2000)
CUSTOMER_ID	VARCHAR2(20)
CUSTOMER_LAST_NAME	VARCHAR2(30)
CUSTOMER_FIRST_NAME	VARCHAR2(30)
CUSTOMER_MIDDLE_NAME	VARCHAR2(30)
CUSTOMER_PHONE	VARCHAR2(30)
CUSTOMER_PHONE_EXT	VARCHAR2(5)
CUSTOMER_CALL_BACK	VARCHAR2(1)
REPORTED_BY_LAST_NAME	VARCHAR2(30)
REPORTED_BY_FIRST_NAME	VARCHAR2(30)
REPORTED_BY_MIDDLE_NAME	VARCHAR2(30)
REPORTED_BY_PHONE	VARCHAR2(30)
REPORTED_BY_PHONE_EXT	VARCHAR2(5)
REPORTED_BY_CALL_BACK	VARCHAR2(1)
COMPANY	VARCHAR2(60)
STREET_NUMBER	NUMBER
NUMBER_SUFFIX	VARCHAR2(5)
STREET_NAME	VARCHAR2(40)
STREET_DIRECTION	VARCHAR2(3)
SUITE	VARCHAR2(10)

POST_OFFICE_BOX	VARCHAR2(10)
CITY	VARCHAR2(40)
STATE_PROVINCE	VARCHAR2(4)
POSTAL_CODE	VARCHAR2(15)
CUSTOMER_ADDRESS_ID	NUMBER
DEPARTMENT	VARCHAR2(10)
AREA	VARCHAR2(10)
ACCOUNT_NO	VARCHAR2(75)
WORK_ORDER_NO	VARCHAR2(7)
WORK_ORDER_TASK_NO	VARCHAR2(2)
CREW	VARCHAR2(5)
BACKLOG_GROUP	VARCHAR2(6)
ACTUAL_START_DATE	DATE
ACTUAL_FINISH_DATE	DATE
ACTUAL_DURATION	NUMBER
WORK_COMPLETED	VARCHAR2(1)
FAILURE_CODE	VARCHAR2(10)
REPAIR_CODE	VARCHAR2(10)
FURTHER_ACTION	VARCHAR2(10)
COMMENTS	VARCHAR2(2000)
INSPECTED_BY	VARCHAR2(30)
INSPECTED_BY_DATE	DATE
SIGNOFF_BY	VARCHAR2(30)
SIGNOFF_BY_DATE	DATE
CLOSED_BY	VARCHAR2(30)
CLOSED_BY_DATE	DATE
CUSTOMER_BILLING_REQUIRED	VARCHAR2(1)
DEPOSIT_DATE	DATE
DEPOSIT_TYPE	VARCHAR2(20)
DEPOSIT_AMOUNT	NUMBER
BILL_NO	VARCHAR2(20)
BILL_DATE	DATE
BILL_AMOUNT	NUMBER

PAYMENT_DATE	DATE
PAYMENT_TYPE	VARCHAR2(20)
PAYMENT_AMOUNT	NUMBER
ASSET_POSTED_IND	VARCHAR2(1)
ASSET_POSTED_DATE	DATE
ATTRIBUTE1	User-Defined
ATTRIBUTE2	User-Defined
ATTRIBUTE3	User-Defined
ATTRIBUTE4	User-Defined
ATTRIBUTE5	User-Defined
ATTRIBUTE6	User-Defined
ATTRIBUTE7	User-Defined
ATTRIBUTE8	User-Defined
ATTRIBUTE9	User-Defined
ATTRIBUTE10	User-Defined
NUMBER_PREFIX	VARCHAR2(3)
DEPOSIT_CHECK_NO	NUMBER(10,0)
PAYMENT_CHECK_NO	NUMBER(10,0)
TAX_ID	VARCHAR2(20)
CUSTOMER_HOME_PHONE	VARCHAR2(30)
REPORTED_BY_HOME_PHONE	VARCHAR2(30)
PROBLEM_LAST_NAME	VARCHAR2(30)
PROBLEM_FIRST_NAME	VARCHAR2(30)
PROBLEM_NUMBER_PREFIX	VARCHAR2(3)
PROBLEM_STREET_NUMBER	NUMBER
PROBLEM_NUMBER_SUFFIX	VARCHAR2(5)
PROBLEM_STREET_NAME	VARCHAR2(40)
PROBLEM_STREET_DIRECTION	VARCHAR2(3)
PROBLEM_SUITE	VARCHAR2(10)
PROBLEM_CROSS_STREET	VARCHAR2(40)
PROBLEM_CITY	VARCHAR2(40)
PROBLEM_STATE_PROVINCE	VARCHAR2(4)
PROBLEM_POSTAL_CODE	VARCHAR2(10)

PROBLEM_PHONE	VARCHAR2(30)
PROBLEM_PHONE_EXT	VARCHAR2(5)
PROBLEM_HOME_PHONE	VARCHAR2(30)
PROBLEM_CALL_BACK_IND	VARCHAR2(1)
PROBLEM_SAME_AS_CUSTOMER_IND	VARCHAR2(1)
REPORTED_BY_NUMBER_PREFIX	VARCHAR2(3)
REPORTED_BY_STREET_NUMBER	NUMBER
REPORTED_BY_NUMBER_SUFFIX	VARCHAR2(5)
REPORTED_BY_STREET_NAME	VARCHAR2(40)
REPORTED_BY_STREET_DIRECTION	VARCHAR2(3)
REPORTED_BY_SUITE	VARCHAR2(10)
REPORTED_BY_CITY	VARCHAR2(40)
REPORTED_BY_STATE_PROVINCE	VARCHAR2(4)
REPORTED_BY_POSTAL_CODE	VARCHAR2(10)
REPORTED_BY_POST_OFFICE_BOX	VARCHAR2(10)
REPORTED_BY_SAME_PROBLEM_IND	VARCHAR2(1)
TYPE_OF_WORK	VARCHAR2(20)
TYPE_OF_WORK_UNITS	VARCHAR2(20)
TYPE_OF_WORK_AMOUNT	NUMBER
STREET_NUMBER_CHAR	VARCHAR2(10)
PROBLEM_STREET_NUMBER_CHAR	VARCHAR2(10)
REPORTED_BY_STREET_NUMBER_CHAR	VARCHAR2(10)
CLOSE_WO_IND	VARCHAR2(1)
COMPONENT_CODE	VARCHAR2(10)
FAILURE_MODE	VARCHAR2(10)
ROOT_CAUSE	VARCHAR2(10)
SENT_TO_INTERFACE_IND	VARCHAR2(1)
JOB_SEQ_NO	NUMBER
FA_ID	VARCHAR2(38)
SCHEDULE_DATE	DATE
SERVICE_POINT_ASSET_ID	VARCHAR2(15)
PROBLEM_ADDRESS_ID	NUMBER

DIRECTION*	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

\* required fields

---

# Chapter 26

## Storeroom

The Storeroom interface provides a means to import and export storeroom quantities and vales in Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table:** WAIF\_STOREROOM

**Stored Procedure:** WIFP\_STOREROOM\_INTERFACE

**Business Rule:** STOREROOM INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

WIFP\_STOREROOM\_INTERFACE(JOB\_IN,  
PLANT\_IN,  
DIRECTION\_IN,  
PURGE\_IN,  
OPTION\_IN,  
PRE\_IN,  
POST\_IN)

Only processed records are purged.

The job number is used to determine which records to inactivate, if the Inactivate option is chosen.

On outbound, if you select the Update option only those records updated after the last job run are interfaced. Unprocessed records are not purged and you may have more than one record for the stock code and storeroom code in the table.

If you do not select the Update option, purge "Y" will remove all existing outbound records and replace them with new ones. Unless you are sending only updates, you should always set the purge flag to "Y".

## Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to the Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the interface table to the Oracle Utilities Work and Asset Management application.

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y** = If **DIRECTION = I**, records will be deleted from the interface table as they are processed.

If **DIRECTION = O**, records will be deleted from the interface table before writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_STOREROOM\_INTERFACE runs again.

**OPTION\_IN = VARCHAR2** - (Optional) Character string that identifies the special processing options.

If **DIRECTION = I** and...

- **OPTIONS = I** – The interface procedure will ‘Inactivate’ all pre-existing storeroom records in the application that were not created or updated, by the current run of the interface procedure. This assumes that the interface table contains all the active storeroom items and that all other storeroom items are no longer active.

If **DIRECTION = O** and...

- **OPTIONS = A** - The interface procedure will **only** write ‘Active’ storeroom records to the interface table.
- **OPTIONS = U** - The interface procedure will **only** write storeroom records that have been updated since the last run of the interface procedure to the interface table.
- **OPTIONS = AU** - Both of the above options apply.

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Business Rule Format

This business rule only applies to pre-existing inbound records.

**RULE ID = STOREROOM INTERFACE**

**RULE TYPE = INTERFACE**

**RULE STYLE = LIST**



**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound storeroom record that already exists in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_STOREROOM column to be updated.

**UPDATE** = YES or NO. If this value is set to YES then the corresponding field will be updated.

## Interface Table Layout

### WAIF\_STOREROOM

(Same fields as SA\_STOREROOM)

Field Name	Data Type
PLANT*	VARCHAR2(3)
STOCK_CODE*	VARCHAR2(15)
STOREROOM*	VARCHAR2(3)
PRIMARY_BIN	VARCHAR2(20)
STOREROOM_STATUS*	VARCHAR2(20)
AUTO_REORDER_IND	CHAR(1)
INVENTORY_QUANTITY	NUMBER(15,5)
ON_ORDER_QUANTITY	NUMBER(15,5)
ON_DEMAND_QUANTITY	NUMBER(15,5)
MINIMUM_QUANTITY	NUMBER(10,2)
REORDER_POINT_QUANTITY	NUMBER(10,2)
REORDER_QUANTITY	NUMBER(10,2)
PRICE_TYPE	VARCHAR2(10)
AVERAGE_UNIT_PRICE	NUMBER(15,4)
STANDARD_PRICE	NUMBER(15,4)
ACTIVITY_IND	CHAR(1)
NOTES_IND	CHAR(1)
TOTAL_VALUE	NUMBER(15,2)
MAXIMUM_QUANTITY	NUMBER(10,2)
INSPECTION_IND	CHAR(1)
COMPONENT_TRACKING_IND	CHAR(1)
TRANSFER_QUANTITY	NUMBER(15,5)
IN_PHYSICAL_INVENTORY_IND	CHAR(1)
REPAIRABLE_IND	CHAR(1)

Field Name	Data Type
ASSET_ACCOUNT_NO	VARCHAR2(75)
ASSET_EXPENSE_CODE	VARCHAR2(10)
CREDIT_ACCOUNT_NO	VARCHAR2(75)
CREDIT_EXPENSE_CODE	VARCHAR2(10)
DEBIT_EXPENSE_CODE	VARCHAR2(10)
REPAIR_QUANTITY	NUMBER(10,2)
SOURCE_CODE	VARCHAR2(4)
USE_DEFAULT_MARKUP_RATE	CHAR(1)
MARKUP_RATE	NUMBER(5,3)
QC_REQUIRED_IND	CHAR(1)
QC_INSPECTOR	VARCHAR2(6)
MTD_USAGE_QUANTITY	NUMBER(15,5)
YTD_USAGE_QUANTITY	NUMBER(15,5)
LAST_INVENTORY_NO	NUMBER
LAST_INVENTORY_DATE	DATE
LAST_PO_NO	VARCHAR2(10)
LAST_PO_DATE	DATE
LAST_PO_ITEM	VARCHAR2(3)
LAST_BLANKET_PO_NO	VARCHAR2(10)
LAST_BLANKET_RELEASE_NO	NUMBER(4)
LAST_INVOICE_PRICE	NUMBER(15,4)
LAST_ISSUE_DATE	DATE
LAST_UPDATE_DATE*	DATE
USE_MARKUP_IND	CHAR(1)
LOT_IND	CHAR(1)
ABC_CLASS	CHAR(1)
ABC_CLASS_DATE	DATE
SET_ABC_CLASS_IND	CHAR(1)
SET_REORDER_POINT_PER_ABC_IND	CHAR(1)
SET_MAXIMUM_PER_ABC_IND	CHAR(1)
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_DATE*	DATE
CREATED_BY*	VARCHAR2(30)

Field Name	Data Type
ATTRIBUTE1	USER-DEFINED
ATTRIBUTE2	USER-DEFINED
ATTRIBUTE3	USER-DEFINED
ATTRIBUTE4	USER-DEFINED
ATTRIBUTE5	USER-DEFINED
ATTRIBUTE6	USER-DEFINED
ATTRIBUTE7	USER-DEFINED
ATTRIBUTE8	USER-DEFINED
ATTRIBUTE9	USER-DEFINED
ATTRIBUTE10	USER-DEFINED
RETURN_CREDIT_VALUE	NUMBER(15,2)
AFTER_REPAIR_VALUE	NUMBER(15,2)
TOTAL_REPAIR_COUNT	NUMBER(10)
TOTAL_REPAIR_AMOUNT	NUMBER(15,2)
STOCK_TYPE	VARCHAR2(15)
ALLOCATED_QUANTITY	NUMBER(10,2)
IN_RECEIPT_QUANTITY	NUMBER(15,5)
DISCREPANT_HOLD_QUANTITY	NUMBER(10,2)
INSPECTION_HOLD_QUANTITY	NUMBER(10,2)
MANUAL_HOLD_QUANTITY	NUMBER(10,2)
ALLOCATE_IND	VARCHAR2(1)
SAFETY_STOCK_QUANTITY	NUMBER(10,2)
ECONOMIC_ORDER_QUANTITY	NUMBER(10,2)
MAXIMUM_ORDER_QUANTITY	NUMBER(10,2)
MULTIPLE_ORDER_QUANTITY	NUMBER(10,2)
MATERIAL_PLANNER	VARCHAR2(6)
BUYER	VARCHAR2(3)
FEDERAL_TAX_CODE	CHAR(1)
STATE_PROVINCE_TAX_CODE	CHAR(1)
DUTY_CODE	CHAR(1)
MANUFACTURER_CODE	VARCHAR2(20)
MANUFACTURER_PART_NO	VARCHAR2(50)
VENDOR_CODE	VARCHAR2(30)

Field Name	Data Type
VENDOR_PART_NO	VARCHAR2(50)
SENT_TO_INTERFACE_IND	VARCHAR2(1)
JOB_SEQ_NO	NUMBER
REORDER_REVIEWER_CODE	VARCHAR2(10)
LEAD_TIME	NUMBER(3)
ADDITIONAL_LEAD_TIME	NUMBER(3)
DIRECTION*	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

\* required fields

---

# Chapter 27

## Timekeeping

The Timekeeping interface provides a means to import and export the payroll information used for recording timecards in Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table:** WAIF\_TIMEKEEPING

**Stored Procedure:** WIFP\_TIMEKEEPING\_INTERFACE

In the application, timesheet headers may be reopened and new lines added to them. In the case of outbound processing, there may be more than one copy of the header record in the interface table. You must ensure that each line is processed only once, and not once PER HEADER. You may need to use the job sequence number or the update dates to match each line up with the proper version of the header. In the case of inbound processing, you must ensure that there is only one header record per employee number and timesheet date. This interface will not add new timesheet lines to existing timesheet headers. If this is attempted, the interface will consider these headers to be in error, and will not process them. If you need to add timesheets to existing headers, you must do that manually using the application.

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

```
WIFP_TIMEKEEPING_INTERFACE(JOB_IN,  
PLANT_IN,  
DIRECTION_IN,  
PURGE_IN,  
OPTION_IN,  
PRE_IN,  
POST_IN)
```

The stored procedure processes both IN data, data being put into the Oracle Utilities Work and Asset Management Timekeeping module, and OUT data, data being sent from the Oracle Utilities Work and Asset Management' Timekeeping module to a client's Timekeeping System.

## Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies which record type(s) to process.

**I** = Indicates that IN records in the interface table are to be processed.

All records in the interface table with DIRECTION = 'I' will be moved to the appropriate Oracle Utilities Work and Asset Management table as is. Records will be removed from the interface table if successfully moved, an error will be recorded in the job manager log and the record will be left in the interface table if an error is encountered.

**O** = Indicates that OUT records are to be written to the interface table.

All POSTED timesheets that have lines that haven't already been sent to the interface table will be written to the interface table with DIRECTION = 'O'.

After writing a timesheet to the interface table, the lines will be marked as SENT TO INTERFACE.

**PURGE\_IN = VARCHAR2** - Single character that identifies if records will be removed from the interface table.

**Y** = Records will be deleted from the interface table after they are processed for IN records and before that are processed for OUT records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_TIMEKEEPING\_INTERFACE runs again.

**OPTION\_IN = VARCHAR2** - Not used in this interface.

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before records in the interface table are processed.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after IN records in the interface table are processed.

## Sequence Numbers

SA\_TIMESHEET

## Interface Table Layout

### WAIF\_TIMEKEEPING

(same fields as SA\_TIMESHEET\_HEADER and SA\_TIMESHEET)

Field Name	Data Type
The following fields are required for both Header and Line records.	
PLANT*	VARCHAR2(3)
EMPLOYEE_NO*	VARCHAR2(6)

Field Name	Data Type
TIMESHEET_DATE*	DATE
CRAFT*	VARCHAR2(5)
LAST_UPDATE_USERNAME*	VARCHAR2(30)
LAST_UPDATE_DATE*	DATE
PERIOD_YEAR*	VARCHAR2(4)
PERIOD_NO*	VARCHAR2(2)
PAYROLL_STATUS*	CHAR(1) O = Open, C = Closed
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_BY*	VARCHAR2(30)
CREATED_DATE*	DATE
WEB_SERVICE_INBOUND_ID	NUMBER
DIRECTION*	CHAR(1) – ‘I’nbound or ‘O’utbound
RECORD_TYPE*	CHAR(1) – ‘H’header or ‘L’ine
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER
The following fields available for Header Records	
EMPLOYEE_NAME	VARCHAR2(45)
APPROVAL_STATUS*	VARCHAR2(20)
SUPERVISOR_TITLE	VARCHAR2(6)
SUPERVISOR_NAME	VARCHAR2(60)
CREW	VARCHAR2(5)
DEPARTMENT	VARCHAR2(10)
AREA	VARCHAR2(10)
NEXT_APPROVER	VARCHAR2(6)
WORK_START_TIME	VARCHAR2(5)
WORK_STOP_TIME	VARCHAR2(5)
AUTO_CREATE_TIMESHEET_IND*	VARCHAR2(1)
SOCIAL_SECURITY_NO	VARCHAR2(9) – from SA_EMPLOYEE
The following fields available for Line Records	
CHARGE_TYPE*	CHAR(1)

Field Name	Data Type
CHARGE_NO*	VARCHAR2(15)
WORK_ORDER_TASK_NO	VARCHAR2(2)
TIMESHEET_STATUS*	VARCHAR2(20)
REGULAR_SHIFT	VARCHAR2(5)
REGULAR_SHIFT_DIFFERENTIAL	NUMBER(8,4)
REGULAR_HOURS	NUMBER(10,2)
PREMIUM_SHIFT	VARCHAR2(5)
PREMIUM_SHIFT_DIFFERENTIAL	NUMBER(8,4)
PREMIUM_TYPE	VARCHAR2(6)
PREMIUM_HOURS	NUMBER(10,2)
COMMENTS	VARCHAR2(2000)
CHARGE_DESC	VARCHAR2(1)
REGULAR_TYPE	VARCHAR2(6)
ASSET_RECORD_TYPE	VARCHAR2(1)
COMP_TIME_HOURS	NUMBER(10,2)
ACCRUED_COMP_TIME	NUMBER(10,2)
APPROVER_NAME	VARCHAR2(30)
APPROVAL_DATE	DATE
ACCOUNT_NO*	VARCHAR2(75)
CHARGE_TYPE_GROUP*	VARCHAR2(30)
REGULAR_WAGE_RATE	NUMBER(8,4)
REGULAR_AMOUNT	NUMBER(10,2)
REGULAR_SHIFT_AMOUNT	NUMBER(10,2)
REGULAR_EXPENSE_CODE	VARCHAR2(10)
PREMIUM_MULTIPLIER	NUMBER(8,5)
PREMIUM_AMOUNT	NUMBER(10,2)
PREMIUM_SHIFT_AMOUNT	NUMBER(10,2)
PREMIUM_EXPENSE_CODE	VARCHAR2(10)
COMP_TIME_AMOUNT	NUMBER(10,2)
COMP_TIME_SHIFT_AMOUNT	NUMBER(10,2)
SENT_TO_INTERFACE_IND	CHAR(1)
REGULAR_MULTIPLIER	NUMBER(8,5)
REGULAR_ADDER	NUMBER(5,2)



Field Name	Data Type
PREMIUM_ADDER	NUMBER(5,2)
REGULAR_OT_RATE_MULTIPLIER	NUMBER(8,5)
PREMIUM_OT_RATE_MULTIPLIER	NUMBER(8,5)
ATTRIBUTE1	User-Defined
ATTRIBUTE2	User-Defined
ATTRIBUTE3	User-Defined
ATTRIBUTE4	User-Defined
ATTRIBUTE5	User-Defined
ATTRIBUTE6	User-Defined
ATTRIBUTE7	User-Defined
ATTRIBUTE8	User-Defined
ATTRIBUTE9	User-Defined
ATTRIBUTE10	User-Defined
JOB_SEQ_NO	NUMBER
WORK_STARTED_TIME	DATE
WORK_STOPPED_TIME	DATE
TRAVEL_TIME	NUMBER(5,2)
TIMESHEET_SEQ_NO	NUMBER
SUBPROJECT_ID	NUMBER
INVALID_IND	VARCHAR2(1)
WEB_SERVICE_INBOUND_ID	NUMBER

\* required fields

---

# Chapter 28

## Vendor

The Vendor interface provides a means to import and export the Vendor information in Oracle Utilities Work and Asset Management.

This interface requires:

**Interface Table:** WAIF\_VENDOR

**Stored Procedure:** WIFP\_VENDOR\_INTERFACE

**Business Rule:** VENDOR INTERFACE

[Batch Job Procedure](#)

[Keywords and Parameters](#)

[Business Rule Format](#)

[Sequence Numbers](#)

[Interface Table Layout](#)

### Batch Job Procedure

The stored procedure is scheduled and run in the Oracle Utilities Work and Asset Management job manager according to the client's requirements.

WIFP\_VENDOR\_INTERFACE(JOB\_IN,  
PLANT\_IN,  
DIRECTION\_IN,  
PURGE\_IN,  
OPTION\_IN,  
PRE\_IN,  
POST\_IN)

Only processed records are purged.

The job number is used to determine which records to inactivate, if the Inactivate option is chosen.

On outbound, if you select the Update option only those records updated after the last job run are interfaced. Unprocessed records are not purged and you may have more than one record for the vendor code in the table.

If you do not select the Update option, purge "Y" will remove all existing outbound records and replace them with new ones. Unless you are sending only updates, you should always set the purge flag to "Y".

## Keywords and Parameters

**JOB\_IN = Integer** - The job number assigned by the Oracle Utilities Work and Asset Management job manager.

**PLANT\_IN = VARCHAR2** - Character string that identifies the PLANT\_CODE.

**DIRECTION\_IN = VARCHAR2** - Single character that identifies Inbound or Outbound processing.

**I** = Indicates data will be loaded from the interface table to the Oracle Utilities Work and Asset Management application.

**O** = Indicates data will be loaded from the Oracle Utilities Work and Asset Management application to the interface table.

**PURGE\_IN = VARCHAR2** - Single character that identifies how records will be removed from the interface table.

**Y** = If **DIRECTION = I**, records will be deleted from the interface table as they are processed.

If **DIRECTION = O**, records will be deleted from the interface table before writing out new records.

**N** = Records will not be deleted from the interface table. Records will be left in the interface table indefinitely, this assumes some other process will actually purge the interface table before WIFP\_VENDOR\_INTERFACE runs again.

**OPTION\_IN = VARCHAR2** - (Optional) Character string that identifies the special processing options.

If **DIRECTION = I** and...

- **OPTIONS = I** - The interface procedure will 'Inactivate' all pre-existing vendor records in the Oracle Utilities Work and Asset Management application that were not created, or updated, by the current run of the interface procedure. This would be assuming that the interface table contains all the active vendors and that all other vendors are no longer active.

If **DIRECTION = O** and...

- **OPTIONS = A** - The interface procedure will only write 'Active' vendor records to the interface table.
- **OPTIONS = U** - The interface procedure will only write vendor records that have been updated since the last run of the interface procedure to the interface table.
- **OPTIONS = AU** - Both of the above options apply.

**PRE\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call before the interface procedure begins.

**POST\_IN = VARCHAR2** - (Optional) Character string that identifies the custom stored procedure to call after the interface procedure completes.

## Business Rule Format

This business rule only applies to pre-existing inbound records.

**RULE ID = VENDOR INTERFACE**

**RULE TYPE = INTERFACE**

**RULE STYLE = LIST**

**RULE LIMIT** = OFF

**RULE DESCRIPTION** = Controls the fields that will be updated on an inbound vendor record that already exists in Oracle Utilities Work and Asset Management.

**COLUMN NAME** = Name of SA\_VENDOR column to be updated.

**UPDATE** = YES or NO. If this value is set to YES then the corresponding field will be updated.

## Sequence Numbers

SA\_VENDOR

Vendor Codes may have two parts. This only creates the first part.

## Interface Table Layout

### WAIF\_VENDOR

(Same fields as SA\_VENDOR)

Field Name	Data Type
PLANT*	VARCHAR2(3)
VENDOR_CODE*	VARCHAR2(30)
LAST_UPDATE_DATE*	DATE
VENDOR_NAME	VARCHAR2(60)
DIVISION	VARCHAR2(40)
ADDRESS	VARCHAR2(200)
CITY	VARCHAR2(40)
STATE_PROVINCE	VARCHAR2(4)
POSTAL_CODE	VARCHAR2(10)
VENDOR_CONTACT	VARCHAR2(20)
MINIMUM_ORDER_VALUE	NUMBER(15,2)
VENDOR_STATUS*	VARCHAR2(8)
VENDOR_QUALIFIED_DATE	DATE
VENDOR_PHONE_NO	VARCHAR2(30)
VENDOR_FAX_NO	VARCHAR2(30)
EMERGENCY1_PHONE_NO	VARCHAR2(30)
EMERGENCY2_PHONE_NO	VARCHAR2(30)
TERMS	VARCHAR2(2)
CURRENCY_CODE	CHAR(3)
FOB	VARCHAR2(20)

Field Name	Data Type
DISCOUNT_DESC	VARCHAR2(14)
CARRIER	VARCHAR2(30)
PAY_TO_VENDOR_IND	CHAR(1)
PAY_TO_VENDOR_CODE*	VARCHAR2(30)
INSURANCE_EXPIRATION_DATE	DATE
GOVERNMENT_ID_NO	VARCHAR2(12)
WCB_REGISTERED_NO	VARCHAR2(12)
EXPEDITE_CONTACT	VARCHAR2(20)
EXPEDITE_PHONE_NO	VARCHAR2(30)
MBE_IND	CHAR(1)
WBE_IND	CHAR(1)
OVERALL_RATING	CHAR(1)
DELIVERY_CRITERIA	CHAR(1)
PRICE_CRITERIA	CHAR(1)
REJECTION_CRITERIA	CHAR(1)
NEXT_EVALUATION_DATE	DATE
YTD_PURCHASE_AMOUNT	NUMBER(15,2)
YTD_ORDER_COUNT	NUMBER(7)
LAST_YEAR_PURCHASE_AMOUNT	NUMBER(15,2)
LAST_YEAR_ORDER_COUNT	NUMBER(7)
PREVIOUS_YEARS_PURCHASE_AMOUNT	NUMBER(15,2)
PREVIOUS_YEARS_ORDER_COUNT	NUMBER(7)
EVALUATOR	VARCHAR2(20)
LAST_UPDATE_USER*	VARCHAR2(30)
CREATED_DATE*	DATE
CREATED_BY*	VARCHAR2(30)
ATTRIBUTE1	USER-DEFINED
ATTRIBUTE2	USER-DEFINED
ATTRIBUTE3	USER-DEFINED
ATTRIBUTE4	USER-DEFINED
ATTRIBUTE5	USER-DEFINED
ATTRIBUTE6	USER-DEFINED
ATTRIBUTE7	USER-DEFINED

Field Name	Data Type
ATTRIBUTE8	USER-DEFINED
ATTRIBUTE9	USER-DEFINED
ATTRIBUTE10	USER-DEFINED
APPROVED_VENDOR_STATUS	VARCHAR2(8)
APPROVED_VENDOR_STATUS_DATE	DATE
RESTRICTED_IND	CHAR(1)
VENDOR_EMAIL_ADDRESS	VARCHAR2(50)
VENDOR_WEBSITE_ADDRESS	VARCHAR2(100)
VENDOR_CONTACT_TITLE	VARCHAR2(30)
VENDOR_COUNTRY_CODE	VARCHAR2(2)
AUTO_PAY_IND	CHAR(1)
PO_EMAIL_ADDRESS	VARCHAR2(100)
VENDOR_SEGMENT_1	VARCHAR2(30)
VENDOR_SEGMENT_2	VARCHAR2(30)
LOCATION_NAME	VARCHAR2(60)
ORDER_FROM_IND	CHAR(1)
PAY_TO_IND	CHAR(1)
SENT_TO_INTERFACE_IND	VARCHAR2(1)
JOB_SEQ_NO	NUMBER
DIRECTION*	CHAR(1)
SOURCE_SYSTEM	VARCHAR2(3)
INT_BATCH_NUMBER	NUMBER
JOB_STATUS	VARCHAR2(15)
JOB_MESSAGE	VARCHAR2(4000)
INT_INSTANCE_ID	NUMBER

\* required fields

---

# Chapter 29

## Web Services

This section describes basic information regarding Oracle Utilities Work and Asset Management web services to aid in interfacing or integrating with other products.

### Finding Web Services

To open web services for Oracle Utilities Work and Asset Management, append “services” to the end of the application URL.

For example: <http://wamaux09.us.oracle.com:7778/QA19Y/synergen/services>

This opens a listing of all the web services with a link to the associated wsdl.

### Web Service Components

Individual components of the web service include:

[Communication Layer](#)

[Messaging](#)

[Service Provider](#)

[Service Definition File \(WSDL\)](#)

#### Communication Layer

The communications layer specified in the Web Services standards can be about anything that can deliver the SOAP messages (HTTP, FTP, SMTP, MQ, IIOP, etc.). Typically web service communicate over HTTP.

#### Messaging

The messaging payloads being communicated from web services are XML-based in standard SOAP format. The format of the XML embedded inside of the SOAP XML container is defined in the WSDL for the particular service.

#### Service Provider

A web service provider is a program that can be called via HTTP and can respond to SOAP message requests and provide SOAP message responses. The service provider should be an executable piece of business processing logic.

#### Service Definition File (WSDL)

Web Services Description Language (WSDL) is a document written in XML to describe a web service. The document specifies the location of the service and the operations (or methods) the service exposes.

## WSDL Elements

A WSDL document defines a web service using these major elements:

Element	Defines
<portType>	The operations performed by the web service
<message>	The messages used by the web service
<types>	The data types used by the web service
<binding>	The communication protocols used by the web service

A WSDL document can also contain other elements, such as extension elements and a service element that makes it possible to group together the definitions of several web services in one single WSDL document.

### <portType>

The **<portType>** element is the most important WSDL element. It defines a web service, the operations that can be performed, and the messages that are involved. The port defines the connection point to a web service. It can be compared to a function library (or a module, or a class) in a traditional programming language. Each operation can be compared to a function in a traditional programming language.

## WSDL Structure

The following shows the main structure of a WSDL document:

```

<definitions>
  <types>
    definition of types.....
  </types>

  <message>
    definition of a message....
  </message>

  <portType>
    definition of a port.....
  </portType>

  <binding>
    definition of a binding....
  </binding>
</definitions>

```



## WSDL Operation Types

WSDL defines four types:

Type	Definition
One-way	The operation can receive a message but will not return a response
Request-response	The operation can receive a request and will return a response
Solicit-response	The operation can send a request and will wait for a response
Notification	The operation can send a message but will not wait for a response

The request-response type is the most common operation type.

### *Request-Response Operation Example*

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>

<binding type="glossaryTerms" name="b1">
<soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
  <operation>
    <soap:operation
      soapAction="http://example.com/getTerm"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

## WSDL Bindings Information

WSDL bindings information defines the message format and protocol details for a web service. This defines the linkages between the service and the SOAP messages.

### *Binding Element Attributes*

The binding element has two attributes - the **name** attribute and the **type** attribute. The name attribute defines the name of the binding, and the type attribute points to the port for the binding. In the above example the "glossaryTerms" port was used. You can use any name.

The soap:binding element has two attributes - the **style** attribute and the **transport** attribute. The style attribute can be "rpc" or "document". In the above example we use document.

The transport attribute defines the SOAP protocol to use. In this case we use HTTP.

The operation element defines each operation that the port exposes. For each operation the corresponding SOAP action has to be defined. You must also specify how the input and output are encoded. In this case we use "literal".

### Web Services Standard Date Formats

Note the following formats with respect to using dates with web services. Dates should be represented in these exact formats with the same punctuation. Note that the "T" appears literally in the string, to indicate the beginning of the time element, as specified in ISO 8601.

```

Year:
    YYYY (eg 1997)
Year and month:
    YYYY-MM (eg 1997-07)
Complete date:
    YYYY-MM-DD (eg 1997-07-16)
Complete date plus hours and minutes:
    YYYY-MM-DDThh:mmTZD (eg 1997-07-16T19:20+01:00)
Complete date plus hours, minutes and seconds:
    YYYY-MM-DDThh:mm:ssTZD (eg
1997-07-16T19:20:30+01:00)
Complete date plus hours, minutes, seconds and a decimal
fraction of a second
    YYYY-MM-DDThh:mm:ss.sTZD (eg
1997-07-16T19:20:30.45+01:00)
where:
    YYYY = four-digit year
    MM   = two-digit month (01=January, etc.)
    DD   = two-digit day of month (01 through 31)
    hh   = two digits of hour (00 through 23) (am/pm NOT
allowed)
    mm   = two digits of minute (00 through 59)
    ss   = two digits of second (00 through 59)
    s    = one or more digits representing a decimal
fraction of a second
    TZD  = time zone designator (Z or +hh:mm or -hh:mm)
  
```

#### Time Zones

- Times are expressed in UTC (Coordinated Universal Time), with a special UTC designator ("Z").
- Times are expressed in local time, together with a time zone offset in hours and minutes. A time zone offset of "+hh:mm" indicates that the date/time uses a local time zone which is "hh" hours and "mm" minutes ahead of UTC. A time zone offset of "-hh:mm" indicates that the date/time uses a local time zone which is "hh" hours and "mm" minutes behind UTC.

## Design Element – Service Provider

Developing a web service provider requires the following:

1. A WSDL that describes the service, its messages and formats, and how to call it; the WSDL needs to be accessible via HTTP.
2. The ability to consume SOAP messages via HTTP to execute appropriate business logic
3. The ability to execute business logic
4. The ability to respond using a SOAP message response via HTTP to report results of business logic execution

## Toolset

A design time, a developer will choose a toolset and a development approach based on the type of business process you are building into the web service. The common scenario is JAVA or .NET toolset going to existing business logic in another language. An additional toolkit called Apache Axis is used in the JAVA scenario to speed web services development by hiding the SOAP message coding. In any case, a web application environment is built to be able to handle HTTP requests/responses using SOAP messaging.

The web service will typically be a wrapper around existing business logic in another language. For example, in Oracle Utilities Work and Asset Management there is an existing JAVA framework that can be leveraged, along with the Axis API, to support web services development. Oracle Utilities Work and Asset Management framework has built a generator to take an existing Oracle PL/SQL stored procedure and generate a java web service and the corresponding WSDL information from that. Web service and WSDL can then be deployed to java application server.

## Document and RPC Style

In general Web Services supports two different styles of structuring the SOAP messages: Document and RPC style.

From an external perspective RPC style and Document Style development is very similar. In fact, using today's toolkits (.NET and Apache Axis for example) the client developer can be abstracted away from this concept. Tools can examine the WSDL document of the web service and generate server and client side stubs. Developers can use what is generated to: 1) setup the request using simple getter/setters, 2) obtain the service interface from a locator object, 3) make the call to the service, and 4) obtain and manipulate the response with simple getter/setters. The developer doesn't even need to know that they are using SOAP or XML let alone which style of messaging they are using.

There is also a serialization/de-serialization step involved in the RPC call on both the client and server side. The objects (could be any language here) need to be translated into XML sent to the service provider. Subsequently, the service provider must de-serialize the request to objects it can understand. The same thing must happen in the other direction for the response. In document style, the XML is passed directly to/from the service provider. What is inside the <soap:body> element must be defined by whatever is in the WSDL "types" section.

RPC style web services will be used for integration points that are "function" oriented. An example of such a transaction would be ActivatePO in which a Client application would call the ActivatePO web service on Provider application passing a PO number as a parameter. The XML passed to the service provider conforms to the standard SOAP RPC conventions (section 7 of the SOAP 1.1 specification). The structure of the <soap:body> must contain just one element that is named after the operation - all parameters are expressed as sub-elements.

Document style web services would be used for more data rich integration points such as CreatePO in which the Client application would call the CreatePO web service on

Application B passing a PO XML document that contained header and line data for a PO.

### When to use which style?

If you are starting from existing code (java, C#, etc.) that you just want to expose via web services (like existing CORBA methods, EJBs, etc) then RPC style is a natural fit. Everything is already there for you. If you are starting from preexisting schema documents that you want to support through web services, SOAP encoding will just get in your way. Document Style would be best here.

One thing to consider, the Web Services Interoperability organization (WS-I) does not support SOAP encoding and banned its use in their Basic Profile of SOAP 1.1. It doesn't ban RPC style messaging - just RPC/encoded style. However a lot of development has gone into the interoperability of this style - many toolkits are available (.NET and Axis for example) for one to use. The RPC style still remains a convenient way to expose existing business logic as web services.

## Design Element – Client

A web service client is a program that can build up a SOAP message request using the WSDL information to properly formatting it, call the appropriate provider web service via HTTP, and process the response if necessary from the provider SOAP messages.

Developing a web services Client requires the following:

1. The ability to build requests based on WSDL information to call a web service
2. The ability to use HTTP and SOAP messaging to call a web service
3. The ability to consume a SOAP message response via HTTP from the web service

The Client application developer needs to know the URL address of the WSDL of the provider machine and the web service and operation name to call. It also needs to know the details of how to map application specify fields to the web services parameters just as in normal interface system development. The WSDL should give some documentation information that is readable to help understand describe the published web services but the developer will still need to do the mapping analysis between the client application and the web service.

The Client application developer can manually inspect the WSDL to build the appropriate SOAP messaging needed or a toolkit can be used, such as Axis or .NET, to point to the provider WSDL to generate stubs and SOAP constructs to call the web service.

The basic development process would be:

1. Find and inspect the appropriate WSDL describing the web service desired
2. Do the mapping analysis between client application information and web service parameters
3. Build a program to based on the WSDL schema and service information that can construct the SOAP message required by the web service, call the web service via HTTP, and handle the web service response if any.

## Security

Application tier security is handled separately for each application with a username and password required to access web services. For example, if the application group is implementing using Apache Axis for web services implementation, there is a setting to require a username/password be given to use the web service.

For database tier security, most implementations have a standard username/password for application tier access to the database that can then be affected by database security.

## Architecture Overview

Web Services Architecture is an interface specification that is based on open standards, XML messaging, and web technologies. A web service is a self-describing, self-contained, modular unit of application logic that exposes some business functionality to other applications through an internet connection. Applications access web services via ubiquitous web protocols and data formats, such as HTTP and XML, with no need to worry about how each web service is implemented. Web services can be mixed and matched with other web services to execute a larger workflow or business transaction.

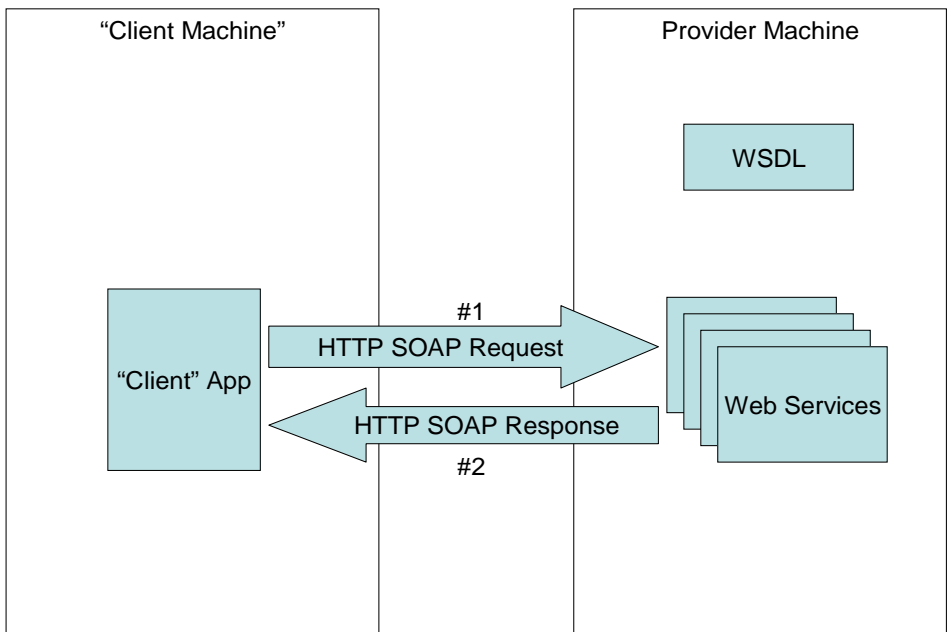
### Web Services Provider Interface

A web service provider interface consists of an HTTP protocol handler written to process SOAP formatted messages and a web services definition file (WSDL) that defines the web services available with the associated parameters and formatting information to call the web service. The WSDL can be generated or built manually from the web service definition when the interface is designed.

### Web Services Client Interface

A web service “client” interface consists of an HTTP protocol handler written to inquire, at runtime, the target system’s WSDL for service information in order to build the proper SOAP request for the service being called. The WSDL information can be cached on the client machine after the first call or pushed to the client machine in some other way as long as the client application can access it dynamically to know how to build the request to the web services. Caching is not handled by web services standards so individual application frameworks would have to handle this. Otherwise manual methods would need to be put in place to push changes to clients.

The following diagram shows the process flow at runtime:



1. Based on the WSDL information built into the Client application at design time, the client application builds a SOAP request in the proper format and makes an HTTP call to the web service.
2. The Provider web service unpacks the SOAP message and executes the service using the information in the SOAP message. It then responds, if appropriate, with a SOAP response with the results of the execution.

## Exposing Business Logic via Web Services

Oracle Utilities Work and Asset Management business logic in Oracle stored procedures can be exposed via web services. In some cases these web services can be generated from existing stored procedures. In other cases the stored procedures must be created first. Once the web service is deployed, the client application can then request the WSDL file via http by calling the web service with the "?WSDL" parameter. This can be done at runtime or design time.

It is the responsibility of the client application to properly call the Oracle Utilities Work and Asset Management web service based on the WSDL provided. It is also the client application's responsibility to "store" the web service URL (and other information) regarding the location of the web service.

Web services are built by pointing the `plsql2webservices` generator at a stored procedure or package to generate. The generator builds a java source file (that extends Oracle Utilities Work and Asset Management framework classes) to create a deployable servlet and a WSDL that is the web service.

## Consuming Web Services Provided by Other Applications

When a given event occurs in Oracle Utilities Work and Asset Management that requires calling a foreign application's web service, a piece of java code is written/executed. This java code consumes the WSDL file provided by the web service

provider. Data is gathered and passed to the web service as described in the WSDL. The response, if any, is then processed in the java code.

Sample prototype client code is shown below:

```
e.printStackTrace();
    }
}

package synergen.webservice;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import javax.xml.rpc.ParameterMode;
import javax.xml.namespace.QName;
import java.util.*;

public class TestClient2
{
    public static void main(String[] args) {
        try
        {
            // Setup
            String endpoint =
"http://localhost/synergen/services/WorkOrder";
            String targetNamespace = "WorkOrder";
            String targetOperation = "SdbpGisCreateWo";

            // Define qnames
            QName serviceName = new QName(targetNamespace,
"WorkOrder");
            QName portName = new QName(targetNamespace,
"WorkOrder");
            QName operationName = new QName(targetNamespace,
targetOperation);

            // Create service
            Service service = new Service();
            Call call = (Call) service.createCall();
            call.setPortTypeName(portName);
            call.setOperationName(operationName);
            call.setTargetEndpointAddress(endpoint);

            // add parameters
            call.addParameter( "arg1", XMLType.XSD_STRING,
ParameterMode.IN );
            call.addParameter( "arg2", XMLType.XSD_STRING,
ParameterMode.IN );
            call.addParameter( "arg3", XMLType.XSD_STRING,
ParameterMode.IN );
            call.addParameter( "arg4", XMLType.XSD_STRING,
ParameterMode.IN );
            call.addParameter( "arg5", XMLType.XSD_STRING,
ParameterMode.INOUT );
```

```
        call.addParameter( "arg6", XMLType.XSD_STRING,
ParameterMode.INOUT );
        call.addParameter( "arg7", XMLType.XSD_STRING,
ParameterMode.INOUT );
        call.setReturnType( XMLType.XSD_STRING );

// Out parameters
String sError=null;
String sErrorMsg = null;
String sWO = null;

// Invoke the Webservice
String result = (String) call.invoke( new Object[] {
"01", "Web Service Test", "M-1520", "Rob", sWO, sError,
sErrorMsg } );

// Print results
System.out.println("result : " + result);
// Example get Parameters
Map outparams = call.getOutputParams();

// Example Get Array of Parameter Values
Collection sel = outparams.values();
Object obj1[] = sel.toArray();

sWO = (String)obj1[0];
System.out.println("sWO : " + sWO);
}
catch (Exception e)
{
```



# **Oracle® Utilities Work and Asset Management**

Mobile Guide

Release 1.9.0.3

July 2011

Oracle® Utilities Work and Asset Management Mobile Guide for Release 1.9.0.3

Doc v1 rev.0 7/12/11

Copyright ©1986, 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: TUGBU Documentation

Contributing Author: n/a

Contributor: n/a

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any lost, costs, or damages incurred due to your access to or use of third party content, products or services.

---

---

# Contents

<b>What's New .....</b>	<b>4</b>
New Features for Release 1.9.0.3 .....	4
<b>Basic Navigation and Functionality .....</b>	<b>5</b>
About the Mobile Application .....	5
Process Overview .....	6
Downloading and Uploading .....	7
Using the PDA Interface .....	9
Viewing Records .....	11
Creating New Records .....	12
<b>Work Orders &amp; Service Requests .....</b>	<b>13</b>
Creating Work Orders .....	13
Creating Service Requests .....	14
Views .....	14
Actions .....	17
<b>PM Routes .....</b>	<b>21</b>
Viewing and Updating PM Routes on the PDA .....	21
<b>Appendix A: Using Mobile with CC&amp;B Integration .....</b>	<b>24</b>
Service Request Record .....	24
Meter Information View .....	25
<b>Appendix B: "Classic" PDA Setup .....</b>	<b>27</b>
Basic Installation Overview .....	27
Installing Mobile on the PDA .....	27
Configuring the PDA .....	28
<b>Appendix C: PDA Setup .....</b>	<b>29</b>
Installation Guidelines .....	29
Configuring the PDA .....	33
<b>Appendix D: Requirements and Recommendations .....</b>	<b>34</b>
Mobile Device .....	34
Application Server Software .....	34
Wireless Access Point/Router .....	34
Recommendations .....	35
<b>Appendix E: Web Services Installation .....</b>	<b>36</b>
Pre-Installation Guidelines .....	36
Additional Information .....	45
<b>Appendix F: Uninstalling Mobile from the PDA .....</b>	<b>49</b>

---

---

# What's New

This section describes some of the major feature and functionality changes that were implemented for this release.

## **New Features for Release 1.9.0.3**

Changes in these releases include the following:

- Updated installation procedure for newer, higher resolution PDAs.
- Certified on Windows Mobile 6.x

### **Related Topics**

[Online Help](#)

[Printed/Electronic User Guides](#)

[Basic Navigation](#)

[Subsystems and Modules](#)

Generated: July 12, 2011 9:55 am for Product Release 1.9.0.3.

## Basic Navigation and Functionality

Oracle Utilities Work and Asset Management Mobile allows users to enter and reference work related information remotely using a Personal Digital Assistant (PDA). The process begins when you download information from the main Oracle Utilities Work and Asset Management database to the Mobile application on your PDA. You can then view records and enter transactions related to the activities you are performing. When you finish, upload the transactions from the PDA to the main application, where they are processed by interface logic to update or create the appropriate Oracle Utilities Work and Asset Management records.

### About the Mobile Application

The Oracle Utilities Work and Asset Management Mobile PDA system is basically a stand-alone or a store & forward system. The mobile devices are not connected to the database except when they are docked. At docking, the device is synchronized with the database via the OracleMobileService web service hosted by IIS.

### Web Service Connection to Database

The OracleMobileService web service makes it's own connection to the database based on information in the web.config file using a hard coded database username and password: SYNERGEN\_MOBILE/SYNERGEN\_MOBILE.

The SYNERGEN\_MOBILE and MOBILE database role are normally created as part of the base product installation. A script is available to run when the SYNERGEN\_MOBILE user is missing. If this circumstance occurs, please contact support.

Installation of Oracle Utilities Work and Asset Management software on PDA and IIS are documented in Appendixes C and D.

### Mobile PDA Connection to the Web Service

The OracleMobileService web service utilizes “anonymous access” as its authentication method. An IUSER\_ComputerName account is needed on the same server as the IIS/web service. This account name is for the operating system authentication, not for database authentication. It is set up automatically by IIS/ASP.net at installation. Normally, “Anonymous Authentication” will be enabled by default for the OracleMobileService web site by the default IIS installation.

## Enable Anonymous Authentication

The following steps may need to be completed if a different version of IIS is used instead the recommended default (currently, IIS 6).

These steps refer to the process in IIS 6. The procedure may vary slightly in other versions.

1. Access the web site representing the OracleMobileService web service.
2. Right click the icon and click Properties in the menu.
3. Select the “Directory Security” tab in the resulting dialog box.
4. Click the “Edit” button in the “Authentication and access control section”.
5. Open the Authentication Methods dialog box within the Microsoft IIS Manager utility.

Near the top this dialog box, there should be an “Enable anonymous access” check box. This must be checked to enable Anonymous Access. In IIS version 6, this “Authentication Methods” dialog box can be reached by taking the following steps:

## Download to Mobile

Connect the PDA to your company intranet. Open the Oracle Utilities Work and Asset Management Mobile application and synchronize. On the Download tab, enter the selection criteria for the data you want to download and tap the Download button. The records are copied to your PDA.

*Note:* If you receive an error message while downloading or uploading data, please make a note of the message and inform your system administrator.

## Work with Mobile

- Create work orders and manage materials, labor, and service history
- Create service requests, enter meter information and access customer billing information
- Reference PM Routes and update the status of route stops as they are completed
- Charge time for completed work
- Record stock issued and other charges from your mobile storeroom
- Update task progress or service request closeout
- Enter notes for service requests and work orders

## Upload to the Database

Connect the PDA to your company intranet. Open the Oracle Utilities Work and Asset Management Mobile application and synchronize. The number of transactions on the device to be uploaded displays on the Upload tab. When you tap the Upload button, all transactions you performed on the PDA are uploaded and batch processing performs the necessary updates in the Oracle Utilities Work and Asset Management database.

## Downloading and Uploading

Before you can access information remotely on the PDA, you must download records from the Oracle Utilities Work and Asset Management database. This insures that you have the most current information and allows you to select the records appropriate for

the work planned for the day. Later you can upload the transactions that you recorded on the PDA for inclusion in the main database. Only after your transactions are uploaded can the system create new records, post time and material charges, and update task progress records to document your mobile activities.

Before using Oracle Utilities Work and Asset Management Mobile for the first time, you or your system administrator must configure your PDA for Oracle Utilities Work and Asset Management Mobility.

For information on configuring the PDA for Oracle Utilities Work and Asset Management Mobility, please see Appendix B.

For information on configuring wireless connections, please see the owner's manual for your PDA.

### Related Topics

[Using Upload and Download](#)

## Using Upload and Download

[How to Download to the PDA](#)

[How to Upload Data to Oracle Utilities Work and Asset Management](#)

### How to Download to the PDA

#### 1. Connect your PDA to your company intranet.

While you may be able to connect to your network by docking the PDA in the cradle and using a pass-through connection through your PC, this type of connection is not always stable and is not recommended by most PDA makers.

#### 2. Open the Oracle Mobile application on the PDA.

#### 3. Select Admin from the menu.

#### 4. Select Synchronize.

#### 5. Select Download and choose which set of data you want to download.

If you have any files to upload from the device, you should complete the upload first before conducting a download.

You must conduct a download for each category individually. For example, if you want both Service Requests and PM Routes, you have to complete this process for Service Requests then complete it again for PM Routes. The order in which you download the information is not relevant.

**PM Routes** - Downloads active PM Routes and related route stop data.

**Service Requests** - Downloads active service requests and related data for the schedule date and crew.

**Work Orders** - Downloads active work orders and related data for the crew or employee selected. The Employee filter limits the download to records where the entered Employee ID is assigned to an active or finished Work Order Task in the Assignments view (not the Assigned To field). If a Crew filter is also applied, the Employees must also be on the specified crew.

**Lookup Data** - Downloads the background information needed to run the application on the pda. You should select this option the first time you use the PDA interface.

Note: Please refer to [Appendix E: Web Services Installation](#) guide for information on how to avoid Out of Memory errors when downloading lookup data.

Caution: If you have already downloaded data, choosing this option will clear the pda and of all data including the PM Routes, Work Orders, and Service Requests.

Each time you download Lookup Data, you will also need to download the additional records as well. However, it IS recommended that you complete this download periodically to pick up any information that has recently been created in the main application.

The system copies records, filtered by the selected fields. This includes Asset ID and asset description of active assets that are not V(ehicle), P(rocess) or F(unction) record types, customer information including addresses, employee data, vendors, settings from business rules associated to work order and service request processing, stock codes and descriptions of active items from the designated storeroom and additional background information. You can select up to three Asset codes to help filter the number of records downloaded to the device. The Asset Codes fields located on the Asset record in the main application can be used to establish the appropriate code filters.

**6. As necessary, select filter criteria to determine which information is downloaded to the pda.**

If you need to change the values displayed, tap the Refresh Lists button and then select from the dropdown lists.

Leaving a field blank signifies that you want all data from that field. Wild cards cannot be used.

**7. Tap the Start button to initiate the download.**

The system displays a confirmation message when the download is complete.

**8. Continue downloading record types as necessary until you have all of the data that you need loaded to the pda.**

The option to download ALL is not available because the downloads can include date fields. It is important that you verify that the entered date ranges are valid before you conduct the download.

### How to Upload Data to Oracle Utilities Work and Asset Management

**1. Connect your PDA to your company intranet.**

**2. Open the Mobile application on the PDA.**

**3. Select Admin from the menu.**

**4. Select Synchronize.**

**5. Select Upload from the menu.**

The number of transactions that are currently stored on the device and are ready for upload is displayed.

**6. Tap the Upload button.**

All transactions stored on the PDA are uploaded to the Oracle Utilities Work and Asset Management database, where batch processing will update or create the necessary records.

A confirmation message is displayed on the PDA when the upload is complete.

## Using the PDA Interface

Select from menus at the bottom of the PDA screen to navigate to the screen you want to use. The menu options change depending on which screen you have open. The primary menu selections include:

**Open** - Use this option to open Service Request or Work Order. After you open the record you can use views and actions to record time, issue and return stock, or conduct other common tasks.



**Create** - Opens screens where you can create a Work Order or Service Request.

**Admin** - Select this option to synchronize the PDA with Oracle Utilities Work and Asset Management.

Once you open a Service Request or Work Order, you have the following options:

**Navigate** - Shows the screens you currently have open or can access.

**View** - Additional information associated with the record you are viewing.

**Action** - Select Action to have access to common tasks from the record such as entering crew time, task progress, closeout information, stock charges, or other charges.

## Closing Windows

When you have finished viewing a window, you can close it by tapping the OK button in the upper right corner of the open window. The window closes and the previous window is displayed.

## Viewing Text Popups

Tap any text field to open a larger popup window containing the text in that field. This window can make it easier to read text that may be too small or incomplete in the main PDA window.

Close the popup by tapping OK.

## Using the Scroll Bars

When more information than can fit on the screen needs to display in one window, as is often the case with Search Results and some of the other windows, scroll bars appear to provide access to the rest of the information.

You can move through the display to see all of the information by tapping the arrows at the end of the scroll bars, or anywhere on the scroll bar itself. Tap the scroll bar slider with the stylus and drag it up and down, left or right in the same way you would with the mouse pointer.

## Resizing Columns

To make a column wider or narrower, hold the stylus on the divider between the columns you want to resize and drag the column divider right or left.

## Using Drop-Down Lists

Some fields have an associated lists of values that you can use to select data. These fields are identified by a down arrow at the right edge of the field. On many screens, you must first populate the list by tapping a button such as “Refresh Lists.”

## Date Fields

Date fields appear with an asterisk button next to them. Tap the asterisk to enter the current date. Otherwise you can use the transcriber or the calendar tool to enter or select a date.

## Searching for Field Values

Several screens include a Find feature which allows you to enter search criteria to locate values for individual fields. In the example below we will find an Asset ID to populate a new work order. Use a similar process to find other field values.

### How to Search for an Asset

Select Create then Work Order from the main menu to complete this activity. Tap the Find button.

The Find button for this example is on the Create Work Order screen.

1. **Select Asset ID from the popup menu.**

A new screen opens where you can enter your search criteria.

2. **Enter your search criteria.**

You can enter a partial Asset ID (use a leading %), a partial description, or select from dropdown lists for department, area, and asset codes.

If asset codes were used when downloading data, those codes appear in the asset code fields as greyed-out text. You cannot change these values, but you can add additional asset codes if not all the code fields have been used.

To filter by an additional asset code, tap the down arrow for the appropriate empty asset code field and select a value from the drop down list.

3. **Tap the Search button.**

4. **Tap the appropriate Asset ID.**

Use the scroll bars to locate the appropriate asset in the results list.

If an Asset Code field does not contain a value, you can select from the list of values to add an additional filter to your search.

5. **Tap Yes to accept the asset you selected.**

The Create Work Order screen opens with the Asset ID field populated with the asset you selected.

## Viewing Records

Finding and viewing records on the PDA involves conducting a search then selecting from a results of search.

### How to View a Record

1. **Select Open from the main screen.**

From here you can select the type of record to open and a search screen opens allowing you to select criteria to find the records you need.

2. **Enter or Search on a specific Record ID, or enter other criteria such as Status, Type, or other record defining details.**

Once you Enter Criteria and tap the Search button, a results screen opens showing the record IDs and descriptions of all the records matching your search.

3. **Tap the record ID to open the record.**

From the Search, Results of Search, or the record screen select Navigate from the menu to open one of the other screens.

## Creating New Records

The Mobile application currently includes the ability to create new work orders and service requests on the device.

When these records are created, the system assigns a temporary record ID preceded by an R or an S respectively to indicate that the new record was created from the PDA (as opposed to having been downloaded from the main database). Once the new records are uploaded to the database the system verifies sequence numbering and adds prefixes to the record ID as defined in your system settings.

Remember that these new records have been created on the PDA only. You can view and record charges against them, but they are not available to other Oracle Utilities Work and Asset Management users until you synchronize your mobile transactions with the main database.

## Chapter 2

---

# Work Orders & Service Requests

The Oracle Utilities Work and Asset Management Mobile records accessed on your mobile device contain information similar to that contained in the related module headers Oracle Utilities Work and Asset Management application. In addition to the record ID and description the records also include status, type, and other information relevant to the record. Please refer to the Oracle Utilities Work and Asset Management Users Guide for a complete description of fields.

In addition to the information on the record windows, you can select from the View menu to see additional information about the work order task or service request.

## Creating Work Orders

Work Orders that you download to your PDA have been preplanned and approved. However, if you discover a required repair in the field, you can create a work order in Oracle Utilities Work and Asset Management Mobile for the repair and even issue parts and record time against the new work order. When you return from the field and synchronize your PDA with Oracle Utilities Work and Asset Management, the new work order and associated charges will be copied to the main application.

### How to Create a Work Order

1. **Select Create then Work Order from the menu.**
2. **Enter a description.**  
Use the touch pad or transcriber to enter a brief description of the work required.
3. **Enter a required date and select a priority.**
4. **Select the Asset ID.**  
When you select the Asset ID from the drop-down list, the asset description is automatically populated. You can also select Find from the menu to search for Asset IDs.
5. **Select a failure code and enter deficiency tag, class and category if required.**
6. **Tap the Save button.**  
The system confirms that the work order was created successfully and displays the Work Order number.

Now you can record charges against the work order if needed.

## Creating Service Requests

Service requests downloaded to the PDA have been already been entered in the system and are provided for your reference. However, if you discover the need to create a service request in the field, you can do so directly on the pda and later upload the record to the main application.

### How to Create a Service Request

1. **Select Create then Service Request from the main menu.**

The Create Service Request screen opens.

2. **Enter or select a Type, Customer ID, Problem Code and description as applicable.**

If you select a Customer ID the system populates the address and contact information from the customer record. This information cannot be manually updated on this screen. After the new record is created, open the Customer Billing view to update address information. Any modifications made to the customer info only apply to the newly created service request.

Only customers in active status are represented in the list.

You can create a new service request without entering the customer ID. Enter as much address information as possible. Only the problem code is required.

3. **Enter a additional information as needed.**

The Schedule Date and Service Point ID (SP ID) are only used when Oracle Utilities Work and Asset Management is integrated with CC&B.

4. **Tap the Save button.**

The system confirms that the service request was created successfully and displays the record number.

New records are created and uploaded to the application in ACTIVE status.

## Views

Views that are most commonly used or are useful out in the field are included on the pda. Notes and Service History appear in both modules. Work Order also includes Material, Labor and Asset Data. Service Request includes Customer Billing Info, Meter Info and Closeout.

### Common Views

Notes and Service History are available for both Service Request and Work Order transactions

[Notes](#)

[Service History](#)

#### Notes

The Notes view shows any standard notes that have been attached to the work order task. You cannot create note attachments on the PDA, but you can view notes that have been created in the Oracle Utilities Work and Asset Management application. Tap the Description field in the Notes view to read the note.

#### Service History

Use the Service History view to review and maintain a maintenance record for the asset listed on the Work Order Task or Service Request. The upper portion of the screen contains a list of the type and category of specifications assigned to the asset. The lower portion shows the attributes for the type and category selected above, and the values that have been entered previously.

After doing work on the asset or conducting an inspection, you can update the Service History by entering new values for the appropriate attributes.

You cannot add new service history on the pda, only update of existing values is allowed.

### How to Update the Service History

1. **Open the appropriate record.**
2. **Select Service History from the View menu.**
3. **Select a type and category.**  
When you tap the type, the bottom portion of the screen changes to show the attributes that Type and Category.
4. **Tap the Value button for the attribute you want to update.**
5. **Enter the new value.**
6. **Tap the Save button.**

### How to Restore a Service History Value

If you make an error while updating a Service History value, you can undo your error by tapping the Restore Original Value button. The Restore Original Value button is only available after a value has been entered for the attribute. The originally downloaded value is restored to the field.

## Work Order Views

The following views are only available for Work Order transactions on the pda.

[Work Order Material](#)  
[Work Order Labor](#)  
[Work Order Asset Data](#)

### Work Order Material

The display only Material view shows a listing of items required to complete the work order task. Both inventory and direct purchase items that have been planned against the work order task are included. Items are identified by stock code and the issuing storeroom. The quantity planned is listed in the Estimated column. The quantity checked out of the storeroom may or may not be listed in the Actual column, depending on when you downloaded information to your PDA and when the stock was issued. If you need to issue additional material to complete work on a task, you do not record that information here. Rather, you can use the Stock Charges action to issue the parts from your mobile storeroom.

### Work Order Labor

The display only labor view shows the labor needs associated with the work order task, including the number of people needed from each craft and the total number of hours. Oracle Utilities Work and Asset Management enters the actual regular and premium labor time when timesheets are posted by batch processing. Although you cannot enter time directly on this view, you can use the Crew Time action to record time for crew members against the work order task.

### Work Order Asset Data

The Asset Data view shows location information for the asset. Depending on the type of asset, this could be a location within a building, a street address, or some other location.

If the Asset record has additional descriptive information, such as size, length or capacity, you can view these details in the Specification screen. Tap the asterisk button

to the right of the Specification field to open a screen containing specification details for the asset.

In Oracle Utilities Work and Asset Management, specifications are used to record different types of information. Asset specifications describe static features of the asset, while service history specifications describe repairs, tests, and other actions taken. Asset specifications are maintained in the main application and cannot be updated on the PDA.

#### How to View Specification Details

1. **Open the appropriate work order task record.**
2. **Select Asset Data from the View menu.**
3. **Tap the asterisk button to the right of the Specification field.**  
The Specification screen opens containing the specification details for the asset.
4. **Tap any attribute in the lower section of the screen.**  
A text popup opens containing the Attribute description and value.
5. **Tap the OK button to close the popup.**
6. **Tap the OK button to close the Specification screen.**

## Service Request Views

The following views are only available for Service REquest transactions on the pda.

[Service Request Customer Billing](#)  
[Closeout View](#)

### Service Request Customer Billing

Select Customer Billing from the Service Request view menu to review and update customer information related to the service request, as needed.

All fields except for the Customer ID are updatable. Please note that the fields represented refer to the customer only, not the problem or the reported by. Any updates made to the customer information only apply to the current service request. If you were to update the billing information then create another new service request with the same customer ID the original downloaded customer information would be applied to that new service request.

### Closeout View

Select Closeout from the Service Request view menu to enter closeout information. Select values for the fields as appropriate for your business practices.

## Common Actions

Commonly used actions such as Crew Time, Stock Charges, and Other Charges, are included in the PDA application for both work orders and service requests.

[Recording Crew Time](#)  
[Recording Stock Charges](#)  
[Recording Other Charges](#)

### Recording Crew Time

After individuals or crews complete work, you can record time for the entire crew or for selected members of the crew.

### How to Record Crew Time

1. **Open the appropriate record.**
2. **Select Crew Time from the Action menu.**  
The Crew Time screen opens.
3. **Verify the date.**  
If it is necessary to change the date, tap the down arrow to open the calendar tool. Then select the correct date.
4. **Select the crew or an employee.**  
The Crew Time screen opens with the default crew name in the Employee field. If you are recording time for the entire crew, leave this selection unchanged.  
  
If you are recording time for an individual, select the appropriate the crew member from the drop-down list.
5. **Enter regular or premium time as appropriate.**  
Select the Shift, Type, and the number of hours from the drop-down lists.
6. **Double check all fields.**
7. **Tap the Add button.**  
The application records and displays the time you have entered.  
  
You can continue to add time or you can tap the OK button to return to the work order task.

*Note:* **Note:** If you make an error you can select Clear Selected or Clear All from the List menu to remove a selected line or all lines from the times already added.

### Recording Stock Charges

As you complete work you may need to use inventory stock from your mobile storeroom to complete your tasks. Use the process described below to record stock charges. Do not use this process to record planned material items checked out from a central storeroom. Those charges are tracked in Oracle Utilities Work and Asset Management through the Inventory subsystem and can be seen, but not modified, in the Material view in Oracle Utilities Work and Asset Management Mobile.

When stock charges are uploaded from the mobile application, entries are created in the Inventory Log and an issue ticket. These entries can be viewed in the Stock Checkout module of the main application.

### How to Record Stock Charges

1. **Open the appropriate work order task.**
2. **Select Stock Charges from the Action menu.**  
The Stock Charges screen opens showing any stock codes already charged to the work order task.
3. **Select the stock code for the item you want to add.**  
Remember, the drop-down list only includes stock items in the storeroom that was downloaded to the device.  
  
You can also select Find from the menu to search for Stock Codes by Stock Code and Stock Description.
4. **Enter a quantity for the item.**  
Select a quantity from the list or enter a number from the touch pad or transcriber.
5. **Tap the Add button.**  
The application adds the quantity to the list of items charges to the work order task.



You can repeat steps 3-5 to continue adding items. Tap the OK button to return to the work order task when you are finished.

If necessary you can select List then a Clear option to delete any of the stock charges.

## Recording Other Charges

You can use the Other Charges view to record charges other than parts and labor incurred while completing work. For example, you may need to rent equipment for a Work Order Task and then relate the rental cost to the Work Order.

### How to Record Other Charges

1. **Open the appropriate record.**
2. **Select Other Charges from the Action menu.**

The Other Charges screen opens.

3. **Select the Charge Type from the drop-down list.**

The list shows the charge types used by your organization, such as mileage, lodging, or meals.

When you select the charge type, the related units and price are automatically entered. You can change these values if needed.

4. **Select or enter the Quantity to charge.**  
When you enter the quantity, the system calculates the total.
5. **Enter a Reference Number and Vendor Code if required.**  
You can select Find from the menu to search for vendor codes.

6. **Tap the Add button.**

The charge is displayed in the upper section of the screen.

You can repeat steps 3-5 until you are finished adding other charges, or you can tap the OK button to return to the originating record.

## Work Order Actions

Task Progress is an additional action available for work orders.

### Recording Task Progress

You can record your progress toward completing the work order task on the Task Progress screen. Unlike the main application, you can enter task progress information even if you do not plan to set the record status to Finished. You can record progress toward completion as often as needed until the task is complete.

### How to Record Task Progress

1. **Open the appropriate work order task record.**
2. **Select Task Progress from the Action menu.**
3. **Change the Status if appropriate.**

Change the status to Finished if you want to finish the work order task. You can still record progress on a work order task in Finished status.

4. **Change the following as needed.**

**Phase** - As work progresses, updating the phase can help planners track the work process.

**Work Done** - Use these three fields to record the work done for this task. Select from the drop-down lists. An example might be: “3 quarts oil replaced”. When the pda information is uploaded to the main application, the system creates an Activity Tracking record when the Work Done fields are completed on this screen.

**Comments** - Use the touch pad or transcriber to add a description of the work done if needed to describe the task progress.

**Failure** - Select a failure code from the drop-down list to record the type of problem that required the work to be done. An example might be: “Hydraulic” for a leak in a hydraulic system.

**Mode** - Select or enter a failure mode if necessary. The use of this field varies from organization to organization.

**Component** - If the work was done on a component of the asset, select from the list to identify the component.

**Repair** - Select a repair code from the drop-down list to record the type of work done. An example might be: “Resealed” for replacement of the seals in the hydraulic system mentioned above.

**Next Action** - Select the appropriate code from the drop-down list if further action needed - for example an inspection after some period of operation.

5. **Tap the Save button when you are finished entering data.**

You can return to the Task Progress screen to enter additional information or update the status as work continues on the task.



---

# Chapter 23

## PM Routes

The Oracle Utilities Work and Asset Management Mobile application is designed to manage PM Routes so that you can access route stop information from remote locations and record completed work on the routes as it is done.

### Viewing and Updating PM Routes on the PDA

The fields on the PDA are the same as those that can be found on the header in the main application. None of the fields can be edited.

#### How to View and Update PM Routes on the PDA

1. **Open the PM Routes mobile module and search for the desired record.**

The lower section of the screen shows the total number of stops on the route and their respective statuses. Stops To-Do are in Created Status.

None of these fields can be edited.

2. **Select Route Stop (List) from the View menu to see details on the route stops included on the route.**

In this example, PM Route No. ILB001 has two stops both in Created status.

3. **Tap the status field to change the status from Created to Finished or Missed.**

There is no restriction on changing the status. You can go from Finished to Created or Missed to Finished, etc., as needed. Tap Save after making changes.

4. **Select a stop to see the full details.**

You can also change the status on this screen and enter comments.

Tap Save after making changes.

You can view even more information by selecting More Details from the View menu. This screen provides information about the asset indicated on the stop.

Return to the list of stops screen and select Update all stops as Finished from the Action menu if you want to change the status for all stops at one time.

It is important that you review all stops before selecting this Action.

Once you have completed work on the route stops and have changed statuses as appropriate, upload the information back to the main Oracle Utilities Work and Asset Management application.



---

# Appendix A: Using Mobile with CC&B Integration

The Oracle Utilities Work and Asset Management mobile application includes functionality to capture badge numbers and meter readings, and associate them with service points. When this CC&B integrated functionality is activated, special data entry requirements must be met so that the information entered on the pda is correctly uploaded to Oracle Utilities Work and Asset Management and transferred to CC&B without errors. To use this functionality the Service Request Status Trigger rule key in the Product Integration business rule must be set to ON. Additional rule key settings are also required. Please refer to the WAM - CCB Integration Guide for more information.

## Service Request Record

Existing service records that have been downloaded to the pda include a Service Point ID (SP ID) and description based on information from the Asset record in the main application.

The record type for SP IDs is not shown on the pda, but will be represented in the main application when information is uploaded. Record types for service points are defined in the

## Deleting Service Requests

Service Requests on the pda cannot be deleted. However, you can select Closeout from the View menu to cancel a service request.

Even if a service request is canceled, it is still updated to Oracle Utilities Work and Asset Management so that crew time, stock charges or direct charges can be applied against the canceled record if necessary.

**Note:** Canceling a service request on the PDA will also cancel the service request in Oracle Utilities Work and Asset Management. However, Oracle Utilities Work and Asset Management will NOT transmit the canceled service request to CC&B.

### How to Create a Service Request

1. **Create a new service request as described for standard processing.**
2. **Enter a Schedule Date and SP ID.**

When CC&B integrated processing is enabled the Schedule Date and Service Point ID (SP ID) are required on new service requests.

The drop-list for SP ID includes all assets where the CCB\_SOURCE field on the Asset record is set to "SERVICE POINT."

Regular assets are distinguished by the Asset Record type, Asset ID combination. The Service Point ID is equivalent to an Asset ID, but the record type is not assigned until the

record is uploaded. The system assigns the record type based on the setting for the SP Asset Record Type rule key in the Product Integration CCB business rule.

### 3. Tap the **Save** button.

The system confirms the successful creation of the request and displays the record number. New records are created and uploaded to the application in ACTIVE status. The system uses the department, area, and account number corresponding to the service point ID as configured in the Default Accounts for Interfaces business rule.

## Meter Information View

Multiple badge numbers (meters) can be entered on one service request. Select Meter Info from the View menu to enter readings or to add additional meters. This can be useful if you need to swap the badge number of an old meter for a new one being installed.

This can also be used to insert a meter badge number for cases where the meter badge number was not validated on the service request before downloading to the PDA.

You can enter new badge numbers by selecting List on the Meter Info menu then Add.

Enter the badge number, configuration type, disconnect location and status. The system automatically enters the current date and time as the reading date but these values can be changed.

To update an existing meter tap a field on the grid in the Meter Info screen to open the same Meter Readings detail screen.

When updating you can update everything except the SR and Badge No. The existing reading is overwritten when the pda information is uploaded back to the main application.

In the lower portion of the screen you can enter or update readings on existing rows. The only value that can be modified is the reading.

Add new rows by tapping List then selecting Add.

The Service Request number and Badge number are brought over from the Meter Info of the badge number you are working on.

Enter UOM (unit of measure), TOU (time of use such as Peak or Off-Peak), Reading and a Sequence number.

The sequence number must be unique. These are used to order the readings. You may want to increment the sequence numbers by 5 or 10 in case you need to insert a reading in between existing readings.

The UOM TOU combination must be unique for the badge number even if the TOU is empty.

## Deleting Meter Info Records

Meter readings and badges can be deleted on the pda even if they were downloaded from the main application. These deletions will delete the readings in the Meter Information view of the Service Request module when the pda information is uploaded.

Tap the row header of the row to delete then select List and Clear Selected. You can also select Clear All without selecting a badge. The system confirms whether or not you want to complete the deletion.



---

## Appendix B: “Classic” PDA Setup

### For older pda devices with no phone and screen resolution of 240x320

This section provides instruction on how to install older touch screen mobile devices that do not include a phone. It is intended for devices with screen resolutions of 240 x 320 and has been used for devices running the following:

- Pocket PC 2003
- Windows Mobile 5
- some Windows Mobile 6 devices

This installation requires .NET Compact Framework version 1.1, which is automatically provided by the installation program for devices that are running .NET Compact Framework version 2.0. For newer (up to Windows Mobile 6.5) touch screen devices with higher screen resolutions and phones please see [Appendix C: PDA Setup](#).

## Basic Installation Overview

Install the Oracle Utilities Work and Asset Management Mobile application by completing the following:

1. Dock the PDA
2. Run the installer program from your computer
3. Configure the PDA to connect to the proper database
4. Download data

If you are currently on an older version of the mobile application (Synergen, SPL EAM, or Oracle Utilities Work and Asset Management prior version), it is best if you uninstall the older version before you install the upgrade. In some cases failure to remove previous versions before upgrading can result in the loss of data or performance problems. Please refer to [Appendix F: Uninstalling Mobile from the PDA](#) for more information.

## Installing Mobile on the PDA

To install the Oracle Utilities Work and Asset Management Mobile application on the PDA, you must dock the PDA in the cradle. You cannot install this software via the network connection.

Depending on the operating system on the device, the installer will install:

- .Net Compact Framework
- SQL Server CE 2.0
- Oracle Utilities Work and Asset Management Mobile Application

The installer detects which components are needed based on the version of the operating system.

### How to Install Mobile on the PDA

1. **Dock the PDA in the cradle.**
2. **Double-click the installation file “OracleMobile\_14151\_Install.exe”.**
3. **Click the Deploy button on the installer screen.**

The installation begins.

During the installation you may see additional messages guiding you to something on the PDA.

If you see a setup message warning that the installed program was designed for an earlier version of the Window Mobile software, you can safely ignore it. Microsoft advises that the warning appears only to alert users that the application they are about to use does not support screen rotation, a new feature in Windows Mobile 2003 Second Edition.

4. **Tap the OK button to clear the Setup message.**

The installer will show a success message, and the installation is complete.

Follow the steps in the “Configuring the PDA” section to connect to the proper database and download data for your Oracle Utilities Work and Asset Management Mobile work session.

## Configuring the PDA

After you install Oracle Utilities Work and Asset Management Mobile onto the PDA, you must configure the device to point to the proper application server.

### How to Configure the PDA

1. **Open the Oracle Utilities Work and Asset Management Mobile application on the PDA.**

Start the application by tapping on Start > Programs > Mobile.

When you start the application on the PDA for the first time, a message displays asking you to download data.

2. **Tap on the OK button to clear the message.**
3. **Select Admin from the menu then Configure.**
4. **In the Server Name field, enter the application server and folder where the Web Service was installed.**

The server name “192.168.31.69/prod” is shown in the example to the right.

Multiple instances of the Web Service can be installed in separate folders on the same Internet Information Services (IIS).

5. **When finished entering the Server Name, tap the Save button.**
6. **The PDA is now ready for downloading data.**

---

---

## Appendix C: PDA Setup

### **For devices with phones and screen resolutions of 480 x 800 or 480 x 640.**

This section provides instruction on how to install newer touch screen mobile devices that include a phone. It is intended for devices with screen resolutions of 480 x 800 and 480 x 640. This installation is for devices running the following:

- Windows Mobile 6.5
- some Windows Mobile 6 and 6.1 devices

This installation requires .NET Compact Framework version 3.5 on the device, which is provided via an additional CAB file installation if needed. For older touch screen devices with lower screen resolutions and no phone please see [Appendix B: "Classic" PDA Setup](#).

#### **Notes:**

- Please review this entire document before beginning installation.
- The Windows Phone 7 operating system is not currently supported.
- 480 x 640 resolution may require additional setup via configuration file settings.
- If an older version of the mobile application (Synergen, SPL EAM, or Oracle Utilities Work and Asset Management prior version) is currently installed, uninstall the older version before installing the upgrade. Please refer to [Appendix F: Uninstalling Mobile from the PDA](#) for instructions.

## **Installation Guidelines**

The "classic" version of Oracle Mobile was installed via a special installation program using an ActiveSync connection. Like the "classic" installation method, an Active Sync connection is required between a PC and a mobile device (ActiveSync has been superseded by Windows Mobile Device Center on newer Windows OS versions). However, unlike the classic installation method, an installation program is not used to install this version of the mobile application on the PDA. The newer version of Oracle Mobile is installed via one or more CAB files. This method of installation provides improved granularity of control over what is installed on the mobile device.

## **How to Install CAB Files**

CAB files are similar to self-extracting zip files. They often contain more than one file, they are compressed to save space, and they can be "run" to decompress them and copy them to a location set by the CAB file creator.

**To install a cab file**

1. Copy the file from the PC to any folder on to the mobile device.  
The File Explorer that can be opened from ActiveSync or WMDC on the PC will provide the means to do this.
2. Tap the CAB file on the mobile device to start the installation.  
Tapping the CAB file on the device causes the individual files within the CAB file to be decompressed automatically and copied to the correct folder (it does not matter what originating folder the CAB file is in). Some CAB files may also complete tasks such as creating registry entries and startup icons.
3. During CAB file installation, choose the best answer for all of the questions asked in the install wizard.  
If the install asks if it can put the files on a removable storage device, the answer should be "no".

After all questions have been answered and the CAB file install is complete, the CAB file is no longer needed and can be deleted from the mobile device.

**Determining which CAB Files to Install on the PDA**

During the install of the full Client Server WAM application, a directory will be created on the server containing the four CAB files needed to install the Oracle Mobile application on the PDA (see the "Oracle Utilities Work and Asset Management Upgrade Install Guide" for location). One of these CAB files installs the application itself and creates the program icon on the Start screen. One of them installs the .NET Compact Framework 3.5 files, which may not be installed on the target PDA but are needed by the mobile application. Another installs SQL Server CE, which is used by the application. The remaining CAB file installs a utility which can be used to manage and expose information about the .NET Compact Framework versions installed on the mobile device. Depending upon what software is already installed on the mobile device, between one and four CABs must be installed. Follow the instructions in subsequent sections to install the files.

Use the following information to locate the CAB files on the PDA and to determine which files to install:

- **OracleMobileDeviceinstaller.CAB** - This file must always be installed. It contains the Oracle Mobile application.
- **Microsoft\_.Net\_Configuration.CAB** - This file is recommended and sometimes required. It shows which versions of the .NET Ccompact Framework are installed. It is also used to ensure that the Oracle Mobile application runs using .NET CF 3.5 if multiple .NET CF versions are installed on the device.
- **NETCFv35.wm.armv4i.cab** - This file must be installed if the .NET Compact Framework version 3.5 is not already installed on the device. The Oracle Mobile application requires version 3.5 of the .NET CF to run.
- **sqlce.phone.wce5.armv4i.CAB** - This file must be installed if SQL Server CE is not already installed on the device. It installs SQL Server CE 3.5. Older versions of SQL Server CE which are already installed on the device may be sufficient for this mobile application. If an older version is already installed, skip installing this file. However, install this file to upgrade to SQL Server CE 3.5 if you experience problems.

## Installing the Mobile Application on the PDA

The installation process for installing the mobile application and supporting files consists of installing from one to four CABs. Instructions for installing the proper CAB files are provided below.

### Step 1: Connect to the PDA

1. Connect the PDA to the installation computer.  
Use a USB cable to make the connection. Do not try to install this software via a standard network connection.
2. Wait for ActiveSync to connect.  
After connecting, wait for ActiveSync on the device to connect with ActiveSync on the PC (or Windows Mobile Device Center if running Windows 7 on the PC).
3. Launch the file explorer from the PC's Active Sync or Windows Mobile Device Center application.
4. Navigate to the directory on the PC containing the installation CAB files.  
Please refer to the section titled [Determining which CAB Files to Install on the PDA](#) for the list of files and their location on the source PC.

### Step 2: Install Microsoft\_.Net\_Configuration.CAB on the mobile device

This procedure installs the .NET Configuration utility and adds an icon for it to the Start screen. This utility can be used to see which version(s) of the .NET Compact Framework are installed on the PDA. It is also used to associate the Oracle Mobile WAM application with the correct version of the .NET Compact Framework (i.e. version 3.5) if multiple versions of the .NET Compact Framework are installed on the mobile device.

1. Locate **Microsoft\_.Net\_Configuration.CAB** in the proper directory on the source computer.
2. Install this CAB file on the mobile device as described in [How to Install CAB Files](#).

### Step 3: Install NETCFv35.wm.armv4i.cab on the mobile device (if necessary)

This CAB, if installed, will add the .NET Compact Framework to the mobile device. However, some recent mobile devices will come with the .NET Compact Framework 3.5 pre-installed. To see if version 3.5 of the .NET compact framework is already present on the mobile device, run the .NET Configuration utility by tapping the icon on the start menu. When the program opens, it will present a tab showing which versions of the .NET Compact Framework are installed. If version 3.5 is not already on the mobile device, install it using the following steps:

1. Locate **NETCFv35.wm.armv4i.cab** in the proper directory on the source computer.
2. Install this CAB file on the mobile device as described in [How to Install CAB Files](#).

### Step 4: Install sqlce.phone.wce5.armv4i.CAB on the mobile device (if necessary)

This CAB, if installed, will add SQL Server CE 3.5 to the device. SQL Server CE is a necessary component of the Oracle Mobile WAM application. However it is possible that it already exists on some PDAs. If 3.5 has already been installed, or even in some cases if an earlier version of SQL Server has already been installed, it may be usable as

is. If an older version exists that is not usable by Oracle Mobile WAM and is no longer needed for other purposes, it can be uninstalled by using a method similar to the one for de-installing the main program. Please refer to [Appendix F: Uninstalling Mobile from the PDA](#).

If no usable version of SQL Server CE is installed on the PDA, use the following steps to install SQL Server CE 3.5:

1. Locate **NETCFv35.wm.armv4i.cab** in the proper directory on the source computer.
2. Install this CAB file on the mobile device as described in [How to Install CAB Files](#).

#### **Step 5: Install OracleMobileDeviceinstaller.CAB on the Mobile Device**

This procedure installs the Oracle Mobile WAM application (OracleMobile.exe) into a folder named "OracleMobile" inside the Program Files folder and also adds an icon for it to the Start screen.

1. Locate **OracleMobileDeviceinstaller.CAB** in the proper directory on the source computer.
2. Install this CAB file on the mobile device as described in [How to Install CAB Files](#).

#### **Step 6: Associate OracleMobile.exe with the appropriate .NET CF 3.5 if necessary**

If it was necessary to install version 3.5 of the .NET Compact Framework on the device or if there were already two or more versions of the .NET Compact Framework on the device, it will be necessary to associate the Oracle Mobile WAM application specifically with version 3.5 of the .NET Compact Framework. To do this, use the following steps:

1. Tap the .NET Configuration utility icon from the start screen.
2. Click the Application Policy tab at the bottom right of the utility.
3. Select the OracleMobile.exe application path.  
This can be found in the topmost drop down list entitled "Override device policy for application:"
4. Select the 3.5.x .NET CF version number.  
This can be selected from the drop down list entitled "Run with .NET Compact Framework version." This will likely be a value such as, 3.5.7283, but the last digits might be different.
5. Tap the Apply button.
6. Close the utility.

#### **Step 6. Restart the PDA**

Power the PDA off and then back on to make sure that the changes have taken effect.

## Configuring the PDA

After installing Oracle Utilities Work and Asset Management Mobile onto the PDA, configure the display and configure the device to point to the appropriate application server.

### Configuring the Device Display

On many devices no configuration of the device display is necessary – the program will automatically attempt to configure itself to the available display size and resolution. However, some devices may require additional instructions to properly configure the height of combo boxes and the width of grid columns. Should adjustments to these types of dimensions be necessary, a configuration file can be added to the OracleMobile directory that will inform the program of the proper dimensions.

1. Open any pure text editor on the source PC (e.g. Notepad, etc.).
2. Create a new file and add only the following lines to the very top:  
`COMBOBOX_FONTSIZE=11.0`  
  
`GRID_COLWIDTHMULTIPLIER=2`
3. Adjust the Combo box font size if necessary.  
 If combo boxes in the application are too tall or too short, adjust the COMBOBOX\_FONTSIZE value down or up, as appropriate. If grid column widths are too wide or too narrow, adjust the GRID\_COLWIDTHMULTIPLIER down or up as appropriate.
4. Save this file as a text file with the name “omdev.cfg” (not omdev.cfg.txt). Use the option to save as a UTF8 file if possible.
5. Copy this file to the OracleMobile directory on the PDA.
6. Restart OracleMobile on the PDA and observe the effect of these configurations. Readjust and reapply as necessary.

### Configuring the Device for Data Acquisition

After installing Oracle Utilities Work and Asset Management Mobile onto the PDA, configure the device to point to the proper application server.

1. Open the Oracle Utilities Work and Asset Management Mobile application on the PDA.  
 Start the application by tapping on Start > Programs > Mobile.  
  
 When the application is started on the PDA for the first time, a message displays asking you to download data.
2. Tap on the OK button to clear the message.
3. Select Admin from the menu then Configure.
4. In the Server Name field, enter the application server and folder where the Web Service was installed.  
 The server name “192.168.31.69/prod” is shown in the example to the right. Multiple instances of the Web Service can be installed in separate folders on the same Internet Information Services (IIS).
5. When finished entering the Server Name, tap the Save button.
6. The PDA is now ready for downloading data in preparation for daily use.

---

---

# Appendix D: Requirements and Recommendations

## Mobile Device

- ARM Processor (most common are the XScale processors)
- Microsoft Pocket PC 2003 or greater (a.k.a. Microsoft Windows Mobile)
- TCP/IP connectivity
  - Wireless (802.11 compliant B or G) - either built in to the device or using a wireless CompactFlash card.
  - Wired - using a CompactFlash to Ethernet card.

## Application Server Software

- Operating System (any of the following Microsoft products):
  - Windows 2000 Professional
  - Windows 2000 Advanced Server
  - Windows 2003 Server
  - Windows XP Professional

There are various supporting applications that need to be installed on the Application Server. Please refer to the [Verify Application Server Prerequisites](#) section of the Mobile Web Service Installation Guide for these prerequisites.

## Wireless Access Point/Router

- 802.11 compliant B or G (G is recommended). This would only be needed if using wireless device connectivity.

## Recommendations

The purpose of this section is to help advise clients who have questions regarding the hardware they should purchase when implementing the Oracle Utilities Work and Asset Management Mobile solution. This is in NO WAY INTENDED to promote or endorse a certain brand of equipment or that we are certifying the usage of our mobile solution on specific equipment.

## Mobile Devices

- Dell Axim



- X30 with 802.11 wireless technology
- X50 with 802.11 wireless technology
- Intermec 700C Barcode scanning device

### **Wireless Access Point/Router**

- Linksys
  - Wireless-G Access Point (WAP54G)
  - Wireless-G Broadband Router (WRT54G)

The “G” standard is preferred, but customers with “B” type equipment can continue using that equipment with Mobile.

---

# Appendix E: Web Services Installation

## Pre-Installation Guidelines

If you are upgrading from a prior version, you must review the following sections to determine whether you need to follow additional steps to complete a successful upgrade.

## Upgrading from Synergen

---

Synergen Mobile Users ONLY – complete this section.

---

If you are upgrading from the Synergen Mobile solution (which uses a Microsoft Access database to move data to and from the PDA device) you must follow the instructions in this section before upgrading to Oracle Utilities Work and Asset Management 1.8.1. Failure to do so will render the existing Mobile solution unusable (based on a per database instance) and you will not be able to upload your PDA transactions to the system.

If this does not apply to your version of the application please proceed to section 1. Before Upgrading to Oracle Utilities Work and Asset Management 1.8.1.

1. Upload data from ALL PDA devices to the database (specifically the smu\_transactions table). This is data that is specific to the database you are upgrading. If you have data on a PDA that isn't from the database you are upgrading then this step isn't relevant.
2. Run the smu\_interface.sdbp\_mu\_interface batch job. This MUST be run to load the transactions into the appropriate database tables.
3. Check the smu\_transactions table to make sure the SENT\_TO\_INTERFACE\_IND = 'Y' for ALL rows. Do this after the batch job has finished running. If the indicator is not set to yes, research and fix the issue(s) then re-run the batch job.

After all rows have been successfully uploaded into the system, you can proceed to the Oracle Utilities Work and Asset Management 1.8 installation.

## Guidelines for Users That will NOT use Service Request

---

ALL Users - Before Upgrading

---

The PDA allows you to manage both work orders and service requests. To support Service Request functionality, the data for ALL active customers (including their addresses) is downloaded to the device. This can take a long time and can use a lot of resources on the PDA. If you are not using service request functionality it is recommended that you add a line of code to the where clause in the SMD\_CUSTOMER view, as shown in the example below:

```
CREATE OR REPLACE FORCE VIEW SYNERGEN.SMD_CUSTOMER

(PLANT, CUSTOMER_ID, CUSTOMER_LAST_NAME,
CUSTOMER_FIRST_NAME, CUSTOMER_MIDDLE_NAME, COMPANY)

as

SELECT plant, customer_id, customer_last_name,
customer_first_name,

customer_middle_name, company

FROM sa_customer

WHERE customer_status = 'ACTIVE'

AND rownum < 2; -- Limit Customers downloaded to PDA
```

The view in this example may look slightly different than the view in the database because commented lines, like versioning information, were removed.

## 1.0 Verify Application Server Prerequisites

- Oracle Utilities Work and Asset Management version 1.8.
- Microsoft Internet Information Services (IIS)
- Microsoft.Net Framework Ver. 1.1
- Microsoft ASP.Net
- Microsoft Data Access Components (MDAC) 2.6 or later
- Oracle 9i Client Release 2
- .NET Managed Provider for Oracle 1.0 or later

*Note:* During this installation process, the Application Server will have to be rebooted.

### 1.1 Verify Oracle Utilities Work and Asset Management version 1.8x

### 1.2 Microsoft Internet Information Services (IIS)

To verify that Microsoft Internet Information Services (IIS) is installed on the application server, go to Control Panel and double on the Administrative Tools icon. In the Administrative Tools folder, you will see an Internet Information Services icon, if IIS is installed on the application server. If it is not installed, please ask your Network Administrator to install it.

### 1.3 Microsoft.Net Framework Ver. 1.1

The Microsoft.Net Framework is included with Windows Server 2003 software. For Windows 2000 Professional or Windows 2000 Advanced Server, you can install it as an option. To verify that the Microsoft.Net Framework is installed on a Windows 2000 application server, open the

Control Panel and double-click on the Add or Remove Programs icon. On the list of currently installed programs, look for Microsoft .NET Framework 1.1. If it is not installed, please ask your Network Administrator to install it. This can normally be installed using Microsoft Windows Update which can be accessed by clicking Start > Windows Update.

#### 1.4 ASP.Net

*Windows 2000 Professional or Windows 2000 Advanced Server:* ASP.Net is installed with the Microsoft.Net Framework.

*Windows 2003 Server:* ASP.Net is already included with the product but sometimes has to be installed. To verify that ASP.Net is installed on your Windows 2003 Server or to install it, do the following:

1. Open the **Control Panel** and double-click on the **Add or Remove Programs** icon.
2. Click on the **Add/Remove Windows Components** button located on the left side of the window.
3. When the **Windows Components Wizard** opens, click on the word (not the check box) **Application Server** then click the **Details...** button.
4. In the **Application Server** window that opens, locate the **ASP.NET** subcomponent. If the check box for this item is checked, ASP.NET is already installed and you can exit out of the setup. If it is not, click on the check box to enter a check mark then click **OK**.
5. In the **Windows Components Wizard** screen click the **Next** button to begin the installation.
6. When the installation is complete, click the **Finish** button.
7. Later in this document steps will be listed showing how to verify that the ASP.NET extensions exist in your IIS program.

#### 1.5 Microsoft Data Access Components (MDAC)

To verify that Microsoft Data Access Components are installed on the application server, open the **Control Panel** and double-click on the **Add or Remove Programs** icon. On the list of currently installed programs, look for Microsoft Data Access Components. If it is not installed, please ask your Network Administrator to install it.

#### 1.6 Oracle 9i Client Release 2

Note: Search for A99330-0 on [edelivery.oracle.com](http://edelivery.oracle.com). This part number works for both x86 32-bit and 64-bit

To verify that the Oracle component is installed, see if you have SQL\*Plus connectivity. To do this, click on **Start -> Programs**, select the appropriate Oracle folder and start SQL\*Plus. Connect to the database that your Web Service will be connecting to. If you cannot find this icon or cannot connect to the database, please ask your Network or Database Administrator to set this up.

The TNSNames.ora entry for the database you will be connecting to must exist in the TNSNames.ora file on the application server, you should *not* use global TNSNames.

## 2.0 Install Web Service on the Application Server

Before you begin your installation, determine whether or not you have either of the following requirements and review the related Appendix sections if applicable:

- Overlaying the Existing Web Service
- Multiple Web Services for the Same Database

## 2.1 Unzip the contents of the Mobile18.zip file

After unzipping the files into a temporary directory, click on the **setup.exe** file to start the Oracle Utilities Work and Asset Management Mobile InstallShield Wizard.

## 2.2 Install Web Service on the Application Server

From this screen, you have options:

- **Oracle WAM Mobile Service 1.8.1** – Installs the Web Service that is compatible with Oracle Utilities Work and Asset Management release 1.8.1 and 1.8.1.x.
- **Oracle WAM PDA 1.8** - Extracts the setup files onto the mobile device. The setup file can be stored to a location where it can be installed on the mobile device later. This version is compatible with Oracle WAM Mobile Service 1.8.1.

Choose an option and click Next to set the web service location.

## 3.0 Identify the top level directory to Install the Supporting Web Service Files

Click Next to set the web service instance. Accept the default or click Browse to select a location.

Example: C:\synergen\

## 4.0 Identify the Web Service Instance to be Installed

Example: prod7

Click Next to set the database.

## 5.0 Identify the Database to be Installed

This value must match an existing TNS Names value that exists in the TNSNames.ora file.

Click Next to confirm your settings and install the web services.

Click Install to begin the installation of the web service file and creation of the IIS service.

## 5.1 Verify that the Web Service Installed to IIS Correctly

Open IIS by clicking on the **Internet Information Services** icon, which is usually located under **Administrative Tools** in the **Control Panel**. When IIS opens, follow these steps:

1. Expand this list on the left for the computer IIS is running on.
2. Expand the **Web Sites** list.
3. Expand the **Default Web Site** list (if you installed the Web Service on a different web site, expand the list for that web site).
4. Click on **OracleMobileService** in the list and then click your right-mouse button and choose **Properties**.
5. On the Properties window, click on the **Directory** tab.
6. Click on the **Configuration** button.
7. In the **Application Configuration** window that opens, on the **Mappings** tab, you will see an **Extension** list. In this list you should see a **.asmx** entry. If you don't see this entry in the list, follow the instructions in the **Repair IIS mappings for ASP.NET using Aspnet\_regiis.exe** section below.
8. Exit IIS.

## 5.2 Repair IIS Mappings for ASP.NET Using aspnet\_regiis.exe

To repair the IIS mappings for ASP.NET, do the following:

1. From the **Start** menu, click **Run**.
2. In the **Run** dialog box, enter **cmd** in the **Open** box and click on **OK**.
3. At the command prompt in the window that opens, type the following line:  
`"%systemroot%\Microsoft.NET\Framework\version\aspnet_regiis.exe" -i`

**Note 1:** %systemroot% is the directory where Windows is installed. (e.g., C:\Windows)

**Note 2:** In this path, version represents the version number of the .NET Framework installed on your server. You must replace this placeholder with the actual version number when you type the command. An example of this path would be:

C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322

**Note 3:** You must include the quotation marks in the command; the **-i** switch is after the end quote.

4. When the command begins running it will say, "Start installing ASP.NET..." When it finishes it will say, "Finished installing ASP.NET..."

## 6.0 Grant Permissions on Oracle Home Directory

There is an Oracle limitation that prevents Microsoft IIS from properly accessing the Oracle Home directory (Oracle Metalink Doc ID 215255.1). To resolve this issue, the following steps must be followed:

1. Launch Windows Explorer and navigate to the ORACLE\_HOME folder.  
This is typically the "Ora92" folder under the "Oracle" folder.
2. Right-click on the ORACLE\_HOME folder and choose the **Properties** option from the drop down list.
3. Click on the **Security** tab of the **Properties** window.
4. In the **Group or user names** list, you should see a group called **Authenticated Users**. Click on this Group. If you do not see this option, refer to [Add Authenticated Users Group](#) below.

5. In the **Permissions** list, *uncheck* the **Read and Execute** box, which is located under the **Allow** column.
6. Click the **Apply** button. There may be a slight delay while the process finishes.
7. Now, re-check the **Read & Execute** box located under the **Allow** column.
8. Click the **Apply** button again.
9. Click the **OK** button until you close out of all the security properties windows.
10. Reboot the server. THIS IS VERY IMPORTANT.

*Note:* If you later find that the Web Service is not able to query the database, you should complete the following additional steps:

1. Complete steps 1 through 7 above.
2. When you get to step 8 click on the **Advanced** button.
3. In the **Advanced Security Settings** window that appears, click on the **check box** next to “Replace permission entries on all child objects with entries shown here that apply to child objects”. (Note: if you later look at this setting you will NOT see a check here. After permissions are replaced, the check mark will disappear).
4. Click the **Apply** button.
5. When the security message appears, click **Yes** to continue. The process that will start may take a few minutes to complete.
6. Click the **OK** button until you close out of all the security properties windows.
7. Reboot the server.

## 7.0 Add Authenticated Users Group

If the **Authenticated Users** Group does not exist in the Group or user names list, you will need to do the following to add the Group:

1. On the **Security** tab of the **Properties** window, click on the **Add...** button.
2. In the **Select Users, Computers, or Groups** window, verify that the name of your server is shown in the **From this location** field. If not, do the following steps to select your server:
3. Click on the **Locations...** button.
4. Click on the item at the top of the Locations list, which should be the name of your server and click the **OK** button.
5. In the **Select Users, Computers, or Groups** window, click on the **Advanced...** button.
6. Click the **Find Now** button.
7. In the list that appears, select **Authenticated Users** and click the **OK** button, then click **OK** again. You should be back to the **Security** tab of the **Properties** window.
8. Click the **Apply** button.
9. You can now continue with setting the permissions for the Authenticated Users group.

## 8.0 Test the Installation

To test the installation, start Internet Explorer **on the application server**.

Enter the following URL in the address bar, replacing “prod7” with the name of the Virtual Directory/Service you entered when creating the Web Service:

`http://localhost/prod7/OracleMobileService/DataAccess.asmx`

If the IIS portion of the Web Service is working correctly, you will see a DataAccess page which lists the supported operations.

Click the “getPlants” link to open a debug page where you can test the HTTP Post protocol by following the instructions on the screen.

If the Web Service is connecting to the database properly, you will see a screen similar to the first of the two screens below. If the Web Service does NOT have proper connectivity it will look similar to the second screen.

### 8.1 Web Service has Proper Connectivity to Database

- Notice in the first section you see a reference to “sa\_plant” and then a reference to “PLANT”.
- In the bottom portion of the screen you should at least one Plant listed, like <PLANT>01</PLANT>. You will not see other plants like 02, 03, etc. unless you are running a multi-plant environment.

### 8.2 Web Service is NOT Connecting Properly to Database

- No Plants are visible on this screen because the Web Service is NOT connecting to the database.
- If a connection is NOT being made to the database you should go back and check the following:
  - Check TNSNames.ora file to make sure the proper TNS Name exists in this file.
  - In SQL\*Plus, sign on to the database and run a query to make sure you get data back.
  - Open the Web.config file in the OracleMobileService directory (or the applicable MobileService directory if you are installing one of the earlier services) to make sure the Data Source value was saved properly. See the “Web.config file” section in the Appendix for more information.
  - Check the permissions for the Oracle Home directory as described in the “Grant Permissions on Oracle Home directory” section, earlier in this document.

## 9.0 Set Up the Mobile Batch Job

When transactions are uploaded from PDA devices, the information is stored initially in a temporary transaction table. Batch processing (smu\_interface.sdbp) then uses data from this table to create and update the appropriate Oracle Utilities Work and Asset Management records. Existing Mobile customers have already configured the batch job and no changes are



necessary. New Mobile customers should follow the instructions in this section to setup the batch job.

1. Open the Job Manager module.
2. Select Create Job from the Actions list on the Search Options screen.
3. Enter job information.

In the description field, enter `smu_interface.sdbp_mu_interface`, followed by the job number, the plant code that you want this job to process and 'N', followed by a semi-colon. Below is an example where the available job number is 21.

```
smu_interface.sdbp_mu_interface(21,'01','N');
```

*Note:* **Note:** It is recommended that you enter the last value as 'N' instead of 'Y' so that records will remain in the `smu_transactions` table after they are loaded into the various tables. If this indicator is set to 'Y', the records will be purged from `smu_transactions` after they load into the application tables.

You can then enter a time interval for automatic cycling of the procedure and the next run date (and time) that you want the procedure to begin cycling. It is recommended that an interval of 10 minutes is used. To set an interval of 10 minutes, you would enter the following in the Interval field in Job Manager: `SYSDATE + .00694`

4. Click the Save icon when you have entered all necessary information.

## 10.0 Extract the PDA Files

To extract the PDA installation files, to be used for installing the PDA application onto the mobile device, start the WAMMobile InstallShield application, select the **Oracle WAM PDA 1.8** option and click **Next**. On the next screen the folder indicated will be defaulted into the Destination Folder field.

You can either use this folder or use the **Browse...** button to select a different destination. Click **Next** and on the following screen, then click **Install**. This will copy the PDA setup files into a folder called WAMPDA18, in the folder you selected (the default folder is C:\MobileSetup). From there you can install the application onto the mobile device.

## 11.0 Final Steps

You should now be able to connect to the Web Service from the Mobile device. To test connectivity from the device, open the Oracle Mobile application on the device and do the following:

1. From the **Action** menu, select **Synchronize**.
2. On the **Synchronize** page, select the **Config** tab.
3. In the **Server Name** field enter the name of the application server for which you want to test connectivity and click the **Save** button.
4. Select the **Download** tab.
5. Tap the **Refresh Lists** button. You will see the Pocket PC clock/timer icon spinning.

The clock/timer icon will go away when the refresh process is complete. This refresh process should take less than twenty seconds. If the refresh process goes on for more than one minute there is a problem with the connection. If this is the case, you may have to press the Reset button on your device.

WARNING: You may lose data on the device that has not been saved by pressing the Reset button.

6. To test a full download, enter selection in the Plant, Storeroom & Crew fields and click the **Download** button. It will take a few minutes to complete the data download to the device.

## Additional Information

### Steps to Overlay the Existing Web Service

For this example, the existing web service is “prod7,” the existing database is also “prod7,” and the virtual directory is located in the following location:

C:\synergen\prod7

1. You will need to stop and restart IIS to make sure there are no locks on the service. There are various ways to do this. One way is to open the Windows Services panel, click on **IIS Admin** and click on the **Restart Service** button. NOTE: This will kill any other IIS processes that are running on the server.
2. Begin the installation and go through the steps until you get to the “WAM Mobile Installation Root” screen (where you select the Destination Folder for the web service). If your root directory for the web service is the same as the default c:\synergen, click **Next**. If the root is different, click the **Browse...** button and select your existing root directory then click Next to continue.
3. On the next screen, where you enter the “Service” name, enter the name of your existing virtual directory/web service. In this example it would be “prod7”. Click Next to continue.
4. On the next screen, where you enter the “Database” name, enter the name of your existing Database. In this example it would also be “prod7”.
5. Click **Next** and finish out the installation by clicking Install on the installation screen.

### Multiple Web Services for the Same Database

It is possible for customers to install more than one of the three web services, included in this release, onto the same application server, for use in the same database environment. The new Oracle Mobile Service can coexist in the same virtual directory as one of the earlier Mobile services (1.2.12 or 1.3.1). The two earlier services cannot, however, exist together in the same virtual directory, because the both use the name “WAMMobileService”.

To workaroud this limitation, you will need to enter a unique name for the “Service”, which will be your virtual directory, when running the InstallShield application. For example, instead of calling the Service “Prod7”, you may want to call it “Prod71219” (“1219” is used to reflect the version of the PDA app that it is compatible with). The “Database” value will be the TNS for your 1.8 environment.

*Note:* If you want to continue using the 1.2.19 or 1.2.21 of the EAM Mobile solution on some devices, you will still need to overlay your existing web service with proper EAM Mobile Service (as outlined in the “Overlaying existing web service” section).

After installation you should see the following directory structure on your application server:

C:\synergen\PROD7\OracleMobileService

And the following web service in IIS:

PROD7\OracleMobileService

## Errors Encountered on PDA and Steps for Troubleshooting

Below are error messages your users may encounter when connecting their devices to the Web Service. Also listed are possible reasons for the error and possible solutions to help resolve the issue.

1. Error: Unable to connect to http://<Server>/OracleMobileService/DataAccess.asmx

Possible Reason: The app is not finding the URL/Web Service.

Possible Solutions:

- Check spelling on value entered in the Server Name field.
- Enter the http://<Server>/OracleMobileService/DataAccess.asmx URL in a browser to see if the page can be hit.
- Open up IIS on the web server and verify that the Web Service is running.

2. Error: Out of Memory error

Possible Reason: You are downloading a very large dataset and the system cannot process it as a whole at one time.

Possible Solution: modify the Web.config file to process the data in “chunks.” See section 9.4 for more details.

3. Error: Web Service cannot find database.

Possible Reason: The Web Service is not connecting to the database.

Possible Solutions:

- In IIS, look at the Web.config file for the Web Service. Make sure there is a valid TNS entry in the Data Source= field.
- TNS Connectivity - Make sure the TNSNames.ora file on the web server contains a valid database string, for the database you are trying to connect to.
- The permissions may not be set correctly on the Oracle directory on the web server.

## Web.config file

During installation of the Web Service, the TNS Names value for the database you were installing was placed in a file called Web.config. This value should match the TNS Names value in the TNSNames.ora file. If you were to switch databases, and keep the same OracleMobileService installation, you would need to change the value in the Web.config file. To update this value do the following:

1. In **Windows Explorer** navigate to the directory where the OracleMobileService Web Service was installed.
2. Open the Web.config file. Look for the “Data Source=” entry. The line you will be looking for looks like the following line:  
`<add key=”Data_Source” value=”Data Source=PROD7;”/>`
3. Change the value where “PROD7” is located in this example, to the new TNS Names value.
4. Save and exit when finished updating this entry.

## Preventing Out of Memory Errors

Large download datasets sometimes cause out of memory errors. For instance, at around 20,000 asset rows we have seen out of memory errors occur even when there is enough physical space left on the mobile device. To eliminate the problem of server side out of memory errors while preparing very large datasets, an alternative download mechanism was added that will be employed for large datasets. This mechanism breaks down the large dataset into “chunks,” several small datasets that can be downloaded individually to prevent these errors.

The Web.config file is an XML file stored in the OracleMobileService web service directory. It consists of multiple sections of hierarchical tags. The tag section that will control chunked downloading in the mobile app is called the “appSettings” tag section. Here is an example with all the chunk size settings set especially low:

```
<appSettings>
  <add key= "Data_Source" value="Data Source=DV18;" />
  <add key="WebService_Version" value="18-3" />
  <add key="Log_Path" value="\omwslogs\" />
  <add key="Log_Strategy" value="VERBOSE" />
  <add key="Asset_ChunkSize" value="10000" />
  <add key="Customer_ChunkSize" value="10000" />
  <add key="CustomerAddr_ChunkSize" value="10000" />
  <add key="Vendor_ChunkSize" value="10000" />
</appSettings>
Data Source
```

This value should be set to the TNS name of your database in the TNSNames.ora file on the application server where your Web Service is running.

**WebService\_Version** - The version number of the current Web Service. This value should only be modified by the development team.

**Log\_Path** - This value indicates the physical path where the log file will be saved. Verify that it is set to the appropriate location and change it if necessary. The default should be set so that the file is stored inside the web services directory.

**Log\_Strategy** - This indicates how the log will be written: ON, OFF or VERBOSE. Setting this to ON or VERBOSE in production is not advised as these settings may slow down performance. The setting is case-sensitive so should always be in all caps.

ON = the standard setting - provides error logging only

OFF = no logging at all - not even of errors

VERBOSE = This setting provides extended logging and is useful for debugging and profiling.

**<module>ChunkSize** - – The value entered represents the number of records that will be included in each chunk. For example, if you enter a value of 500 and there are 10001 records to be downloaded, the records will be taken in three “chunks”, two chunks of 500 records and one chunk of 1 record. Entering a higher number will lead to a faster download but with a higher risk for out of memory errors.

### **Notes on ChunkSize Values:**

- The value entered should be a literal string such as 10000 with no commas.
- Setting the value to zero disables chunking entirely (all records will be downloaded with no chunking).
- Starting defaults are usually set to 10000. Higher numbers such as 20000 may result in out of memory errors for some sets of records but not other. This depends on how much data is typically in each record. Given this, it is possible that for some modules the setting of 10000 will be too high.
- Chunking only addresses server side out of memory errors. Client side memory issues can easily be encountered with very large record sets and required add-on memory.

---

# Appendix F: Uninstalling Mobile from the PDA

If this is your first installation of Mobile on the PDA this section does not apply.

If you are an existing Mobile user, it is possible that you are upgrading from SPL EAM Mobile, Synergen Mobile or an older version of Oracle Mobile. In this case it is highly recommended that you upload any existing transactions and uninstall the older version before continuing with the upgrade. In some cases failure to remove previous versions before upgrading can result in the loss of data or performance problems.

To uninstall, you need to upload existing transactions, remove the database file, and remove the application program.

## Upload Existing Transactions from the PDA

You can verify if the PDA has transactions by starting the Mobile application and synchronizing. Tap the Upload tab if it is not already selected. If a message appears saying "Transactions on Device: 0", no transactions exist on the PDA and you are ready to proceed with the Uninstall process. If transactions do exist on the device, upload them into the system before continuing.

[For more information on uploading transactions, please refer to the section titled "How to Upload Data to Oracle Utilities Work and Asset Management".](#)

## Uninstall the Mobile Application from the PDA

1. Close Mobile if the application is running.
2. Tap the Start menu, then tap Settings.
3. Select the System tab.
4. Tap the Remove Programs icon.
5. Locate the mobile file (e.g. SPL WorldGroup, Inc. SPL EAM Mobile, Oracle Mobile) in the "Programs in storage memory" box.
6. Select the mobile file, then tap the Remove button.
7. When prompted to remove the program tap Yes and the uninstall process begins.
8. After the uninstall process is complete, tap on the "OK" button to return to the main screen of the PDA.

## Uninstall the Database File from the PDA

1. Tap the Start menu, then tap Programs and tap File Explorer.
2. Navigate to the top level of the "My Device" tree.
3. Locate and delete the database file (e.g. SPLEAMMobile, ORACLEMobile).  
To delete the file, tap and HOLD the stylus on the file until the menu appears.

- 
4. Tap Delete.
  5. When asked to confirm the deletion, tap the Yes button.  
The uninstall process is now complete.

---

# Index

## A

- Actions 17
- Asset Data 16

## C

- CC&B Integration 24
- Chunking 47
- Closeout 17
- Common Actions 17
- Common Views 14
- Configuring the PDA 28, 33
- create new 14
- Crew Time 17
- Customer Billing 17

## D

- Dates 10
- Download 7
- Drop-Down Lists 10

## F

- Field Values 10

## I

- Installing Mobile 27, 31

## L

- Labor 16

## M

- Material 15
- Meter Information 25

## N

- Navigation 5
- Notes 14

## O

- Other Charges 18
- Out of Memory Errors 47

## P

- PDA
  - configuring 28, 33

- PDA Interface 9

- PDA Requirements 34

- Permissions 40

- PM Routes 21

- Popups 9

## R

- Resizing 10

## S

- Scroll Bars 10
- Searching 10
- Server Prerequisites 37
- Service History 15
- Service Requests 14
- Specification Details 16
- Stock Charges 18

## T

- Task Progress 19

## U

- Uninstalling Mobile 49
- Upload 8

## V

- Views 14

## W

- Web Service 39
- Work Order Material 15
- Work Orders
  - create new 13