Oracle® Utilities Work and Asset Management

Interfaces Guide

Release 1.9.0.3

July 2011

ORACLE

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

Doc v1 rev.0 7/12/11

Primary Author: TUGBU Documentation

Contributing Author: n/a

Contributor: n/a

Contents

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

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

Description	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".
Syntax	WIFP_ACCRUAL_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_ACCRUAL
Business Rule	ACCRUAL INTERFACE
Sequence Number	N/A

Interface Name	**Alert**
Direction	IN
Description	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.
Syntax	WIFP_ALERT_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_ IN,PURGE_IN, OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_ALERT
Business Rule	NONE
Sequence Number	N/A

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

Interface Name	**Blanket Contract**
Direction	IN/OUT

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

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

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

Interface Name	**Cost Adjustment**
Direction	IN
Description	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.
Syntax	WIFP_COST_ADJUSTMENT_INTERFACE(JOB_IN,PLANT_IN, DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)

Table Name	WAIF_COST_ADJUSTMENT
Business Rule	NONE
Sequence Number	SA_ADJUSTMENT

Interface Name	**Customer**
Direction	IN
Description	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.
Syntax	WIFP_CUSTOMER_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_CUSTOMER
Business Rule	CUSTOMER INTERFACE, INTERFACE PARAMETERS where Parameter Name = CUSTOMER ADDRESS PARSER VALUE: wipf_parse_address
Sequence Number	SA_CUSTOMER

Interface Name	**Direct Charges**
Direction	IN/OUT
Description	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.
Syntax	WIFP_DIRECT_CHARGE_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_DIRECT_CHARGE
Business Rule	NONE
Sequence Number	SA_DIRECT_CHARGES

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

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

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

Interface Name	**Fuel**
Direction	IN
Description	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.
Syntax	WIFP_FUEL_INTERFACE(JOB_IN,PLANT_IN, DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_FUEL_SYSTEM_DATA
Business Rule	EXPENSE CODES where Category = CONSUMABLES
Sequence Number	N/A

Interface Name	**General Ledger**
Direction	OUT
Description	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.

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

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

Interface Name	**Invoice**
Direction	IN/OUT
Description	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.
Syntax	WIFP_INVOICE_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_INVOICE
Business Rule	INTERFACE PARAMETERS where Parameter Name = INVOICE STATUS
Sequence Number	N/A

Interface Name	**Invoice AP Data**
Direction	IN
Description	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.

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

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

Interface Name	**Meter Reading**
Direction	IN
Description	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.
Syntax	WIFP_METER_READING_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_METER_READING
Business Rule	N/A
Sequence Number	N/A

Interface Name	**Purchase Order**
Direction	IN

Description	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).
Syntax	WIFP_PURCHASE_ORDER_INTERFACE.PO_INTERFACE(JOB_IN, PLANT_IN,DIRECTION_IN,PURGE_IN,OPTION_IN,PRE_IN,POST_ IN)
Table Name	WAIF_PURCHASE_ORDER
Business Rule	UPDATE PRIMARY VENDOR where Prime Vendor Rule = UPDATE_ PRIME_ VENDOR
Sequence Number	SA_PURCHASE_ORDER

Interface Name	**Receipt**
Direction	IN
Description	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).
Syntax	WIFP_RECEIPT_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_RECEIPT
Business Rule	NONE
Sequence Number	N/A

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

Interface Name	**Service Request**
Direction	IN

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

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

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

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

Syntax	WIFP_VENDOR_INTERFACE(JOB_IN,PLANT_IN,DIRECTION_IN, PURGE_IN,OPTION_IN,PRE_IN,POST_IN)
Table Name	WAIF_VENDOR
Business Rule	VENDOR INTERFACE
Sequence Number	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_inIN 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_inINNUMBER,
plant_in INVARCHAR2,
job_seq_in_outIN OUTNUMBER,
 status_inINVARCHAR2,
 job_desc_msg_inINVARCHAR2,
 error_noIN OUTNUMBER,
 error_msgIN OUTVARCHAR2
)
--
```

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_inINNUMBER,
  plant_inINVARCHAR2,
  job_seq_inINNUMBER,
  message_in INVARCHAR2,
  error_noIN OUTNUMBER,
  error_msgIN OUTVARCHAR2
)
```

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.

PROCEDUREwifp_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

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

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

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

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="tns1: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" />
```

```
    <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>
    </complexContent>
    </complexType>
    </schema>
    </types>
```

```
  - <wsdl:message name="SelectAssetDataResponse">
    <wsdl:part name="parameters"
element="tns1:SelectAssetDataResponse" />
    </wsdl:message>
  - <wsdl:message name="InsertAssetDataRequest">
    <wsdl:part name="parameters" element="tns1:InsertAssetData" />
    </wsdl:message>
  - <wsdl:message name="UpdateAssetDataRequest">
    <wsdl:part name="parameters" element="tns1:UpdateAssetData" />
    </wsdl:message>
  - <wsdl:message name="SelectAssetDataRequest">
    <wsdl:part name="parameters" element="tns1:SelectAssetData" />
    </wsdl:message>
  - <wsdl:message name="DeleteAssetDataResponse">
    <wsdl:part name="parameters"
element="tns1:DeleteAssetDataResponse" />
    </wsdl:message>
  - <wsdl:message name="InsertAssetDataResponse">
    <wsdl:part name="parameters"
element="tns1:InsertAssetDataResponse" />
    </wsdl:message>
  - <wsdl:message name="UpdateAssetDataResponse">
    <wsdl:part name="parameters"
element="tns1:UpdateAssetDataResponse" />
    </wsdl:message>
  - <wsdl:message name="DeleteAssetDataRequest">
    <wsdl:part name="parameters" element="tns1: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:

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

- DeleteSpecificationDataData - used for deleting specification header information from Oracle Utilities Work and Asset Management
- SelectSpecificationDataData - 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="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_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="tns1:UpdateSpecificationDataData" />
    </wsdl:message>
  - <wsdl:message name="InsertSpecificationDetailsDataRequest">
    <wsdl:part name="parameters"
element="tns1:InsertSpecificationDetailsData" />
    </wsdl:message>
  - <wsdl:message name="InsertSpecificationDataDataRequest">
    <wsdl:part name="parameters"
element="tns1:InsertSpecificationDataData" />
    </wsdl:message>
  - <wsdl:message name="DeleteSpecificationDetailsDataResponse">
    <wsdl:part name="parameters"
element="tns1:DeleteSpecificationDetailsDataResponse" />
    </wsdl:message>
```

```
  - <wsdl:message name="InsertSpecificationDataDataResponse">
    <wsdl:part name="parameters"
element="tns1:InsertSpecificationDataDataResponse" />
    </wsdl:message>
  - <wsdl:message name="InsertSpecificationDetailsDataResponse">
    <wsdl:part name="parameters"
element="tns1:InsertSpecificationDetailsDataResponse" />
    </wsdl:message>
  - <wsdl:message name="DeleteSpecificationDataDataResponse">
    <wsdl:part name="parameters"
element="tns1:DeleteSpecificationDataDataResponse" />
    </wsdl:message>
  - <wsdl:message name="SelectSpecificationDataDataRequest">
    <wsdl:part name="parameters"
element="tns1:SelectSpecificationDataData" />
    </wsdl:message>
  - <wsdl:message name="SelectSpecificationDetailsDataRequest">
    <wsdl:part name="parameters"
element="tns1:SelectSpecificationDetailsData" />
    </wsdl:message>
  - <wsdl:message name="UpdateSpecificationDetailsDataRequest">
    <wsdl:part name="parameters"
element="tns1:UpdateSpecificationDetailsData" />
    </wsdl:message>
  - <wsdl:message name="DeleteSpecificationDataDataRequest">
    <wsdl:part name="parameters"
element="tns1:DeleteSpecificationDataData" />
    </wsdl:message>
  - <wsdl:message name="UpdateSpecificationDataDataResponse">
    <wsdl:part name="parameters"
element="tns1:UpdateSpecificationDataDataResponse" />
    </wsdl:message>
  - <wsdl:message name="DeleteSpecificationDetailsDataRequest">
    <wsdl:part name="parameters"
element="tns1:DeleteSpecificationDetailsData" />
    </wsdl:message>
  - <wsdl:message name="SelectSpecificationDetailsDataResponse">
    <wsdl:part name="parameters"
element="tns1:SelectSpecificationDetailsDataResponse" />
    </wsdl:message>
  - <wsdl:message name="SelectSpecificationDataDataResponse">
    <wsdl:part name="parameters"
element="tns1:SelectSpecificationDataDataResponse" />
    </wsdl:message>
  - <wsdl:message name="UpdateSpecificationDetailsDataResponse">
    <wsdl:part name="parameters"
element="tns1: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 'seqence 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

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

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 *added* 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

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

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 **I**N or **O**UT record
RECORD_TYPE	VARCHAR2(1)	Indicates a **H**eader or **L**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

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

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 'H'eader and 'L'ine 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 'H'eader 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 'L'ine 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

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

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.

		Description
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

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 OPTON_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

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, eve 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

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

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 ('H'eader, 'L'ine and 'A'ccount).		
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"nbound or "O"utbound
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

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 he 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

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

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_PURHASE_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 [1]
RECORD_TYPE*	VARCHAR2(1)	Indicates Header, Line, Default Account Or Line Account
SENT_TO_INTERFACE_IND	VARCHAR2(1)	If PO Was Successfully Processed
ACTION_TYPE*	VARCHAR2(1)	Indicates PO Insert, Update, or Cancellation
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 [2]
DIVISION	VARCHAR2(40)	Division name of the vendor address [2]
ADDRESS	VARCHAR2(200)	Street address of the vendor [2]
CITY	VARCHAR2(40)	City of the vendor's address [2]
STATE_PROVINCE	VARCHAR2(4)	State or province of the vendor's address [2]
POSTAL_CODE	VARCHAR2(10)	Zip/Postal code of the vendor's address [2]
VENDOR_CONTACT	VARCHAR2(20)	Contact name for the vendor [2]
VENDOR_PHONE_NO	VARCHAR2(30)	Contact phone number for the vendor [2]
TOTAL_AMOUNT	NUMBER(15,2)	Total amount of PO line items [3]
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
EXPEDIATE_STATUS	VARCHAR2(10)	Status of the expedite process
EXPEDIATE_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[4]
BUYER_PHONE_NO	VARCHAR2(30)	Buyer's phone number
CURRENCY_CODE	CHAR(3)	Currency code for the vendor - Code Table 210 [5]
EXCHANGE_RATE*	NUMBER(15,7)	Exchange rate for the currency code [5]
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 [6]
STOCK_TYPE	VARCHAR2(15)	Stock type code [6]
STOCK_CODE	VARCHAR2(20)	Stock code [6]
ITEM_DESC****	VARCHAR2(2000)	The description of the item to be purchased [6]
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 [6]
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 [7]
UNIT_PRICE*	NUMBER(17,4)	Unit price [7]
ITEM_TOTAL_AMOUNT*	NUMBER(15,2)	Total amount of PO item [7, 8]
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 [9]
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 [10]
UNITS	NUMBER(12,5)	

* required fields

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Currency Code can be provided instead of Exchange Rate, since the appropriate value will be pulled from the appropriate Code Table.

[6] 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.

[7] Between PO Quantity, Unit Price, and Total Amount, only two out of three fields are required.

[8] 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.

[9] Field is NOT available for Change Orders.

[10] 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

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

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 ('H'eader, 'L'ine and 'A'ccount).		
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"nbound or "O"utbound
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

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

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

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 and 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'eader 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

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

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.

Bilding 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.

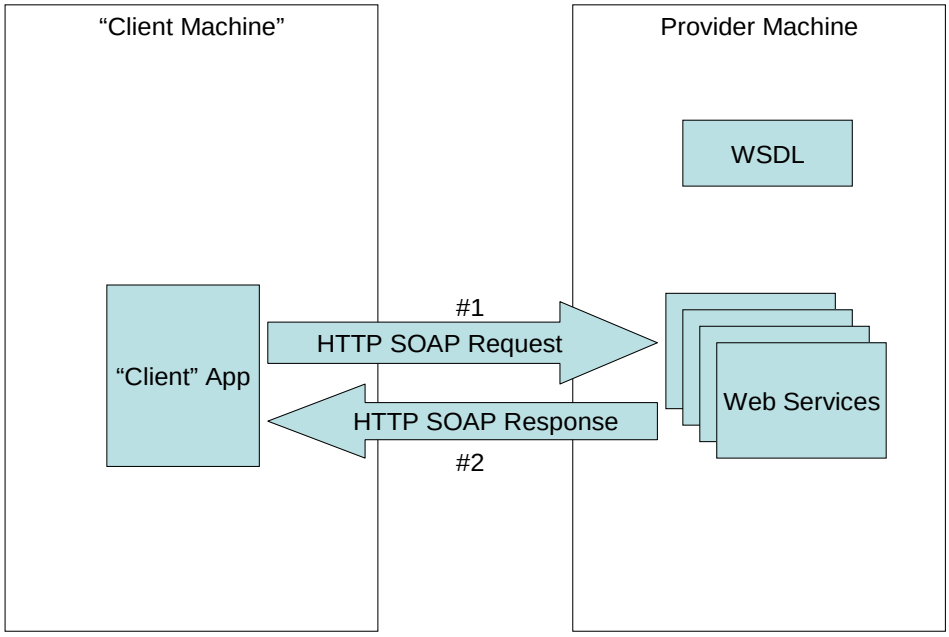
Web Services Provider Interface

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 in 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 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 files (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 applications 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 restults
    System.out.println("result : " + result);
    // Example get Parameters
    Map outparams = call.getOutputParams();

      // Example Get Array of Parameter Values
      Collection se1 = outparams.values();
      Object obj1[] = se1.toArray();

      sWO = (String)obj1[0];
    System.out.println("sWO : " + sWO);
    }
    catch (Exception e)
  {
```