Oracle® Utilities Work and Asset Management

Customization Guide

Release 1.9.1.1

E39488-03

December 2013

Contents

Preface

This guide provides guidelines, information and methods to customize the Oracle Utilities Work and Asset Management application. Extendable areas include the creation of custom actions, custom reports, custom forms and more. This guide also provides information on APIs, SMU transactions, web services and SAPI.

Audience

Oracle Utilities Work and Asset Management Customization Guide is intended for customers, partners and professional services who would like to extend the Oracle Utilities Work and Asset Management application to include custom functionality.

Related Documents

For more information on this release, please refer to the following related documentation.

- *Oracle Utilities Work and Asset Management Configuration Guide*
- *Oracle Utilities Work and Asset Management Installation Guide*
- *Oracle Utilities Work and Asset Management Online Help*
- *Oracle Utilities Work and Asset Management Release Notes Guide*
- *Oracle Utilities Work and Asset Management Upgrade Install Guide*
- *Oracle Utilities Work and Asset Management User Guide*

Conventions

The following text conventions are used:

Convention	Meaning
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	`Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.`

Chapter 1
Naming Conventions

Throughout the Oracle Utilities Work and Asset Management application, certain naming conventions are followed to allow you to easily recognize system objects. These objects fall into two broad categories, objects within the database and objects stored on the application server (these are typically file-naming conventions).

If you are extending the Oracle Utilities Work and Asset Management application, you should avoid using these same conventions to avoid confusion with system objects.

Database Schema Objects

There are many different objects in the Oracle Utilities Work and Asset Management database schema. Each of these objects are prefixed with a string so they are easily recognizable as Oracle Utilities Work and Asset Management objects.

The table below summarizes the various types of objects and their prefix strings.

Object Type	Prefix Convention	Comment
Indexes	PK_ FK_ I_	Primary Key Index Foreign Key Index Index
Procedures/Packages/Functions	SDBP_ WIFP_ SDBV_ SDBF_	Database procedures/packages Database procedures related to integration Specialized global variables package Database functions
Sequences	SSEQ_	Database sequence number generators
Tables	SA_ XT_ SAIF_ WAIF_	Database Table Metadata Database Table V1.8 and older integration tables V1.9 and newer integration tables
Triggers	SDBT_ SAUT_	Database trigger Database trigger related to auditing

Object Type	Prefix Convention	Comment
Views	SV_ SOLAP_ SMD_	Database View View related to a chart analytic View related to the Mobile Device Upload

The majority of the objects in the database begin with the letter "S". It is recommended that a different character be used as a prefix. Often "C" is used to denote "Custom" in place of the "S". Whichever prefix you decide to use, we recommend that the prefix differs from the standards above and that it be used consistently so you can easily distinguish which objects are related to extensions to the application.

Application Server Objects

There are many files delivered with the Oracle Utilities Work and Asset Management application, but most are delivered as binary runtime components. The exceptions to this rule are the report and chart files. These files can be copied and used as templates for new reports or charts.

Report Source Code

Report source code is delivered with the application in the form of Oracle Report files. These files all contain the same naming conventions. A typical file will be named S_RPT044.RDF.

- RPT is the standard prefix for standard reports. All base reports will contain this prefix.
- 44 is a report number assigned to the report.
- RDF is the Oracle Reports source file extension. REP is the Oracle Reports Runtime file extension.

When modifying a report, it is recommended that you copy the report file to a new name, make changes, and then enroll the new report into the Oracle Utilities Work and Asset Management application using the Report Administration module found in the Administration subsystem.

A new naming convention might be to replace the S_RPT with C_RPT to denote a modified report based on the original report. This is important so a software update does not overwrite your new report.

Chart Files

Chart files are spreadsheets that can be enrolled into the Oracle Utilities Work and Asset Management application as analytics. These analytic files do not follow a specific naming convention other than the name is descriptive of the metric. Some examples include:

- PM_forecast.xls
- PO_buyer.xls
- Corrective_vs_preventive_work.xls

You may use these files as templates for new metrics. You should rename the source file so it is not confused with the original analytic. This is important so a software update does not overwrite your new chart.

Chapter 2

Database Procedures

Writing new database procedures to affect the processing is a very common way to extend the application. As discussed earlier, all the core database objects have prefixes that a consistently used. You should be sure to name your new procedures using conventions other than those defined above

There are typically two type of procedures that are used to affect the processing, a database procedure to run as a nightly batch job, or a database procedure to run as part of a database trigger when a specific event occurs in the application.

Batch Database Procedures

There are several standards that you can follow that will allow you to prevent conflicts with software upgrades.

1. **Name your new programs avoiding any of the standard prefix conventions, see the** Naming Conventions **section of this document for examples.**
2. **Pass in at least two parameters, Job Number and Plant.**
   Job Number: This is the job that is assigned when you enroll the procedure in the batch job manager in the Administration subsystem. This number is important because it is a key to the Log and Message table associated to the job manager and will allow you to easily view messages on-line.

   Plant: Plant is also always passed in to support a multi-plant environment.

   A typical procedure definition is shown below.

```
create or replace procedure
custom_procedure_xxxxxxx (
  job_in IN NUMBER,
  plant_in IN VARCHAR2
) is
```

3. **Batch job messages are written in two tables which can be viewed on-line from the Job Manager Log if used correctly. There are two logging tables, sa_job_manager_log and sa_job_manager_log_message.**
   SA_JOB_MANAGER_LOG is the header record that is used to record when a job starts and ends. You should insert one record at the start of the procedure and then update the record upon completion of the procedure. The columns in this table are described below.

Plant:	Set to the plant passed in.
Job:	Set to the Job Number passed in.
Job_seq_no:	A generated sequence number table using the Oracle Sequence SASEQ_JOB_MANAGER_SEQ_NO.

Status: Typical statuses include:

- STARTED
- COMPLETED SUCESSFULLY
- COMPLETED WITH ERRORS
- FAILED

These statuses are user-defined and are applied depending upon where you are in the batch procedure when writing to the log. For example, at the top of the program, you might insert a row with STARTED as the status and then at the end of the procedure update the row with a final status based on the state of the program at completion.

Status_date: This is always sysdate on updates or inserts.

What: This is a column that is descriptive of the procedure.

SA_JOB_MANAGER_LOG_MESSAGE is the message table that contains progress information about the new batch job. This log is typically only inserted into with progress information. In some cases you may choose to write an entry into the log upon completion of each record or step in your process, or you may choose to write a summary record at the end of the job. For example, you may write a record stating, "100 asset processed with 0 errors.". The columns in this table are described below.

Plant: Set to the plant passed in.

Job: Set to the Job Number passed in.

Job_seq_no: A generated sequence number table using the Oracle Sequence SASEQ_ JOB_MANAGER_SEQ_NO.

Message: This is a descriptive string up to 2,000 characters.

Message_date: This is always sysdate.

Using this log effectively will allow you to track the batch procedure and also help to diagnose any problems you encounter with the procedure.

<u>Sample of code writing to the message log</u>

```
IF SQL%ROWCOUNT = 0 THEN
      v_err_message := 'Update of ….. Failed ';
      v_rec_desc := 'PO No. : '||c_rec.po_no||'/'||crec.po_item
      raise e_inside_loop;
END IF;
```

And in the exception block (note: count of errors and success is useful to identify our processing went, it MAY NOT be appropriate for all batch jobs)

```
WHEN e_inside_loop THEN
 -- rollback all previous DML statement
 ROLLBACK;
 -- insert detail record
 v_temp := v_error_message || ': ' || v_rec_desc;
 INSERT INTO sa_job_manager_log_message VALUES (plant_in,
      v_job_seq_no, job_in,'ERROR: ' || v_temp, SYSDATE);
 v_error_count := v_error_count + 1;  -- count # of errors
 COMMIT;
  …
```

4. **For information on how to load the new batch job in the Batch Job Manager, see the Batch Processing section below.**

   Note: When writing new database procedures, be sure to carefully consider the impact on the data in the Oracle Utilities Work and Asset Management system. The system can be adversely affected by customizations that are deployed incorrectly.

You may use these files as templates for new metrics. You should rename the source file so it is not confused with the original analytic. This is important so a software update does not overwrite your new chart.

Trigger Database Procedures

The second use of database procedures is to attach a procedure to a database transition that is identified using a database trigger. Base product has a number of these trigger to do logging, auditing, integration to BI, and other specialized uses.

Database triggers can be defined to be invoked on an insert, update, and/or delete transaction on a particular table. Triggers can be defined as before or after main commit processing on the table; the scope of processing can also be defined to be invoked at the table or row level. Please see the Oracle SQL Reference manual for more details on the creation of triggers.

Care must be taken in the writing of database triggers so as to not conflict with the online system. If you update fields on the table that is being updated in a trigger, you must make sure to do this in a before commit trigger or you risk getting “Record updated by another user” or “Mutating table” errors in Forms online module.

A good use of database triggers is to update other tables or kick off processing that is not trying to update the table that invoked the trigger. SAPI programming (see below) would be the preferred method if you need to do this kind of processing.

The source to the base application triggers is available in the database. You can view these for examples of how to build trigger processing.

Create Alerts

There are standard procedures you can call to send an alert in the application. These procedures are commonly used throughout the application when alerts are created. Alerts in the system will also send e-mail as well if the user profile for the addressee is set up to receive e-mail

To create a simple alert in the system, you can use the stored procedure defined below:

```
        sdbp_send_alert
        (
          addressee_name_in IN     sa_user_profile.username%TYPE,
          description_in    IN     sa_workflow_item.title%TYPE,
          module_name_in    IN     sa_workflow_item.module%TYPE DEFAULT NULL,
          where_clause_in   IN     VARCHAR2 DEFAULT NULL,
          order_by_in       IN     VARCHAR2 DEFAULT NULL,
          comments_in       IN     sa_workflow_item.full_desc%TYPE DEFAULT
NULL,
          locale_code_in    IN     sa_plant.locale_code%TYPE DEFAULT NULL
        ) ;
```

Parameters

- **ADDRESSEE_NAME:** username of the person the alert should go to
- **DESCRIPTION:** brief description of alert (best if kept under 60 characters)
- **MODULE:** form name of module to drill down (OPTIONAL)
- **WHERE_CLAUSE:** SQL Where Clause which will return the correct associated record(s) in a query if MODULE is filled in above. Do NOT include the word "WHERE". (OPTIONAL)
- **ORDER_BY:** if WHERE_CLAUSE above specified, it will return multiple records, you may specify SQL order by. Do NOT include the words "ORDER BY". (OPTIONAL)
- **COMMENTS:** extended description. Use this to give a detailed explanation of the nature of the alert. (OPTIONAL)
- **LOCALE_CODE:** For translatability; for english, set to 1 (OPTIONAL)

Examples

Here is an example to send an alert to JOUSER with a drill down to Work Request to look at a particular work request number. This would be the call from SQL*Plus prompt:

```
       SQLPlus> exec sdbp_send_alert('JOUSER','Take a look at this 
one!','WORKREQ','plant = ''01'' and work_request_no = 
''11333344''',null,'This is a crazy request.', 1);
       SQLPlus>commit;
```

Another way to send an alert to a number of users is to call a routine that sends an alert to everyone on an approval title.

Here is an example call to send an alert to everyone on the BYB approval title with the same work request information from above:

```
       DECLARE
         PLANT_IN VARCHAR2(3);
         APPROVAL_TITLE_IN VARCHAR2(6);
         DESCRIPTION_IN VARCHAR2(200);
         MODULE_NAME_IN VARCHAR2(200);
         WHERE_CLAUSE_IN VARCHAR2(200);
         ORDER_BY_IN VARCHAR2(200);
         COMMENTS_IN VARCHAR2(200);
         ERROR_NO NUMBER;
         DBMS_ACTIVITY VARCHAR2(200);
         ERROR_MESSAGE VARCHAR2(200);
       BEGIN
         PLANT_IN := '01;
         APPROVAL_TITLE_IN := 'BYB';
         DESCRIPTION_IN := 'Take a look at this one!';
         MODULE_NAME_IN := 'WORKREQ';
         WHERE_CLAUSE_IN := 'plant = ''01'' and work_request_no = 
''11333344''';
         ORDER_BY_IN := NULL;
         COMMENTS_IN := 'This is a crazy request.';
         ERROR_NO := NULL;
         DBMS_ACTIVITY := NULL;
         ERROR_MESSAGE := NULL;

         SDBP_SEND_APPROVAL_ALERT(
           PLANT_IN => PLANT_IN,
           APPROVAL_TITLE_IN => APPROVAL_TITLE_IN,
           DESCRIPTION_IN => DESCRIPTION_IN,
           MODULE_NAME_IN => MODULE_NAME_IN,
           WHERE_CLAUSE_IN => WHERE_CLAUSE_IN,
           ORDER_BY_IN => ORDER_BY_IN,
           COMMENTS_IN => COMMENTS_IN,
           ERROR_NO => ERROR_NO,
           DBMS_ACTIVITY => DBMS_ACTIVITY,
           ERROR_MESSAGE => ERROR_MESSAGE
         );
         DBMS_OUTPUT.PUT_LINE('ERROR_NO = ' || ERROR_NO);
         DBMS_OUTPUT.PUT_LINE('DBMS_ACTIVITY = ' || DBMS_ACTIVITY);
         DBMS_OUTPUT.PUT_LINE('ERROR_MESSAGE = ' || ERROR_MESSAGE);
       END;
```

Chapter 3
Batch Processing

A batch process is a task that the system runs automatically. Such tasks can include making updates, purging obsolete data, generating schedules, or performing other automatic functions established by your organization without user intervention. Batch processing uses computer resources efficiently and facilitates off-hour completion of time consuming tasks that can slow down the system.

Job Manager and Job Manager Log modules

Batch processes are stored as database packages and database procedures, and are managed from within the system through the Job Manager module. Each Job Manager record has a system-generated number that uniquely identifies the "job", or scheduled database procedure. The batch job can call one or more procedures to run at the specified run-time interval.

The Job Manager Log lists a complete set of messages issued by procedures as they were run. Each run of a procedure is uniquely identified by the Job Sequence Number, allowing you to research Job Manager Logs for one or many Procedures, and each run of a Procedure.

Batch Processes

Job Manager - sdbp_job_error_log ((job_in, plant_in, sysdate-365);

Generates entries in the Job Manager Log.

Purge Data - sdbp_purge_data;

Purges (deletes) data from the system database tables as defined in the Batch Purge Parameters Business Rule. Currently Alerts and Batch Messages are reviewed, deleting only those records that are older than the number of days defined in the rule for that item.

Run All Batch - sdbp_run_all_batch;

Calls all of the batch procedures defined in the Batch Job Control business rule. SDBP_RUN_ALL_BATCH generally includes all procedures that need to run on a daily basis. Procedures that need to run at different periodic intervals, such as once a year or at the end of the Pay Period should be scheduled separately.

How to Create a New Batch Job

Use this action to set batch jobs to run on an individual schedule when you don't want to include them in Run All Batch.

1. **Open the Job Manager module.**
2. **Select Create Job from the Actions list on the Search Options screen.**
   The Update Job Window opens.

3. **Enter Job information.**
   In the description field, enter the database package or procedure name (as stored in the database), followed by the Job Number (NN) displayed in the upper left corner of the window and the Plant Code ('PC') that you want this Job to process, followed by a semi-colon.

   Packages: PACKAGE . PROCEDURE_NAME (NN, 'PC');

   Procedures: PROCEDURE_NAME (NN, 'PC');

   You can then enter a time Interval for automatic cycling of the procedure and the Next Run Date (and time) that you want the procedure to begin cycling.

4. **Click the Save icon when you have entered all necessary information.**

Batch Job Control Rule

Use the Batch Job Control business rule to include or exclude batch processes from regular batch cycles.

Batch Process, Job, and Option Status - Specific batch processes that can be turned on or turned off are listed in the Batch Process column. All batch procedures listed and turned on will run as part of SDBP_RUN_ALL_BATCH in the order specified in the Batch Process column.

Enter YES in the Option Status column if you want the batch process to run regularly, and NO if you do not want it to run. The Job column identifies the stored database procedure associated to each batch process.

Note that the order in which batch procedures are run makes a difference since a batch procedure may be dependent upon the data processed in another batch procedure. If they are run out of order, you may not see the data (such as cost roll-ups) for the batch run until the next time batch procedures are run.

Chapter 4
SAPI Library

The Application Programming Interface (SAPI) is used to customize Oracle Utilities Work and Asset Management Forms processing. An Oracle Forms library file called SAPI.PLL is included with the standard Oracle Utilities Work and Asset Management application and contains two Forms procedures:

- SAPI_PRODUCT_VERSION - used to keep track of changes to the library file.
- SAPI_TRIGGER - called by Oracle Utilities Work and Asset Management Forms standard processing to provide the ability to customize some Oracle Utilities Work and Asset Management processing.

SAPI_TRIGGER is called from a number of Oracle Utilities Work and Asset Management Form events. These are explained in the following table.

Triggers

The SAPI_TRIGGER procedure is called in each of the following forms events on every Form/ Block in the application:

Action	Trigger Name	Event	Special Parame-ters/Comments
New record	POST-INSERT	Triggered after a new record is created.	Any kind of post-commit processing of an inserted record would go here.
New record	PRE-INSERT	Triggered before a new record is created.	Record validation code and field defaults would normally go here.
Update record	POST-UPDATE	Triggered after an update to a record.	Any kind of post-commit processing of an updated record would go here.
Update record	PRE-UPDATE	Triggered before an update to a record.	Record validation code and field defaults would normally go here
Validation	WHEN-VALIDA TE-ITEM	Triggered when a field has been changed.	Specialized field level validation would go here.
Form Initializati on	PRE-FORM	First event triggered when a new forms starts up.	

Action	Trigger Name	Event	Special Parame-ters/Comments
Form Initialization	PRE-WHEN-NEW-FORM	Special Oracle Utilities Work and Asset Management event that is called before forms WHEN-NEW-FORM trigger.	Please contact Oracle Support before using this event.
Form Initialization	WHEN-NEW-FORM	Triggered when a new form is opened. This is called after PRE-FORM.	
Key	KEY-DELREC	The “Delete Record” key is clicked.	
Key	KEY-NEXT-ITEM	The “Next” button is clicked.	
Key	KEY-PREV-ITEM	The “Previous” button is clicked.	
Key	KEY-CREREC	The “Insert Record” key is clicked.	
Key	KEY-COMMIT	The Save icon is clicked.	
Navigation	WHEN-NEW-RECORD	Triggered when user navigates to a record.	
Navigation	WHEN-NEW-BLOCK	Triggered when a user navigates to a new block.	
Navigation	WHEN-NEW-ITEM-INSTANCE	Triggered when user navigates to a field.	
Navigation	POST-QUERY	After any block query, this trigger is fired to populate non-database fields	
Custom Menu Item	WHEN-CUSTOM-MENU-ACTION-<Action>	A custom menu action is selected.	”<Action>” is defined by text string entered in the “Action” field on the Module Custom Menus definition screen. See Custom Menus documentation for details.

Parameters

The parameters for SAPI_TRIGGER are listed below, together with some explanation of each. The “_IN” at the end of the parameter name means information coming in and the “_OUT” means data going to the calling event processor.

TRIGGER_NAME_IN: This is the name of the event trigger from the above list.

PLANT_IN: The current plant code that the user is logged into.

FORM_NAME_IN: Form name of the module (see Synergen Modules)

BLOCK_NAME_IN: Name of the block that the event occurred in.

TYPE_IN: Values are INSERT or UPDATE depending on the type of event trigger

GOTO_FIELD_OUT: This is used when the ERROR_MESSAGE_OUT is not null and it is desired to put the cursor on a particular field. Put the name of the field in this parameter.

TRACE_ACTIVITY_OUT: This text can be used to describe what actions are taking place for error handling. It will be displayed with the error information if an error occurs.

ERROR_MESSAGE_OUT: This text is set when an error has occurred. If this parameter is not null, then an error will be displayed and all processing for the event will stop.

Sample

A sample SAPI trigger is shown below. In this example, on the Work Order and Work Order Tasks forms, we want to add special processing on update of the header record status to FINISHED. If the UDF field 10 is filled in, we want to launch to a browser window to a special URL.

```
        PROCEDURE SAPI_TRIGGER (
         trigger_name_in varchar2,
         plant_in varchar2,
         form_name_in varchar2,
         block_name_in varchar2,
         type_in varchar2,
         goto_field_out in out varchar2,
         trace_activity_out in out varchar2,
         error_message_out in out varchar2) IS
        -- trigger_name_in varchar2(30) is the name of the trigger
        -- plant_in varchar2(3) is the plant
        -- form_name_in varchar2(8) is the name of the form
        -- block_name_in varchar2(30) is the name of the block
        -- type_in is 'INSERT' , 'UPDATE' etc depending on trigger
        -- goto_field_out varchar2 (61) is used when the error_message_out is
        --  not null and it is desired to put cursor on this item
        --  include block.item
        -- trace_activity_out varchar2(100) is used to trace what is being
done
        --  will be displayed with the error
        -- error_message_out varchar2(200) is the error message
        --  return null if everything is okay
        --  if not null, then the trigger is not done and the
        --  trace_activity_out and error_message_out are displayed
         v_old_status sa_work_order.work_status%type;
         v_current_status sa_work_order.work_status%type;
         v_url varchar2(2000);
        BEGIN
         error_message_out := null;
         goto_field_out := null;

         if form_name_in in ('WORKORD', 'WOTASK') then
         if block_name_in in ('HEADER') then
          if trigger_name_in = 'PRE-UPDATE' then
         v_old_status :=
          get_item_property('header.work_status',DATABASE_VALUE);
         copy('header.work_status', v_current_status );
         if v_current_status = 'FINISHED' and
          v_old_status != v_current_status and
           name_in('header.attribute10') is not null then
            select sia_url||'/WebGateWay' into v_url from sa_internet_
integration where rownum = 1 order by timestamp;
         web.show_document(v_url );
         end if;
          end if;
```

```
  end if;
 end if;

exception
 when others then
  error_message_out := substr(sqlerrm,1,200);
END;
```

Alerts

All base functions of Oracle Forms are supported in SAPI.

All Oracle Utilities Work and Asset Management base forms have the following set of defined alerts that you can reference.

Alert	Buttons	Default Message
COMMIT_DATA	Yes/No/Cancel	Do you want to save the changes you have made?
DUPLICATE_DATA	Return/Clear	A record with these key fields already exists!
GENERAL_STOP	Ok	
THREE_BUTTON_ WARNING	Ok/Cancel/Help	
YES_NO_WARNING	Yes/No/Help	
GENERAL_INFO	Ok	General Info
SAVE_CANCEL_WARNING	Save/Cancel/Help	

Chapter 5
SMU Transaction Processes

System Mobile Upload (SMU) transaction processes allow the Oracle Utilities Work and Asset Management mobile application to upload mobile-acquired data into the Oracle Utilities Work and Asset Management online system. The SDBP_INTERFACE interface was built for the Mobile application data to be uploaded into the system and was later extended to include Cost and Closeout data transactions. The interface allows the following types of record transactions:

- Activity Tracking
- Create Follow-Up Work Request
- Create Service Request
- Create Work Order
- Direct Charges
- PM Routing
- Service History
- Service Request Customer Billing
- Service Request Meter Deletion
- Service Request Meter Information
- Stock Checkout
- Task Progress
- Timekeeping

The interface is normally run as a batch job. Specification is as follows:

```
smu_interface.sdbp_mu_interface
(
      p_job_in,     -- job number generated in Job Manager
      p_plant_in,   -- plant code
      p_purge_in    -- Y to delete the record after successfully
                      processed
);
```

Processing of records is based on TRANSACTION_NO, TRANSACTION_TYPE, DEVICE_ID, SENT_TO_INTERFACE_IND, and READY_PROCESS_IND. TRANSACTION_TYPE values and invoked processing are described below.

The following conditions apply for this interface:

- The DEVICE_ID and READY_PROCESS_IND are set for Mobile Application only.
- Records are processed where SENT_TO_INTERFACE_IND is set to “N” or NULL.

- Processed records will have the SENT_TO_INTERFACE_IND set to “Y”.

Activity Tracking

The following conditions apply for this interface:

- Activity Tracking records are identified by a TRANSACTION_TYPE of “AT”.

This processing does the following:

- Creates Work Activity Tracking records in the system.

Required Fields

```
PLANT
AT_ACTIVITY
AT_ACTIVITY_AMOUNT
AT_ACTIVITY_UNITS
AT_ACTIVITY_DURATION
AT_ACTIVITY_COMMENTS
TRANSACTION_DATE
```

Optional Fields

```
CREATED_DATE
CREATED_BY
WORK_ORDER_NO
WORK_ORDER_TASK_NO
SERVICE_REQUEST_NO
AT_ASSET_RECORD_TYPE
AT_ASSET_ID
CREW
EMPLOYEE_NO
```

Create Follow-Up Work Request

The following conditions apply for this interface:

- Follow-up Work Requests can be generated for Task Progress records that are identified by a TRANSACTION_TYPE of “TP”.
- The TP_CREATE_WORK_REQUEST_IND field must be set to “Y” to generate a follow-up work request for a completed task.

This processing does the following:

- Generates a regular-type work request in PLANNING status based on the work order information of the completed task.
- Adds the generated work request number to the task finish information.

Business Rules

```
WORK REQUEST PROCESSING
```

Required Fields

```
PLANT
WORK_ORDER_NO
WORK_ORDER_TASK_NO
```

Optional Fields

The following fields will be used if populated for the generated work request and task finish information:

```
TP_FURTHER_ACTION_DESC
TP_FURTHER_ACTION_REQ_DATE
```

Create Service Request

The following conditions apply for this interface:

- Create Service Request records are identified by a TRANSACTION_TYPE of “SR”.
- If CI_STREET_NUMBER_CHAR is populated, this processing will attempt to convert it to a number to populate STREET_NUMBER field.

This processing does the following:

- Creates Service Request records from the Mobile application into the system.
- Updates the new service request number onto the SMU_TRANSACTION record.
- Activates the newly created service request, if business rule PRODUCT INTEGRATION CCB key name SERVICE REQUEST STATUS TRIGGER integration is turned ON.

Business Rules

```
PRODUCT INTEGRATION CCB
DEFAULT ACCTS FOR INTERFACES
```

Required Fields

```
PLANT
DEVICE_ID
SR_SERVICE_REQUEST_TYPE
SR_SCHEDULE_DATE
TRANSACTION_DATE
SR_SERVICE_POINT_ASSET_ID (required for CCB integration)
```

Optional Fields

```
SR_PROBLEM_POSTAL_CODE
SR_CANCELED_IND
```

Create Work Order

The following conditions apply for this interface:

- This interface was designed as Mobile only interface but can be called without DEVICE_ID filled in.
- Records are created with user = MOBILE.

Create Work Order records are identified by a TRANSACTION_TYPE of “WO”. This processing does the following:

- Creates Work Order header record in ACTIVE status.
- Creates associated Task record in Active status.
- Creates associated Work Order Task Failure record based on the single asset on the transaction record.
- Updates the new work order number onto the SMU_TRANSACTION record.

Business Rules

```
DEFAULT ACCTS FOR INTERFACES
```

Required Fields

```
PLANT
WO_TASK_REQUIRED_DATE
WO_ASSET_RECORD_TYPE
WO_ASSET_ID
WO_FAILURE_CODE
TRANSACTION_DATE
```

Optional Fields

```
WO_TASK_DESC
WO_DEFICIENCY_TAG
CREW
WO_WORK_CLASS
WO_WORK_CATEGORY
WO_TASK_PRIORITY
```

Direct Charges

The following conditions apply for this interface:

- Timekeeping records are identified by a TRANSACTION_TYPE of “OC”.
- Only Work Order or Service Request charges are supported.
- Employee number or Crew is required for this processing.

This processing does the following:

- Inserts direct charges records into the system based on employee/crew lead and date.

Required Fields

```
PLANT
CREATED_BY
TRANSACTION_DATE
OC_QUANTITY
OC_CHARGE_TYPE
```

Optional Fields

```
REFERENCE_NO
OC_STANDARD_PRICE
OC_UNITS
OC_VENDOR_CODE
```

PM Routing

The following condition applies for this interface:

- PM Routing records are identified by a TRANSACTION_TYPE of “PR”.

This processing does the following

- Creates PM Routing records by the Mobile application into the system.

Required Fields

```
PLANT
DEVICE_ID
PR_PM_ROUTE_NO
PR_SCHEDULE_DATE
PR_PM_ROUTE_STOP
PR_PM_STATUS
PR_COMMENTS
```

Service History

The following conditions apply for this interface:

- Service History records are identified by a TRANSACTION_TYPE of “SH”.
- A Work Order/Task or Service Request is required for this processing.

This processing does the following:

- Updates Service History detail information for Work Orders or activated Service Requests and purges any outstanding Task Progress records business Rules.

Required Fields

```
PLANT
CREATED_BY
SH_ATTRIBUTE_VALUE
SH_ATTRIBUTE_SEQUENCE_NO
SH_SPECIFICATION_CATEGORY
SH_SPECIFICATION_TYPE
TRANSACTION_DATE
```

Service Request Customer Billing

The following conditions apply for this interface:

- Service Request Customer Billing records are identified by a TRANSACTION_TYPE of “CI”.
- Service Request records cannot be updated using this processing when in CREATED or PENDING APPROVAL statuses
- If CI_STREET_NUMBER_CHAR is populated, this processing will attempt to convert it to a number to populate STREET_NUMBER field.

This processing does the following:

- Updates customer information.
- Inserts into Crew Log on activated Service Request records by the Mobile application into the system.

Business Rules

```
PRODUCT INTEGRATION CCB
```

Required Fields

```
PLANT
DEVICE_ID
SERVICE_REQUEST_NO
CI_CUSTOMER_ID
TRANSACTION_DATE
```

Optional Fields

```
CI_STREET_NUMBER_CHAR
CI_CUSTOMER_LAST_NAME
CI_CUSTOMER_FIRST_NAME
CI_COMPANY
CI_STREET_NAME
CI_STREET_DIRECTION
CI_SUITE
CI_POST_OFFICE_CODE
CI_CITY
CI_STATE_PROVINCE
```

```
CI_CUSTOMER_PHONE
CI_CUSTOMER_PHONE_EXT
CI_CUSTOMER_HOME_PHONE
CI_NUMBER_PREFIX
CI_NUMBER_SUFFIX
CI_POST_OFFICE_BOX
```

Service Request Meter Deletion

The following condition applies for this interface:

- Meter deletion records are identified by a TRANSACTION_TYPE of “MD”.

This processing does the following:

- Deletes Service Request meter information on activated Service Request records by the Mobile application into the system.

Required Fields

```
PLANT
DEVICE_ID
SERVICE_REQUEST_NO
MI_BADGE_NUMBER
```

Service Request Meter Information

The following conditions apply for this interface:

- Meter Information records are identified by a TRANSACTION_TYPE of “MI”.
- Meter information records are of two types: record with MI_REGISTER_ID set to NULL and those where it is populated (indicating master/detail records).
- Service Request Meter records cannot be updated using this processing when the Service Request is in CREATED or PENDING APPROVAL statuses.

This processing does the following:

- Updates Service Request meter information on activated Service Request records by the Mobile application into the system. Required Fields.

Required Fields

The following fields are required to have values for this processing when MI_REGISTER_ID is NULL:

```
PLANT
DEVICE_ID
SERVICE_REQUEST_NO
MI_BADGE_NUMBER
MI_METER_STATUS
MI_METER_READ_DATE
MI_DISCONNECT_LOCATION
TRANSACTION_DATE
```

The following fields are required to have values when MI_REGISTER_ID is NOT NULL:

```
PLANT
DEVICE_ID
SERVICE_REQUEST_NO
MI_REGISTER_ID
MI_BADGE_NUMBER
MI_REGISTER_READING
```

```
MI_REGISTER_READ_SEQ_NO
MI_REGISTER_UOM
MI_REGISTER_TOU
TRANSACTION_DATE
```

Optional Fields

```
MI_DISCONNECT_LOCATION
MI_CONFIG_TYPE
```

Stock Checkout

The following conditions apply for this interface:

- Stock Checkout records are identified by a TRANSACTION_TYPE of "SC".
- Work order/Task or Service Request is required for this processing.
- Employee number or Crew is required for this processing.

This processing does the following:

- Affects work order materials based on usage information and processes a work order/service request stock checkout.

Required Fields

```
PLANT
SC_QUANTITY
CREATED_BY
SC_REPORTED_USED_QTY
SC_STOCK_CODE
SC_STOREROOM
WORK_ORDER_NO
WORK_ORDER_TASK_NO
```

Optional Field

```
SC_STOCK_DESC
```

Task Progress

The following conditions apply for this interface:

- Task Progress records are identified by a TRANSACTION_TYPE of "TP".
- A Work Order/Task or Service Request is required for this processing.
- Tasks and Service Requests cannot be in CREATED/PLANNING or PENDING APPROVAL statuses.
- If TP_TYPE_OF_WORK_AMOUNT is populated, an Activity Tracking record is created. In this case, TP_TYPE_OF_WORK and TP_TYPE_OF_WORK_UNITS are required.

This processing does the following:

- Updates Work Order Task and Service Requests information.
- Inserts Task Failure information if appropriate.
- Inserts Activity Tracking if appropriate.

Business Rules

```
PRODUCT INTEGRATION CCB
```

Required Fields

```
PLANT
CREATED_BY
TP_WORK_STATUS
TRANSACTION_DATE
```

Optional Fields

```
TP_PHASE
TP_COMMENTS
TP_TYPE_OF_WORK_AMOUNT
TP_FAILURE_CODE
TP_FAILURE_MODE
TP_COMPONENT_CODE
TP_FURTHER_ACTION
TP_CREATE_WORK_REQUEST_IND
TP_FURTHER_ACTION_DESC
TP_FURTHER_ACTION_REQ_DATE
TP_REPAIR_CODE
```

Timekeeping

The following conditions apply for this interface:

- Employee must have an employee record in the system for the specific plant.
- Supervisor name is looked up from Code Table 310 if a Title is populated on Employee record.
- Pay period must exist for given CT_DATE of Timekeeping record.
- Craft code used is whatever is on the Timekeeping record or Employee record, in that order; one or the other is required.
- Timekeeping records are identified by a TRANSACTION_TYPE of "CT".
- Only Work Order or Service Request charge types are supported.

This processing does the following:

- Inserts timekeeping records into the system based on employee and date.

Business Rules

```
SHIFT DIFFERENTIAL RATES
LABOR COSTING WAGE RATES,
CRAFT RATES
TIMEKEEPING LABOR EARNING TYPE
```

Required Fields

A Work Order/Task or Service Request number
Regular or Premium hours/shift/type values

```
PLANT
CT_EMPLOYEE_NO
CT_DATE
CREATED_BY
```

SMU_TRANSACTIONS Table Layout

The SMU_TRANSACTIONS table layout is as follows:

Column Name	Data Type	Length	Description
TRANSACTION_NO	NUMBER		Record identifier
TRANSACTION_DATE	DATE		Record creation date
TRANSACTION_TYPE	VARCHAR2	2	See above for values
CREW	VARCHAR2	5	Used across several transaction types
PLANT	VARCHAR2	3	Used across all transaction types
WORK_ORDER_NO	VARCHAR2	7	Used across several transaction types
WORK_ORDER_TASK_NO	VARCHAR2	2	Used across several transaction types
CT_DATE	DATE		Timesheet date
CT_EMPLOYEE_NO	VARCHAR2	6	Timesheet Employee Number
CT_REGULAR_SHIFT	VARCHAR2	5	Timesheet Regular Shift code
CT_PREMIUM_SHIFT	VARCHAR2	5	Timesheet Premium Shift code
CT_REGULAR_TYPE	VARCHAR2	6	Timesheet Regular Type code
CT_PREMIUM_TYPE	VARCHAR2	6	Timesheet Premium Type code
CT_REGULAR_HOURS	NUMBER	10,2	Timesheet Regular hours
CT_PREMIUM_HOURS	NUMBER	10,2	Timesheet Premium hours
OC_CHARGE_TYPE	VARCHAR2	10	Direct Charges Charge Type
OC_STANDARD_PRICE	NUMBER	12,4	Direct Charges Standard Price
OC_QUANTITY	NUMBER	10,2	Direct Charges Quantity
SC_STOCK_CODE	VARCHAR2	15	Stock Checkout Stock code

Column Name	Data Type	Length	Description
SC_STOREROOM	VARCHAR2	3	Stock Checkout Storeroom
SC_QUANTITY	NUMBER	10,2	Stock Checkout quantity
SC_STOCK_DESC	VARCHAR2	200	Stock Checkout Stock description
TP_WORK_STATUS	VARCHAR2	20	Task Progress work status
TP_PHASE	VARCHAR2	30	Task Progress
TP_TYPE_OF_WORK_ AMOUNT	NUMBER		Task Progress
TP_TYPE_OF_WORK_UNITS	VARCHAR2	20	Task Progress
TP_TYPE_OF_WORK	VARCHAR2	20	Task Progress
TP_COMMENTS	VARCHAR2	2000	Task Progress
TP_FAILURE_CODE	VARCHAR2	10	Task Progress
TP_FAILURE_MODE	VARCHAR2	10	Task Progress
TP_REPAIR_CODE	VARCHAR2	10	Task Progress
TP_COMPONENT_CODE	VARCHAR2	10	Task Progress
TP_FURTHER_ACTION	VARCHAR2	10	Task Progress
WO_TASK_DESC	VARCHAR2	200	Create Work Order
WO_TASK_REQUIRED_DATE	DATE		Create Work Order
WO_TASK_PRIORITY	NUMBER	1,0	Create Work Order
WO_ASSET_RECORD_TYPE	VARCHAR2	1	Create Work Order
WO_ASSET_ID	VARCHAR2	15	Create Work Order
WO_FAILURE_CODE	VARCHAR2	10	Create Work Order
WO_DEFICIENCY_TAG	VARCHAR2	15	Create Work Order
SH_SPECIFICATION_TYPE	VARCHAR2	10	Service History
SH_SPECIFICATION_ CATEGORY	VARCHAR2	15	Service History
SH_ATTRIBUTE_ SEQUENCE_NO	NUMBER	4,0	Service History
SH_ATTRIBUTE_VALUE	VARCHAR2	50	Service History
SENT_TO_INTERFACE_IND	VARCHAR2	1	Y or N as per normal interface

Column Name	Data Type	Length	Description
EMPLOYEE_NO	VARCHAR2	6	Used across several transaction types
WO_WORK_CLASS	VARCHAR2	10	Create Work Order
WO_WORK_CATEGORY	VARCHAR2	10	Create Work Order
SMU_SEQ_NO	NUMBER	10,0	Unique identifier inside of an Transaction Number
CREATED_BY	VARCHAR2	20	Used across several transaction types
CREATED_DATE	DATE		Used across several transaction types
SERVICE_REQUEST_NO	VARCHAR2	7	Used across several transaction types
REFERENCE_NO	VARCHAR2	30	Direct Charges
DEVICE_ID	VARCHAR2	100	Mobile device number
OC_UNITS	VARCHAR2	10	Direct Charges
SC_REPORTED_USED_QTY	NUMBER	10,2	Stock Checkout
OC_VENDOR_CODE	VARCHAR2	30	Direct Charges
READY_PROCESS_IND	VARCHAR2	1	Y or N Used by Mobile Application
AT_ACTIVITY	VARCHAR2	20	Activity Tracking
AT_ACTIVITY_AMOUNT	NUMBER		Activity Tracking
AT_ACTIVITY_UNITS	VARCHAR2	20	Activity Tracking
AT_ACTIVITY_DURATION	NUMBER		Activity Tracking
AT_ASSET_RECORD_TYPE	VARCHAR2	1	Activity Tracking
AT_ASSET_ID	VARCHAR2	15	Activity Tracking
AT_ACTIVITY_COMMENTS	VARCHAR2	2000	Activity Tracking
SR_SERVICE_REQUEST_ TYPE	VARCHAR2	15	Create Service Request
SR_SCHEDULE_DATE	DATE		Create Service Request
SR_CUSTOMER_ID	VARCHAR2	20	Create Service Request
SR_CUSTOMER_ADDRESS_ ID	NUMBER		Create Service Request

Column Name	Data Type	Length	Description
SR_PROBLEM_LAST_NAME	VARCHAR2	30	Create Service Request
SR_PROBLEM_FIRST_NAME	VARCHAR2	30	Create Service Request
SR_PROBLEM_STREET_ NUMBER_CHAR	VARCHAR2	10	Create Service Request
SR_PROBLEM_STREET_ NAME	VARCHAR2	40	Create Service Request
SR_PROBLEM_STREET_ DIRECTION	VARCHAR2	3	Create Service Request
SR_PROBLEM_CITY	VARCHAR2	40	Create Service Request
SR_PROBLEM_STATE_ PROVINCE	VARCHAR2	4	Create Service Request
SR_PROBLEM_POSTAL_ CODE	VARCHAR2	10	Create Service Request
SR_PROBLEM_PHONE	VARCHAR2	30	Create Service Request
SR_PROBLEM_PHONE_EXT	VARCHAR2	5	Create Service Request
SR_PROBLEM_HOME_PHONE	VARCHAR2	30	Create Service Request
SR_PROBLEM_CODE	VARCHAR2	10	Create Service Request
SR_PROBLEM_ DESCRIPTION	VARCHAR2	200	Create Service Request
CI_CUSTOMER_ID	VARCHAR2	20	Service Request Customer Billing
CI_CUSTOMER_LAST_NAME	VARCHAR2	30	Service Request Customer Billing
CI_CUSTOMER_FIRST_ NAME	VARCHAR2	30	Service Request Customer Billing
CI_COMPANY	VARCHAR2	60	Service Request Customer Billing
CI_STREET_NUMBER_CHAR	VARCHAR2	10	Service Request Customer Billing
CI_STREET_NAME	VARCHAR2	40	Service Request Customer Billing

Column Name	Data Type	Length	Description
CI_STREET_DIRECTION	VARCHAR2	3	Service Request Customer Billing
CI_SUITE	VARCHAR2	10	Service Request Customer Billing
CI_POST_OFFICE_BOX	VARCHAR2	10	Service Request Customer Billing
CI_CITY	VARCHAR2	40	Service Request Customer Billing
CI_STATE_PROVINCE	VARCHAR2	4	Service Request Customer Billing
CI_POSTAL_CODE	VARCHAR2	15	Service Request Customer Billing
CI_CUSTOMER_PHONE	VARCHAR2	30	Service Request Customer Billing
CI_CUSTOMER_PHONE_EXT	VARCHAR2	5	Service Request Customer Billing
CI_CUSTOMER_HOME_ PHONE	VARCHAR2	30	Service Request Customer Billing
MI_BADGE_NUMBER	VARCHAR2	30	Service Request Meter Info and Meter Deletion
MI_METER_STATUS	VARCHAR2	3	Service Request Meter Info
MI_METER_READ_DATE	DATE		Service Request Meter Info
MI_REGISTER_ID	VARCHAR2	10	Service Request Meter Info
MI_REGISTER_READING	NUMBER	15,6	Service Request Meter Info
SR_SERVICE_POINT_ ASSET_ID	VARCHAR2	15	Create Service Request
SR_CANCELED_IND	VARCHAR2	1	Create Service Request
MI_METER_CONFIG_TYPE	VARCHAR2	12	Service Request Meter Info
MI_DISCONNECT_ LOCATION	VARCHAR2	4	Service Request Meter Info
MI_REGISTER_READ_SEQ_ NO	NUMBER	2,0	Service Request Meter Info

Column Name	Data Type	Length	Description
MI_REGISTER_UOM	VARCHAR2	4	Service Request Meter Info
MI_REGISTER_TOU	VARCHAR2	8	Service Request Meter Info
PR_PM_ROUTE_NO	VARCHAR2	10	PM Routing
PR_SCHEDULE_DATE	DATE		PM Routing
PR_PM_ROUTE_STOP	NUMBER		PM Routing
PR_PM_STATUS	VARCHAR2	20	PM Routing
PR_COMMENTS	VARCHAR2	2000	PM Routing
CT_CRAFT	VARCHAR2	5	Timekeeping craft to override employee info
TP_FURTHER_ACTION_ DESC	VARCHAR2	2000	Task Progress
TP_FURTHER_ACTION_ REQ_DATE	DATE		Task Progress
TP_CREATE_WORK_ REQUEST_IND	VARCHAR2	1	Task Progress

Chapter 6
Custom Forms

Clients may want to enroll a new form in the Oracle Utilities Work and Asset Management system that is not tied to another form via a custom menu. This can be accomplished using a "backend" methodology so that a new form may be set to appear in the Application Map.

Since custom forms are not tied to any other entity in the system, they cannot appear on the Forms menus or the Go menu.

Follow the steps indicated below to enroll a new form using this method: Insert row into sa_ form_names. Form Description, Form_name, form_function, Subsystem, and locale_code are required.

```
<?xml version='1.0'  encoding='Cp1252' ?>
<RESULTS>
  <ROW>
   <COLUMN NAME="FORM_DESCRIPTION"><![CDATA[My Test]]></COLUMN>
   <COLUMN NAME="FORM_NAME"><![CDATA[TESTFORM]]></COLUMN>
   <COLUMN NAME="FORM_FUNCTION"><![CDATA[Q]]></COLUMN>
      <COLUMN NAME="PRIMEKEY_NAME_1"><![CDATA[]]></COLUMN>
      <COLUMN NAME="PRIMEKEY_NAME_2"><![CDATA[]]></COLUMN>
      <COLUMN NAME="PRIMEKEY_NAME_3"><![CDATA[]]></COLUMN>
      <COLUMN NAME="PRIMEKEY_NAME_4"><![CDATA[]]></COLUMN>
      <COLUMN NAME="SUBSYSTEM"><![CDATA[RESOURCE]]></COLUMN>
      <COLUMN NAME="REGENERATE_IND"><![CDATA[N]]></COLUMN>
      <COLUMN NAME="RELEASE_VERSION"><![CDATA[]]></COLUMN>
      <COLUMN NAME="STATUS_REQUIRED_IND"><![CDATA[N]]></COLUMN>
      <COLUMN NAME="SIA_MODULE"><![CDATA[]]></COLUMN>
      <COLUMN NAME="ARCHITECTURE_IND"><![CDATA[]]></COLUMN>
      <COLUMN NAME="LOCALE_CODE"><![CDATA[1]]></COLUMN>
      <COLUMN NAME="CREATED_DATE"><![CDATA[07-FEB-11]]></COLUMN>
      <COLUMN NAME="CREATED_BY"><![CDATA[PCALDWELL]]></COLUMN>
      <COLUMN NAME="LAST_UPDATE_DATE"><![CDATA[07-FEB-11]]></COLUMN>
      <COLUMN NAME="LAST_UPDATE_USER"><![CDATA[PCALDWELL]]></COLUMN>
      <COLUMN NAME="FORM_DESCRIPTION_TN"><![CDATA[]]></COLUMN>
     </ROW>
</RESULTS>
```

1. **Open the template form and drag the parameters object group onto your form.**
   template form: rpttmpl8.fmb,

   object group: PARAMETERS

   If you do not start with the template form, your custom form will not include all of the parameters needed to access the form from the Application Map and you may encounter errors. It is advised that clients start with the template for to achieve the best results. Ensure that the Parameters listed in the section below are present if you choose to create a custom form without the template.

2. **Insert row into sa_security_functions. Activity_id, activity_type, activity_group, menu_item, form_name, activity_name, activity_desc, configuration_flag, and locale_code are required.**

```
<?xml version='1.0'  encoding='Cp1252' ?>
<RESULTS>
  <ROW>
     <COLUMN NAME="ACTIVITY_ID"><![CDATA[TEST FORM]]></COLUMN>
     <COLUMN NAME="ACTIVITY_TYPE"><![CDATA[MENU]]></COLUMN>
     <COLUMN NAME="ACTIVITY_GROUP"><![CDATA[RESOURCE]]></COLUMN>
     <COLUMN NAME="MENU_ITEM"><![CDATA[TEST.TESTFORMS]]></COLUMN>
     <COLUMN NAME="FORM_NAME"><![CDATA[TESTFORM]]></COLUMN>
     <COLUMN NAME="ACTIVITY_NAME"><![CDATA[]]></COLUMN>
     <COLUMN NAME="ACTIVITY_DESC"><![CDATA[My Test]]></COLUMN>
     <COLUMN NAME="LAST_UPDATE_DATE"><![CDATA[07-FEB-11]]></COLUMN>
     <COLUMN NAME="LAST_UPDATE_USER"><![CDATA[SYNERGEN]]></COLUMN>
     <COLUMN NAME="CREATED_DATE"><![CDATA[07-FEB-11]]></COLUMN>
     <COLUMN NAME="CREATED_BY"><![CDATA[SYNERGEN]]></COLUMN>
     <COLUMN NAME="CONFIGURATION_FLAG"><![CDATA[S]]></COLUMN>
     <COLUMN NAME="LOCALE_CODE"><![CDATA[1]]></COLUMN>
     <COLUMN NAME="ACTIVITY_DESC_TN"><![CDATA[]]></COLUMN>
     <COLUMN NAME="ACTIVITY_NAME_TN"><![CDATA[]]></COLUMN>
     <COLUMN NAME="ACTIVITY_GROUP_DESC"><![CDATA[Resource]]></COLUMN>
     <COLUMN NAME="ACTIVITY_GROUP_DESC_TN"><![CDATA[]]></COLUMN>
 </ROW>
</RESULTS>
```

3. **Add new module to user's responsibility**
   This will allow a form to be called from SIA app map or home page. It will not allow form to be called from forms menu since menus are not dynamic.

Parameters

These parameters must be included on all forms within the application.

Name	APPROVAL_BLOCK
Parameter Data Type	Char
Parameter Initial Value	APPROVALS

Name	APPROVAL_COUNT
Comments	No. of records in Approval block where comment is not null.
Parameter Data Type	Number
Parameter Initial Value	0

Name	APPROVAL_BOOLEAN
Parameter Data Type	Char
Parameter Initial Value	FALSE

Name	DO_APPROVALS_QUERY
Parameter Data Type	Char

Parameter Initial Value	OFF

Name	B_LOV_ITEM
Parameter Data Type	Char
Parameter Initial Value	--

Name	B_LOV_ITEM_WIDTH
Parameter Data Type	Number
Parameter Initial Value	--

Name	PRIMEKEY_COUNT
Parameter Data Type	Number
Parameter Initial Value	--

Name	ACTION
Parameter Data Type	Char
Parameter Initial Value	--

Name	CURRENT_GROUP_NAME_VIEWS
Parameter Data Type	Char
Parameter Initial Value	--

Name	CURRENT_GROUP_NAME_ACTIONS
Parameter Data Type	Char
Parameter Initial Value	--

Name	DO_NOTES_QUERY
Parameter Data Type	Char
Parameter Initial Value	OFF

Name	NOTES_BLOCK
Parameter Data Type	Char
Parameter Initial Value	NOTES

Name	MASTER_VERSION
Parameter Data Type	Char
Parameter Initial Value	19-23

Name	BASE_TABLE_NAME
Parameter Data Type	Char
Parameter Initial Value	--

Name	ASK_COMMIT
Parameter Data Type	Char
Parameter Initial Value	Y

Name	COMMIT_IT
Parameter Data Type	Char
Parameter Initial Value	N

Name	CANCEL_BLOCK
Parameter Data Type	Char
Parameter Initial Value	--

Name	SAVE_RECORD
Parameter Data Type	Number
Parameter Initial Value	--

Name	COLUMN_START_POSITION
Parameter Data Type	Number
Parameter Initial Value	--

Name	ORDER_BY_COLUMN
Parameter Data Type	Char
Parameter Initial Value	plant

Name	PRE_DIALOG_CURSOR_ITEM
Parameter Data Type	Char
Parameter Initial Value	--

Name	MOVABLE_COLUMNS_ITEM
Parameter Data Type	Char
Parameter Initial Value	--

Name	CURSOR_ITEM
Parameter Data Type	Char
Parameter Initial Value	--
Name	CURSOR_X_POS
Parameter Data Type	Number
Parameter Initial Value	--
Name	PROCESS_BLOCK
Parameter Data Type	Char
Parameter Initial Value	SELECTION
Name	HEADER_PRIVS
Parameter Data Type	Char
Parameter Initial Value	N
Name	INSERT_FROM_SELECTION
Parameter Data Type	Char
Parameter Initial Value	N
Name	NOTES_PRIVS
Parameter Data Type	Char
Parameter Initial Value	N
Name	COSTS_PRIVS
Parameter Data Type	Char
Parameter Initial Value	N
Name	CLEAR_RESULTS_RECORD
Parameter Data Type	Char
Parameter Initial Value	N
Name	RESULTS_RECORDS
Parameter Data Type	Char
Parameter Initial Value	N
Name	PRIMEKEY_NAME_1

Parameter Data Type	Char
Parameter Initial Value	--

Name	PRIMEKEY_NAME_2
Parameter Data Type	Char
Parameter Initial Value	--

Name	PRIMEKEY_NAME_3
Parameter Data Type	Char
Parameter Initial Value	--

Name	PRIMEKEY_NAME_4
Parameter Data Type	Char
Parameter Initial Value	--

Name	PRIMEKEY_1
Parameter Data Type	Char
Parameter Initial Value	--

Name	PRIMEKEY_2
Parameter Data Type	Char
Parameter Initial Value	--

Name	PRIMEKEY_3
Parameter Data Type	Char
Parameter Initial Value	--

Name	PRIMEKEY_4
Parameter Data Type	Char
Parameter Initial Value	--

Name	QUERY_STRING
Parameter Data Type	Char
Parameter Initial Value	--

Name	FIRST_RESULTS_BLOCK
Parameter Data Type	Char

Parameter Initial Value	RESULTS
Name	FIRST_HEADER_BLOCK
Parameter Data Type	Char
Parameter Initial Value	HEADER
Name	RESULTS_BLOCK
Parameter Data Type	Char
Parameter Initial Value	RESULTS
Name	HEADER_BLOCK
Parameter Data Type	Char
Parameter Initial Value	HEADER
Name	SELECTION_BLOCK
Parameter Data Type	Char
Parameter Initial Value	SELECTION
Name	ATTACHMENT_BLOCK
Parameter Data Type	Char
Parameter Initial Value	ATTACHMENT
Name	DO_ATTACHMENTS_QUERY
Parameter Data Type	Char
Parameter Initial Value	OFF
Name	PREVIOUS_FORM
Parameter Data Type	Char
Parameter Initial Value	--
Name	SET_FOCUS
Comments	This is used to turn off saip_set_focus when called from WHEN-VALIDATE-ITEM trigger which uses alerts and then activates the window afterwards.
Parameter Data Type	Char
Parameter Initial Value	YES

Name	USE_WFA_WHERE_CLAUSE
Parameter Data Type	Char
Parameter Initial Value	FALSE
Name	WFA_WHERE_CLAUSE
Parameter Data Type	Char
Parameter Initial Value	--
Name	WFA_ORDER_BY
Parameter Data Type	Char
Parameter Initial Value	--
Name	INITIAL_MODE
Parameter Data Type	Char
Parameter Initial Value	SELECT
Name	FIRST_BLOCK
Parameter Data Type	Char
Parameter Initial Value	SELECTION
Name	REPORT_ID
Parameter Data Type	Char
Parameter Initial Value	--
Name	SELECTION_FILTER
Parameter Data Type	Char
Parameter Initial Value	--
Name	BLOCK_STATUS
Parameter Data Type	Char
Parameter Initial Value	NEW_STATUS
Name	CUSTOM_QUERY
Parameter Data Type	Char
Parameter Initial Value	--
Name	WORKFLOW_AGENT_TYPE

Parameter Data Type	Char
Parameter Initial Value	--
Name	TRIGGER_NAME
Parameter Data Type	Char
Parameter Initial Value	--
Name	PLANT
Parameter Data Type	Char
Parameter Initial Value	--
Name	DEBUG_PJC
Parameter Data Type	Char
Parameter Initial Value	--

Calendar Tool on Custom Forms

The calendar tool is defined by the calendar.fmx form. This form has three input parameters and one output global variable is populated on return. The snippet of code below shows how to call the form using information from a current field. The global variable populated is called "date_from_calendar".

.

```
add_parameter(list_id,'input_date', TEXT_PARAMETER,
    name_in(:system.cursor_item));
add_parameter(list_id,'input_type', TEXT_PARAMETER,
    get_item_property(:system.cursor_item,datatype));
add_parameter(list_id,'input_format_mask', TEXT_PARAMETER,
    get_item_property(:system.cursor_item,format_mask));

copy('N','parameter.ask_commit');
open_form('calendar', ACTIVATE, NO_SESSION, list_id);
copy('Y','parameter.ask_commit');
```

.

Chapter 7
Custom Business Rules

Custom Business Rules may be created at any time using the Business Rule module. This process is very simple and can be used to support any custom process you are building that requires configuration information.

The only simple rule to follow is to make sure that the type field highlighted below is set to **Custom**. If this field is set to Custom, then the upgrade process will ignore the rule during each software distribution upgrade.

There is no naming convention used by the Oracle Utilities Work and Asset Management team, so you may want to create a prefix that you use in naming the rule. Although unlikely, it is possible that a new rule will be created as part of the base application that is exactly the same as your rule name. This would cause you rule to be replaced during an upgrade. A standard prefix will avoid this possible situation.

Chapter 8

Custom Reports

If your organization requires a report that is not already provided, you can add a custom report. You can create custom reports using Oracle Report Builder tools or BI Publisher Reports.

Refer to sections on Custom Menus and Custom Menu Actions for information on how to make custom reports more easily accessible for users after they are created.

Related Topics

Converting an Oracle Report to BI Publisher

Naming Conventions

All Report IDs that begin with "S_" are reports built and maintained by Oracle Utilities Work and Asset Management.

Reports

Standard reports are named with the following format: S_RPTxxx.

Custom reports are named with the following format: C_abcxxx.

Example:
Report 35 is named *S_RPT035.rdf.*
Report 1 for Mike is named *C_MIK001.rdf.*

Views

Each report has an associated view that is named the same as the report. Views are named with the prefix "sv_" for standard reports and "cv_" for custom reports.

Example:
The view for standard report 35 is named *SV_S_RPT035*.
The view for custom report 1 for Mike is named *CV_C_MIK001*.

Creating a New Custom Report

Creating a new custom report involves the following steps:

1. Find the Information for the Report
2. Create a View for the Report
3. Build the Report
4. Build Selection Screens for the Report
5. Make the Report Available to Users

Find the Information for the Report

Before you start designing or building a custom report, it is important to locate and identify the information needed in the report. For example, for a report related to assets, you would need to know the Asset ID, Plant, Account numbers and so on. Furthermore, you need to determine which tables and columns are needed for the report.

You might want to identify an existing report as a starting point. From here you can copy pertinent information or the view needed to build the report. Beyond using an existing report, there are a few sources that can be used to find where the information that you need to build the report lies in the database.

Locate Information Using various Database Resources

Oracle Utilities Work and Asset Management

About Item - If you know where data is entered in Oracle Utilities Work and Asset Management, navigate to the field and select "About Item" under the "Help" menu. For text items, this should give you the column and table name of the item (if there is no table name listed, it means the field is not a "base table item", the field itself exists in another table and is just referenced for the current form).

Oracle Data Dictionary - Select Oracle Data Dictionary from the Admin menu. This will let you search by table, column, and column type. Thus, you can search for something like %ASSET% for all tables that contain the word "Asset". (The % is the recognized wildcard character.)

SQL

The layout of all of the Oracle Utilities Work and Asset Management Tables and Views can be accessed through the Schema Browser window.

Query behind the View

To find the query that a view is based on, use the following code:

```
SELECT TEXT
  FROM USER_VIEWS
 WHERE VIEW_NAME = 'VIEW_NAME'
```

If you do not own the view, try

```
SELECT TEXT
  FROM SYS.ALL_VIEWS
 WHERE VIEW_NAME = 'VIEW_NAME'
   AND OWNER = 'OWNER'
```

Create Report Executables

To add a custom report using Oracle Report Builder, you first need to create the actual report executable (".RDF" or ".REP") files in Oracle. You can either modify existing report executable files (located in the Oracle Utilities Work and Asset Management directory) or create new ones. If you modify existing Oracle Utilities Work and Asset Management files, first duplicate and rename the files to ensure that future updates will not overwrite your modifications.

Create a View for the Report

All selection criteria must be in the view. The view should be made so that order by functions will work. For example, if all the information is stored in the view and grouping occurs in the report, the grouping function will override the order by function.

An example of a script for creating the view follows. When creating custom views use the temblordb database and sign on with the username and password for the company.

```
create or replace view cv_c_splxxx as
 (select asset_id, attribute1
  from sa_asset)
/
```

Privileges and synonyms must also be granted and created. If you do not do this, you will not be able to test the final version of your report in Oracle Utilities Work and Asset Management.

```
CREATE PUBLIC SYNONYM CV_C_RPT000 FOR OWNER.SV_S_000SPL;
GRANT SELECT ON CV_C_SPL000 TO SUPERUSER;
```

Additional Notes

Using an existing view with this new report is not recommended. The view for the older report might change and disable your new report.

Build the Report

This section provides details about building the report in Oracle Report Builder however, some of the tips and hints can also be applied when building in, however, some of the tips and hints can also be applied when building in BI Publisher Reports.

Regardless of which report building tool you use, the same basic steps are required:

1. Create the Report File
2. Design the Data Model
3. Design the Layout
   It is best to make sure that your data model is complete before creating your layout model.

Create the Report File

If a portrait layout is required, use s_rpt000.rdf as a template by copying it to the name of your new report. If a landscape layout is required use s_lan000.rdf. By using these templates, all the fonts will be set up correctly and the report will look similar to existing reports. The reports also have triggers written to work with the report selection form.

Design the Data Model

1. Use the Data Model tool in Oracle Report Builder to create the query of the report from your view.
   This main query should follow several constraints:

   - The query should not *select* *. The columns to be included should be listed, rather than using the * wildcard. This is important because if any additional items are added into the view at a later time, the report may no longer work.

   - The main query should not join any other tables with the view for the report. This is important because it will interfere with the Oracle Utilities Work and Asset Management AI (described in number 3 below) for the reports will interfere with the table join.

   - There should be no *where* or *order by* clause. These will be replaced with &USER_WHERE_CLAUSE and &USER_ORDER_BY.

   An example of a valid query is shown in Query 1, which queries all assets and their components.

   ```
   select PLANT, ASSET_RECORD_TYPE, ASSET_ID, ASSET_DESC,
          PROCESS_NO, ASSET_TYPE, ASSET_CLASS, BUILDING,
          POSITION, LOCATION, COMPONENT_ID, COMPONENT_ID_DESC,
          MANUFACTURER_CODE, MANUFACTURER_DESC,
    MANUFACTURER_MODEL_NO, SERIAL_NO
     from SV_S_RPT089
   &USER_WHERE_CLAUSE
   &USER_ORDERY_BY
   ```

```
Query 1: Q_Asset
```

- A group is used for grouping objects in a query. If the columns you chose have grouping, you can click and drag the fields outside of the group box. For instance, a query might have a natural grouping of components by asset. That is, each asset can have multiple components. Thus, we can move the component information into a component group.

  Note: When the main query is grouped, the order by is set by the data model and will override the selection screen order by.

- If you need additional information in the report that you cannot create in the view, you can create more queries and link them to the main query. This functionality comes in handy if a table must be selected from more than once (limitations in creating a view).
- It is also possible to add a query in a column formula. This is done by selecting the *Formula Column* tool and adding a procedure like the following:

```
begin
   select column_name
       into variable
       from table_name
       where table_name.plant = :plant
         and other_join_condition
    return variable;
exception
    when others then return null;
end;
```

Design the Layout

If you are not completely familiar with the Oracle Report Builder, use the report wizard to do the initial layout. You'll save yourself a lot of time and work. Even you already have a layout and do not know why it does not lay out correctly, it may a good idea to reformat it using the layout wizard.

- If you need a landscape orientation, change the width and height under the properties of the report.
- Repeating frames must have a group associated with them. You can have repeating frames within repeating frames. For example, you can have one parent repeating frame whose source is G_Asset. Within this repeating frame, would be other repeating frames that hold G_Component, G_Process, and G_Building.
- Use normal frames to enclose "similar" objects. Repeating frames will often overwrite (as in lie on top of) other repeating frames, especially if the text fields are allowed to expand. If you place the repeating frame that is being overwritten in a normal frame, the repeating frame will then start printing under it instead of on top of it.
- Show child groups and their headings only when they exist.   For example, if we are displaying a piece of equipment and it's components, we do not want to show the headings for the components when the asset has no components. To do this, we need to create a counter and a format trigger.

  You can use a "counter field" in the main group within the Data Model to count the number of components that the asset has.

  You can use a "format trigger" to control whether the object prints if the count is zero. This will be a PL/SQL statement that tells the object to print itself only if there are fields to be printed. The format trigger is found on the property palette of almost every item and is indicated as a small "P" in the Layout Editor of the Object Navigator.

Create the format trigger on the frame that encloses the headings and the group's repeating frames.

The PL/SQL code for the format trigger follows:

```
begin
  if :cs_component_counter = 0 then
    return (FALSE);
  else
    return (TRUE);
  end if;
end;
```

Build Selection Screens for the Report

In order to use the custom report you must also create a custom selection screen to access the report. Creation of the selection screen requires the following steps:

1. Set up the Environment
2. Build a Custom Selection Screen
3. Create/Add a Custom Selection Screen to the Custom Report

Set up the Environment

In order to develop the report selection screens, you will need the following:

1. Oracle Form Builder
2. Oracle Utilities Work and Asset Management files:
   Libraries

   SRPI.PLL- Library of common procedures for reports

   Object Master for Inheritance

   RPTMSTR.FMB- Master form for referencing common objects

   Report Selection Screen Forms

   RPTTMPL8.FMB- Template for custom selection screens for reports

Note: Make sure that your custom form, srpi.pll, rptmstr.fmb, and rpttmpl8.fmb are all in the same directory as you are in development. Create a shortcut for the Forms builder that has a "start in" path to the same directory. Connect to the database in Forms Builder before compiling /generating a form.

Build a Custom Selection Screen

Oracle Utilities Work and Asset Management provides the following modules to build custom selection screens for your custom reports:

RPTMSTR.FMB - Master form for referencing common objects

RPTTMPL8.FMB - a template form for adding custom reports. This form should be copied to a new name (preferably beginning with the letters 'RPT' and modified to include custom reports. The new form must be added to code table 95 (without the file extension), so that it can be referenced in the Location field in the Report Administration module.

SRPI.PLL - Library of common procedures for reports

Exact Search vs. Wildcard Search

To allow search values to be entered as an exact string (versus a string requiring a wildcard or that returns non-exact values), rename the columns on the selection screen to have a prefix as defined below:

From the srpi_prepare_where_clause comments:

```
    --  create dynamic where clause
    --  from items on a selection menu
    ---- Non base table columns should start with 'NONSEARCH'

    -- The where clause will put in a phrase with
--          LIKE 'abc%'
    -- with abc replaced by the entered column value

    -- Null column values will be skipped

    -- Relational properties can be used by putting the prefix
    -- in front of the database column name
-- 'EQ_', 'LE_', 'GE_'

    -- Relational properties can also be defined by putting the
    -- prefix 'OPER_' in front of the database column name
    -- The list item value will then be used as the relation
    -- so that the user can set which relation will be used with the
    -- column

    -- Non database column 'custom_text' in the block
    -- appends the value of the custom_text column in the variables
```

Create/Add a Custom Selection Screen to the Custom Report

To add the custom selection screen to the custom report you must connect to the oracle database and then complete the following:

Add New Objects for the Custom Selection Screen

1. Create a window template with respect to the following:
   - General – Name. Naming convention of a window is RPT_{report name}, where {report_name} is the file name of the report.
   - Functional – Title
   - Functional – Primary Canvas: will need to come back to this attribute after setting up your new canvas.
2. Create record groups with respect to the following:
   - General – Name. Naming convention being database field_name + numeric portion of the report name, i.e. DEPARTMENT_001. Proper naming of fields will pay off when

you start adding multiple reports to the same report selection form (at a glance, you will be recognize records associated with a report).

- Functional - Record Group Query – your select statement goes here.

3. Create the appropriate lists of values and record groups for items that need them.
   You can copy already existing lists of values to save some time but you must make sure that there is an individual list of values for every item that needs one. Do not share lists of values or record groups. Change the following:
   - General – Name. Naming convention being database field_name + numeric portion of the report name.
   - Functional - Title
   - Functional - Record Group, map to appropriate record from step 2
   - Functional – Column Mapping Properties: will need to come back to this item after step 5
4. Create a canvas for your items to display on with respect to the following:
   - General – Name. Naming convention of a window is RPT_{report name}, where {report_name} is the file name of the report.
   - Physical – Window, map to values created in step 1
   - Now that you have a canvas, go back to Window (step 1) and assign value to Function - Primary Canvas.
5. Create the appropriate datablocks.
   Naming convention of a block is RPT_{report name}, where {report_name} is the file name of the report. Each block is a selection menu for a report.

   Each report must have its own block in the selection form. The reports template has a block called REPORT_TEMPLATE from which you can copy common objects and properties.

   For all items, change the following:
   - General - Name (use same name from your view)
   - List of Values (LOV) – List of Values, Map to value from step 3
   - Physical – Canvas, Map to value from step 4
   - Open the canvas (F2) and position the items.
   - Make sure 'ORDER_BY_TEXT', 'B_RUN_REPORT' and all other items map to the correct canvas.
6. Go back to the list of values and change Functional - column mapping properties to reflect the appropriate datablock items.
7. Complete a full program compile, build and executable, disconnect and save the form when you are done. It is very important to disconnect before saving as the form can become corrupted over time if you don't.

Make the Report Available to Users

Once you have created the appropriate report executable files in Oracle, you need to make the custom report accessible within Oracle Utilities Work and Asset Management. Adding a custom report to the system is a three-step process.

1. Add Custom Selection Form to Code Table 95
2. Add the Custom Report to the Reports List

3. Add the Custom Report to Responsibilities

You can also add a custom menu item to provide easy access to open the report.

Add Custom Selection Form to Code Table 95

Insert the form as a new record in the code table. This is only done when you install the first custom report using this selection form.

Add the Custom Report to the Reports List

1. Open the Report Administration module.
2. Click New.
3. Enter a Report ID.
4. Enter the Selection Block.
   The Selection Block is the report block name (i.e. RPT_{report name}).
5. Select the Location from the list of values.
   This list is controlled by Code Table 95. The Location is the form name of the report (without the extension).

   You may first need to add the appropriate form name to Code Table 95 of the Code Table and Codes module to make it available to the list.
6. Enter Type, Group, and Description.
7. Select the Restrict from Upgrade check box if needed.
   This prevents the record from being modified by subsequent releases.
8. Click Save.
   The system saves the Report Administration record and adds the custom report to the reports list. To have access to the report, you need to add the Custom Report to a Responsibility.

Add the Custom Report to Responsibilities

Anyone with the appropriate responsibility can then run the custom report.

BI Publisher Reports

In addition to calling standard Oracle reports, the system can also be configured to call BI Publisher Reports.

Certified Version

This version of Oracle Utilities Work and Asset Management is certified on:

- BI Publisher v10.1.3.4 for Oracle Utilities Work and Asset Management releases **1.9.0.1 and above**.
- Ensure that JDBC Driver for Oracle Application Server is version 10.2.0.4.0 or above is in used for releases 1.9.0.2 and above.

Note: BI Publisher Reports must be licensed separately from Oracle Utilities Work and Asset Management.

Prerequisite Software

- Java JDK 1.5.x or greater is required for BI Publisher Conversion Tools.
- JDBC Driver 10.2.0.4.0 or above for releases 1.9.0.2 and above.

Getting Started

Complete the following tasks to configure Oracle Utilities Work and Asset Management to call a BI Publisher Report

1. Install and configure the Oracle BI Publisher software.
   Refer to Oracle Business Intelligence Publisher Installation Guide, Release 10.1.3.4 or 11g for installation instructions.

   During the installation:

   - Make a note of the name and path of the directory where the application is installed.
   - Select the "Basic" installation type.
2. Set the WEB CONFIGURATION business rule to indicate the BI Publisher shared folder location and a maximum time-out for accessing the report.
   The shared folder indicates the folder in BI Publisher that contains all of the Oracle Utilities Work and Asset Management reports.

3. Set the Web Services Gateway business rule to indicate the admin username and password used to access BI Publisher, the service URL, and the Gateway URL for the reports.
   - Dataset ID and Consumer Class are pre-configured and should not be modified.
   - SERVICE URL = http://<BI Publisher server name>:<port>/xmlpserver/services/PublicReportService
   - GATEWAY URL = http://<application server>:<port>/<your environment>/synergen/WebGateWay
   - REPORTS CUSTOM VIRT DIRECTORY = http://[server]/reports/rwservlet
   - BI 11G = http://[server]/xmlpserver/
   - BI PUBLISHER RUN REPORT =
     Example: Dataset ID = 11BIPublisherDataset
     Service URL = http://[server]/xmlpserver/services/PublicReportService
     Gateway URL = http://[server]/synergen/WebGateWay"

Where <your environment> is the OC4J instance name.

Use the IP address of the BI Publisher server and the application server.

4. Create reports in BI Publisher.
   You can use an existing Oracle Report and report ID to create a BI Publisher version of an existing Oracle Report, or you can create a new custom report. If you create a custom report, be sure to add a record in the Reports Administration module which references the report ID and additional information for the report.
5. Open the Reports Administration module and navigate to the Report ID for the report.
6. Set the Report Service option to BI Publisher.
   You can also determine modules where the report appears on the actions list by listing the report in the Module Reports view of the Modules Administration - Forms module.

Configuring Oracle BI Publisher

To prepare Oracle BI Publisher for use with Oracle Utilities Work and Asset Management, you must complete three essential steps:

- Create User for Reporting Framework
- Set Report Path - set a path to the directory where the report template files are stored on the server.
- Create Data Source for Oracle Utilities Data Repository - create a JDBC data source on Oracle BI Publisher to allow a connection to the Oracle Utilities Data Repository. This is used when designing and testing reports.

Please refer to the product documentation for Oracle BI Publisher for more information on completing these tasks.

Converting an Oracle Report to BI Publisher

Converting an Oracle Report to a BI Publisher report is essentially a "migration" process. Oracle Utilities Work and Asset Management uses tools provided by BI Publisher which provide a starting place and set of objects that must be modified in order to derive a fully functioning BI Publisher report.

Please refer to the document titled: “Oracle Business Intelligence Publisher Converting Oracle Reports to Oracle Business Intelligence Publisher” available on the Oracle Technology Network (http://www.oracle.com/technetwork/middleware/bi-publisher/overview/index.html) or contact support for more information.

Chapter 9
Custom Menus

Custom menus allow administrators to provide users with easy access to reports or information that is not common to the base application.

Custom Menus View

The Custom Menus view is located in the Modules Administration - Forms module. Use this view to define up to 20 custom menu items for each module. The menu items that you create can open reports or custom API calls that you develop using SQL and SAPI Triggers. Once you complete the fields in this form, the system adds a menu item to the menu bar of the selected module. To users, the added menu items look like the standard options, but will add greater accessibility and functionality specific to your organization.

Users must have each individual module's custom menu option listed in their Responsibility profile in order to view or have access to any custom menus that appear in that module.

The Synergen Application Programming Interface (SAPI) is an empty library delivered with the application that is used to customize forms processing. It contains "hooks" into the core application that allow users to build form level triggers.

Custom Menus Field Descriptions

Form - Identifies the specific form where the new Menu item will appear in the Search Options window.

Menu Title - The text entered in this field will appear exactly as entered on the menu bar. Enter an '&' sign in front of the letter you want to designate as the hot key.

Sequence Number - If you enter multiple menu items number them according to the order that you want them to appear. Submenu options appear on the menu in the order of the sequence numbers, not in the order that they appear on the Custom Menu view.

Submenu Title - The text entered in this field will appear exactly as entered as a submenu of the Menu Title that you create. Enter an '&' sign in front of the letter you want to designate as the hot key.

Display? - Make sure that fields requiring validation have the "Invalid" indicator box checked rather than the "Check" indicator. The "Check" box should only be used for fields that will never be null. If a field is null and the "Check" indicator is set to "Y" the Check process will cause an error which will disrupt correct processing of the record duplication. For example, fields such as Asset ID, Department and Area on Work Requests or Work Orders should always be set to "Invalid".

Click this check box to make the Custom menu appear on the menu bar. This check box is a convenient way to add and remove custom menu items from the menu bar without having to physically enter and delete them from the Custom Menus view.

Type - You can create a custom menu item that performs a CUSTOM function or that links to a REPORT.

Action - The action field determines what the custom menu item will do when you select it from the menu bar. It should reference a specific report code or a custom API routine.

Use Primekeys in Where Clause? - If this check box is marked the "where" clause passed to the report is built using primekeys on the form. Primekeys limit or filter the results generated on a report. If this option is turned off (not checked) and the selection block is not specified in Reports Administration, reports will run automatically with no where clause. This may result in large reports that contain thousands of records which may crash your system.

How to Create a Custom Menu

1. **Open the Module Administration - Forms module in the Administration subsystem and highlight the form to which you want to add a custom menu.**
2. **Select Custom Menus from the Views list.**
   If a custom menu has already been created for that form, that record opens. If no custom menu has been created, a new record opens.

   If a Menu Title already exists you can edit it if necessary.
3. **Enter a Menu Title.**
4. **Click an empty line in the grid or click New.**
5. **Enter a sequence number for the menu option that you are adding.**
6. **Enter the Submenu Title.**
7. **Check the Display? box if you want the menu option to appear on the custom menu when you save the record.**
8. **Select a Type from the pull-down menu.**
   You can select either Report or Custom. Custom executes another action which you must specify. Refer to Custom Menu Actions for more information.

   If you set the type as "REPORT" the value for the Action should exactly match the report name. This must be a report which exists within the system. Set the type as "CUSTOM" to launch a custom action, file, or other executable when the menu item is selected. If you need to access custom reports set the type as "CUSTOM."
9. **Specify the Action to be performed.**
10. **Check the Use Primekeys in Where Clause? check box if you want the Where Clause passed to the report built using primekeys on the form.**
11. **Click Save.**
    The system saves the record and updates the custom menu.

Granting Access to Custom Menus

Users must have a module's custom menu option listed in their Responsibility profile in order to access any custom menus that appear in that module. Please refer to the chapter on Responsibilities for more information on how to add a function to a responsibility.

Custom Menu Actions

Configure the SAPI_TRIGGER "WHEN-CUSTOM-MENU-ACTION" to call the custom action that you define in the Action field of the Custom Menu screen. The event trigger coming in would look like the following example, where "<menu action name> is the value defined in the Action field.

```
        WHEN-CUSTOM-MENU-ACTION<menu action name>
```

Configure SAPI to launch a custom form using syntax similar to the following: (The example is for illustration purposes only. The path must be changed to a location appropriate for your environment.)

```
PROCEDURE SAPI_TRIGGER (

if trigger_name_in = 'WHEN-CUSTOM-MENU-ACTION-LAUNCH2343' then
web.show_document('C:\Files\Custom_Report_2343.xls');

)
```

Use similar syntax to launch an external application, web page, document, or other customized content that you want to access from the custom menu.

Chapter 10

Custom Actions in SIA

Modules built with the SIA architecture use custom actions to mimic the custom menu functionality available in Forms based modules. Currently, custom actions can be used to call a URL or an Oracle report.

Note: SAPI calls are not supported and no user interface is available to maintain custom actions. This document describes the basic configuration of custom actions.

The term 'custom actions' is used rather than 'custom menus' because in SIA modules the links appear on the left sidebar of the module under the standard Actions list. For example, in the Daily Schedule module, the Custom Actions menu would appear as a new section under Actions.

Database Table

Custom actions are defined in XT_CUSTOM_ACTION.

Column	Description
plant	plant code; prime key
custom_action_id	action id; prime key
custom_action_type	action type: REPORT or URL
custom_action_title	a short description that is displayed on toolbar
command_string	for action type REPORT, this value is the report name
	for action type URL, this value is the URL itself
pass_pks	pass prime keys from page (Y or N)
non_pk_names	column names delimited by commas
custom_action_desc	longer description of action
module	SIA module name (i.e. "DailySchedule")
page	the page where action should show up... normally "Data"
display	display action (Y or N)
display_order number	order to display action on custom action bar
created_by	audit fields here and below
created_date	
last_update_user	
last_update_date	

Defining Custom Actions

Until a user interface is available, the new table can be populated only from TOAD or the SQL prompt. Clients wishing to use custom actions should contact Oracle Support for assistance. There are no configuration items (business rules, responsibilities, etc.) associated with custom actions other than the XT_CUSTOM_ACTION table. If actions are defined in the table, they will appear in the specified module for all users having access to the module.

When defining custom actions, you can specify whether to pass the prime keys with the call and/or any non-prime keys you identify in the action.

The following example illustrates how to define a custom action in the Daily Schedule module to call report 41, passing prime keys from the Daily Schedule header to the report:

```
plant = 01
custom_action_id = DAILYSCH1
custom_action_type = REPORT
custom_action_title = My Report
command_string = S_RPT041
pass_pks = Y
module = DailySchedule
page = Data
display = Y
display_order = 1
```

This example defines a second custom action in the Daily Schedule module to call to a URL (www.google.com) without passing prime keys:

```
plant = 01
custom_action_id = DAILYSCH2
custom_action_type = URL
custom_action_title = Google
command_string = http://www.google.com
pass_pks = N
module = DailySchedule
page = Data
display = Y
display_order = 2
```

A final example defines a third custom action in the Daily Schedule module to call a report, passing both prime keys and a non prime key (schedule_type) from the Daily Schedule header:

```
plant = 01
custom_action_id = DAILYSCH3
custom_action_type = REPORT
custom_action_title = My Report with Type
command_string = S_RPT041
pass_pks = Y
non_pk_names = schedule_type
module = DailySchedule
page = Data
display = Y
display_order = 3
```

Converting Custom Menus to Custom Actions

When appropriate, existing custom menu information can be copied from Forms to produce custom actions in the specified SIA module. Below is an example of a migration script to copy data from the SA_FORM_CUSTOM_MENU table for the Weekly Schedule module to the XT_CUSTOM_ACTION table for similar support in the Daily Schedule module.

```
INSERT INTO XT_CUSTOM_ACTION
(
 PLANT,
 CUSTOM_ACTION_ID,
 CUSTOM_ACTION_TYPE,
 CUSTOM_ACTION_TITLE,
 COMMAND_STRING,
 PASS_PKS,
 NON_PK_NAMES,
 MODULE,
 PAGE,
 DISPLAY,
 DISPLAY_ORDER,
 CREATED_BY,
 CREATED_DATE,
 LAST_UPDATE_USER,
 LAST_UPDATE_DATE
)
SELECT DISTINCT
 cm.PLANT,
 'DAILYSHD' || ' ' || cm.SEQUENCE_NO "CUSTOM_ACTION_ID",
 cm.MENU_TYPE "CUSTOM_ACTION_TYPE",
 cm.HEADER_MENU_LABEL || ' ' || cm.MENU_LABEL "CUSTOM_ACTION_TITLE",
 cm.MENU_CALL "COMMAND_STRING",
 cm.MENU_PRIMEKEYS "PASS_PKS",
 '' "NON_PK_NAMES",
 'DailySchedule' "MODULE",
 'Data' "PAGE",
 cm.MENU_DISPLAY "DISPLAY",
 cm.SEQUENCE_NO "DISPLAY_ORDER",
 cm.CREATED_BY,
 cm.CREATED_DATE,
 cm.LAST_UPDATE_USER,
         cm.LAST_UPDATE_DATE
        FROM
         SA_FORM_CUSTOM_MENU cm,
         SA_FORM_NAMES fm
        WHERE
         cm.FORM_NAME = fm.FORM_NAME
         and fm.FORM_NAME = 'WKLYSCHD'
        AND 0 = (select count(*) from XT_CUSTOM_ACTION where PLANT = cm.PLANT
and CUSTOM_ACTION_ID = 'DAILYSHD' || ' ' || cm.SEQUENCE_NO)
        /
        commit
        /
```

Chapter 11

Custom Lists of Values in SIA

You must use a database development tool such as SQL Plus, SQL Developer, or Toad to access the database and create Special LOVs in SIA.

How to Create a Special List of Values in SIA

Create a custom list of values by identifying the field in the XT_NAMEDQUERY table then entering an override SQL statement in the XT_NAMEDQUERY_OVERRIDES table.

1. **Identify the QUERY_NAME for the field that you want to modify.**
   Open the module, right click the field containing the LOV, select Properties and note the Named Query value.

   In this example, select the Crew field in the Daily Schedule module and note that the Named Query value = HeaderCrew.

2. **Open your database development tool and locate the XT_NAMEDQUERY table.**

   ```
   SELECT
      CONTEXT, QUERY_NAME, QUERY_SQL
   FROM SYNERGEN.XT_NAMEDQUERY Tbl
   WHERE CONTEXT = 'DailySchedule' AND QUERY_NAME = 'HeaderCrew'
   ```
   Generally, the CONTEXT = the module name.

3. **Copy the contents of the QUERY_SQL field and paste it into a text editor such as Notepad.**
   Example: select c.crew, c.crew_desc, b.key_value from sa_crew c, sa_rule_key b where c.plant = ? and b.plant (+) = c.plant and b.rule_id (+) = 'DEFAULT BACKLOG GROUPS' and b.key_name (+) = c.crew and upper(crew) like upper(?)||'%' order by c.crew

4. **Edit the SQL as necessary to produce the desired results.**
   For this example, try removing all instances of the join statement '(+)' to yield a list which only shows Crews with a populated backlog group.

   Example: select c.crew, c.crew_desc, b.key_value from sa_crew c, sa_rule_key b where c.plant = ? and b.plant = c.plant and b.rule_id = 'DEFAULT BACKLOG GROUPS' and b.key_name = c.crew and upper(crew) like upper(?)||'%' order by c.crew

   You can modify the statement in any way, however the bind variables (indicated by a question mark '?') MUST remain in the same location as in the original statement.

   The '?' is a placeholder for a variable. In the example above, the variables are set to be plant and crew (in that order). In your sql, you must have exactly two '?' with the first instance of '?' having plant in it, and the second '?' having crew in it.

5. **Locate or create the corresponding ENTITY_NAME and NAMEDQUERY in the XT_NAMEDQUERY_OVERRIDES table.**

This is where you will enter the modified SQL to change the LOV.

```
SELECT
   NAMEDQUERY, ENTITY_NAME, QUERY_SQL
FROM SYNERGEN.XT_NAMEDQUERY_OVERRIDES Tbl
WHERE ENTITY_NAME = 'DailySchedule' AND NAMEDQUERY = 'HeaderCrew'
```

6. **Enter the modified SQL into the QUERY_SQL field and commit.**
7. **Return to the module to see the results.**
   The system displays applicable errors if the SQL does not work properly.

Chapter 12

Custom Depreciation

This chapter establishes guidelines on how to create a custom depreciation procedure whenever a change request is created to add or modify an asset. This chapter also includes a description on how to call this procedure from a SAPI trigger.

For sample transactions and calculations, please refer to "Custom Depreciation_Sample.xls" delivered in the documentation package with this release.

Phase 1: Creating a Custom Depreciation Procedure

The purpose of this section is to establish guidelines on how to create a custom depreciation procedure and on how to call it from an SAPI trigger.

- Create a custom procedure to create the depreciation forecast using the custom depreciation method.
- Update the Fixed Asset information once the custom depreciation forecast is created.

Create Depreciation Custom Procedure

The custom procedure must perform the following:

- Populate the depreciation forecast table
- Update the fixed asset information.

Populate the Depreciation Forecast Table

The depreciation forecast is stored in **SA_ASSET_DEP_FORECAST**. The fields that need to be populated are listed in Table 1: Fields to populate in SA_ASSET_DEP_FORECAST.

Update Fixed Asset Table Details

Update **SA_ASSET_FIXED_ASSET** once the depreciation forecast is created. The fields that need to be updated are listed in Table 2: Fields to update for SA_ASSET_FIXED_ASSET.

Modify the SAPI Trigger

Modify the SAPI trigger to call the newly created custom procedure.

Below is the code inside the SAPI trigger to call the new custom procedure.

```
PROCEDURE sapi_trigger (
  trigger_name_in           VARCHAR2,
3    plant_in                  VARCHAR2,
4    form_name_in              VARCHAR2,
5    block_name_in             VARCHAR2,
6    type_in                   VARCHAR2,
7    goto_field_out     IN OUT VARCHAR2,
8    trace_activity_out IN OUT VARCHAR2,
9    error_message_out  IN OUT VARCHAR2) IS
10
11   error_no NUMBER := 0;
12
13 BEGIN
14   error_message_out := NULL;
15   goto_field_out := NULL;
16
17   IF form_name_in IN ('ASSET', 'EASSET', 'FLTASSET') THEN
18     IF block_name_in = 'DEPRECIATION' THEN
19       IF trigger_name_in IN ('FORECAST', 'POST-INSERT', 'POST-
          UPDATE') THEN
20         <procedure_name>(parameter_1, parameter_2, … parameter_n ,
21                          error_no,
22                          trace_activity_out,
23                          error_message_out);
24
25         IF error_no = 0 THEN
26           sdbp_commit;
27         END IF;
28       END IF;
29     END IF;
30   END IF;
31
32 EXCEPTION
33   WHEN OTHERS THEN
34     error_no := SQLCODE
35     error_message_out := SUBSTR(SQLERRM,1,200);
36 END;
```

Description of Steps

This section provides descriptions of each line number in the procedure. The indicated step number corresponds to each line number.

1. The SAPI trigger called in the Asset, Enterprise Asset and Fleet Asset modules.
2. The name of the trigger.
3. Plant.
4. The name of the form, i.e. ASSET for Asset, EASSET for Enterprise Asset and FLTASSET for Fleet Asset.
5. Name of the block.
6. Type of transaction whether INSERT or UPDATE. Irrelevant in this context.
7. If the called procedure passes an error message, this parameter specifies what field to put the cursor in once processing goes back to the asset forms.
8. This parameter is used to trace what is being done on the called procedure. Helps to pinpoint exactly where in the called procedure the error occurred.
9. The actual error message from the called procedure. Return null if processing was successful.
10. Intentionally left blank.
11. Error number from the called procedure. Pass zero (0) if processing is successful.
12. Intentionally left blank.
13. Start of process.
14. Initialize error message.
15. Initialize go to field.
16. Intentionally left blank.

17. Evaluate form being processed. Calling form must be ASSET, EASSET and FLTASSET. Any other form should not call the custom procedure.
18. Evaluate the block where the transaction took place. Must be DEPRECIATION only.
19. Evaluate the trigger. Should be FORECAST, POST-UPDATE or POST-INSERT. The depreciation forecast should only be created upon save, whether after inserting or after updating the Fixed Asset record.
20. Call the custom procedure. Pass the parameters that are needed to create the forecast.
21. Error number from the called procedure. Pass zero (0) if successful.
22. Trace message. Irrelevant if error_no is zero.
23. The error message. Return null if successful.
24. Intentionally left blank.
25. Evaluate if the called procedure processed successfully.
26. Commit changes if processing is successful.
27. End check if error_no is 0.
28. End check of trigger name.
29. End check of block name.
30. End check of form name.
31. Intentionally left blank.
32. Start of exception block (error handling).
33. Check error type.
34. Set error_no to the SQL error number.
35. Substring SQL error message.
36. End of processing.

Phase 2: Creating a Custom Change Request Procedure

The purpose of this section is to establish guidelines on how to create a custom change request procedure and on how to call it from an SAPI trigger.

- Create a custom procedure to create a new asset, modify an existing asset or revaluate an asset via a Change Request.
- Update the Asset and Fixed Asset information, including the depreciation forecast if it exists, once the Change Request is set to completion.

Create Custom Procedure

There are three Change Request transactions allowed in this phase, namely New, Change, Configure and Revaluation. The values are found in **SA_CHANGE_REQUEST_ASSET_DATA**. The tables referenced in this section can be found in Appendix A: Reference Tables. Asset disposal/retirement is included in Phase 3: Creating a Custom Depreciation Procedure for Specific Asset Transactions.

1. Create a new Asset.
    1.1 Create a new asset in Active status.

    1.2 If the Depreciation section is populated, create a Fixed Asset record and the Depreciation Forecast.

2. Revaluate an Asset.
    2.1 Update the Replacement Value of the Fixed Asset.

3. Modify an Existing Asset.

3.1 Update the Asset with the requested change.

3.2 If the Depreciation section is populated, update Fixed Asset. Modify the forecast based on the changes.

4. Create Account Log records as needed.

Create a New Asset

1. Create a new record in **SA_ASSET**.
   The fields that need to be populated are listed in Table 3: Create a record in SA_ASSET.
   Use **SA_CHANGE_REQUEST_ASSET_DATA** as data source.

2. If the Depreciation section of the Change Request is populated, create a record in **SA_ASSET_FIXED_ASSET**.
   See Table 4: Create a record in SA_ASSET_FIXED_ASSET if the depreciation section of the Change Request is populated for the list of fields that need to be populated.

3. If the Asset ID was created using a sequence number, update the Asset ID field of **SA_CHANGE_REQUEST_ASSET_DATA**.
   Aside from the Asset ID, update the last_update_date and last_update_user fields as well using SYSDATE and USER as values respectively.

4. Create the depreciation forecast by calling the custom procedure created in Phase 1.
5. Update **SA_ASSET_FIXED_ASSET** once the depreciation forecast is created.
   See Table 5: Fields to update in SA_ASSET_FIXED_ASSET for the list of fields that need to be populated.

6. Create an Account Log entry by calling sdbp_batch_depreciation.insert_acct_log and pass the parameters listed in Table 6: Create an Account Log entry.
7. If the Manufacturer Warranty is populated, create record(s) in **SA_ASSET_MFR_WARRANTY**. The source data are in **SA_CHANGE_REQUEST_ASSET_WTY.**
   See Table 7: Create record(s) in SA_ASSET_MFR_WARRANTY for the list of fields that need to be populated.

Revaluate an Asset

1. Update **SA_ASSET_FIXED_ASSET**.
   The fields that need to be updated are listed in Table 8: Fields to update for SA_ASSET_FIXED_ASSET.

   Use **SA_CHANGE_REQUEST_ASSET_DATA** as data source. Use asset_id, asset_record_type and plant as qualifiers.

2. Update **SA_CHANGE_REQUEST_ASSET_DATA**.
   The fields that need to be updated are listed in Table 9: Field to update for SA_CHANGE_REQUEST_ASSET_DATA.

   Use asset_change_seq_no, change_request_no and plant as qualifiers.

Modify an Existing Asset

To modify an existing asset you must update SA_ASSET table. The fields that need to be updated are listed in Table 10: Fields to update for SA_ASSET. The tables referenced in this section can be found in Appendix A: Reference Tables.

- Use **SA_CHANGE_REQUEST_ASSET_DATA** as data source.

- Use asset_id, asset_record_type and plant as qualifiers.

- If any part of the Depreciation section is modified you must verify whether or not a record exists in SA_ASSET_FIXED_ASSET then manage several conditions based on whether or not a record exists:

  1. If no record is found:

     1.1 Create a record in **SA_ASSET_FIXED_ASSET** with fields in Table 11: Create a record in SA_ASSET_FIXED_ASSET.

     1.2 Create the depreciation forecast by calling the custom procedure created in Phase 1: Creating a Custom Depreciation Procedure.

     1.3 Update **SA_ASSET_FIXED_ASSET** once the depreciation forecast is created. The fields that need to be updated are listed in Table 12: Update SA_ASSET_FIXED_ASSET once the depreciation forecast is created.

  2. If a record is found, verify whether or not it already posted.

     2.1 If the Posted Indicator (acquisition_cost_posted_ind) is **unchecked:**

     2.1.1 Update the fields listed in Table 13: Fields to update if the Posted Indicator (acquisition_cost_posted_ind) is unchecked using plant, asset_record_type and asset_id as qualifiers.

     2.1.2 Delete the existing forecast, if it exists, and recreate it by calling the custom procedure created in Phase 1 based on the updated data.

     2.1.3 Create an Account Log entry by calling sdbp_batch_depreciation.insert_acct_log and pass the parameters listed in Table 14: Create an Account Log entry.

     2.2 If the Posted Indicator (acquisition_cost_posted_ind) is **checked:**

     2.2.1 If there is a change in Acquisition Cost:

     2.2.1.1 Create an Account Log entry by calling sdbp_batch_depreciation.insert_acct_log and pass the parameters found in Table 15: Create an Account Log entry if the Posted Indicator (acquisition_cost_posted_ind) is checked.

     2.2.1.2 If the Record Loss Indicator is checked, create an Account Log entry by calling sdbp_batch_depreciation.insert_acct_log and pass the parameters found in Table 16: Create an Account Log entry if the Record Loss Indicator is checked.

     2.2.2 If the Asset Cost Account is changed, create Account Log entries by calling sdbp_batch_depreciation.insert_acct_log and pass the parameters found in Table 17: Create Account Log entries if the Asset Cost Account is changed and Table 18: Create Account Log entries if the Asset Cost Account is changed.

     2.2.3 If the Accum Depreciation Account is changed, create Account Log entries by calling sdbp_batch_depreciation.insert_acct_log and pass the parameters found in Table 19: Create Account Log entries if the Accum Depreciation Account is changed and Table 20: Create Account Log entries if the Accum Depreciation Account is changed.

     2.2.4 Update **SA_ASSET_FIXED_ASSET** plant, asset_id and asset_record_type as qualifiers. See Table 21: Update SA_ASSET_FIXED_ASSET plant, asset_id and asset_record_type as qualifiers. for the fields to be updated.

     2.2.5 If there is a change in Acquisition Cost, Useful Life and/or Salvage Value, create a record in **SA_CHANGE_REQUEST_ASSET_LOG**. See

Table 22: Create a record in SA_CHANGE_REQUEST_ASSET_LOG if there is a change in Acquisition Cost, Useful Life and/or Salvage Value. for the fields to populate.

2.2.6 If there is a change in Acquisition Cost, Useful Life and/or Salvage Value:

2.2.6.1 Recreate the depreciation forecast based on the modifications. Refer to Custom Depreciation_Sample.xls" delivered in the documentation package with this release.

2.2.6.2 After creating the forecast, update the Remaining Life Period, Remaining Life Year, Useful Life and Salvage Value. The Acquisition Cost is not updated once the asset is posted.

Modify the SAPI Trigger

Modify the SAPI trigger to call the newly-created custom procedure.

The following code is included in the SAPI trigger to call the new custom procedure.

```
PROCEDURE sapi_trigger (
  trigger_name_in             VARCHAR2,
  plant_in                    VARCHAR2,
  form_name_in                VARCHAR2,
  block_name_in               VARCHAR2,
  type_in                     VARCHAR2,
  goto_field_out       IN OUT VARCHAR2,
  trace_activity_out   IN OUT VARCHAR2,
  error_message_out    IN OUT VARCHAR2) IS

  error_no NUMBER := 0;

BEGIN
  error_message_out := NULL;
  goto_field_out := NULL;

  IF form_name_in = 'CHNGREQ' THEN
    IF block_name_in = 'HEADER' THEN
      IF type_in = 'STATUS CHANGE' THEN
        IF trigger_name_in = 'CHANGE REQUEST ASSET' THEN
         <procedure_name>(parameter_1, parameter_2, … parameter_n ,
                          error_no,
                          trace_activity_out,
                          error_message_out);

          IF error_no = 0 THEN
            sdbp_commit;
          END IF;
        END IF;
      END IF;
    END IF;
  END IF;
EXCEPTION
  WHEN OTHERS THEN
    error_no := SQLCODE
    error_message_out := SUBSTR(SQLERRM,1,200);
END;
```

Description of Steps

This section provides descriptions of each line number in the procedure. The indicated step number corresponds to each line number.

1. The SAPI trigger called in the Change Request module.
2. The name of the trigger.
3. Plant.
4. The name of the form, i.e. CHNGREQ for Change Request.
5. Name of the block.
6. Type of transaction whether INSERT or UPDATE. Irrelevant in this context.
7. If the called procedure passes an error message, this parameter specifies what field to put the cursor in once processing goes back to the Change Request form.
8. This parameter is used to trace what is being done on the called procedure. Helps to pinpoint exactly where in the called procedure the error occurred.
9. The actual error message from the called procedure. Return null if processing was successful.
10. Intentionally left blank.
11. Error number from the called procedure. Pass zero (0) if processing is successful.
12. Intentionally left blank.
13. Start of process.
14. Initialize error message.
15. Initialize go to field.
16. Intentionally left blank.
17. Evaluate form being processed. Calling form must be CHNGREQ.
18. Evaluate the block where the transaction took place. Must be DEPRECIATION only.
19. Evaluate the type. Must be 'STATUS CHANGE'.
20. Evaluate the trigger. Must be CHANGE_REQUEST ASSET'.
21. Call the custom procedure. Pass the parameters that are needed to make the changes to the asset.
22. Error number from the called procedure. Pass zero (0) if successful.
23. Trace message. Irrelevant if error_no is zero.
24. The error message. Return null if successful.
25. Intentionally left blank.
26. Evaluate if the called procedure processed successfully.
27. Commit changes if processing is successful.
28. End check if error_no is 0.
29. End check of trigger name.
30. End check of type.
31. End check of block name.
32. End check of form name.
33. Start of exception block (error handling).
34. Check error type.
35. Set error_no to the SQL error number.
36. Substring SQL error message.
37. End of processing.

Phase 3: Creating a Custom Depreciation Procedure for Specific Asset Transactions

The purpose of the following section is to establish guidelines on how to create a custom depreciation procedure to support Asset Reversal, Post to Prior Year and Disposal transactions. The tables referenced in this section can be found in Appendix A: Reference Tables. For a more comprehensive guide on how to perform Reversal, Post to Prior Year and Disposal transactions, refer to Appendix B: Performing Custom Depreciation Phase 3.

- Create a custom procedure to support Reversal, Post to Prior Year and Disposal of an asset via a Change Request.
- Update Asset and Fixed Asset information, including the depreciation forecast, once the Change Request is set to completion.

Create Custom Depreciation Procedure

Reversal and Post to Prior Year are asset change transactions where the concatenated accounting year and period of the Effective Date is less than or equal to the Last Period Processed in the Asset Depreciation Business Rule. The similarity between Reversal and Post to Prior Year is that the forecast records that have already been posted from the Effective Date up to the Last Period Processed are backtracked and corresponding account log entries are written to record the change. The difference between the two is that after the records have been reversed and the forecast recreated to reflect the new changes, Post to Prior Year posts the new forecast record from the Effective Date up to the end of the previous year. Account log entries are also made to reflect the posting of the new forecast records. Both transactions are extensions of Asset Change.

Asset Disposal is the same as asset retirement.

Modify Fixed Asset

1. If there is a change in Acquisition Cost, call sdbp_batch_depreciation.insert_acct_log to create an Account Log entry and pass the parameters found in Table 23: Create an Account Log entry..
2. If the Record Loss Indicator is checked, create an Account Log entry by calling sdbp_batch_depreciation.insert_acct_log and pass the parameters found in Table 24: Create an Account Log entry if the Record Loss Indicator is checked..
3. If the Asset Cost Account is changed, create Account Log entries by calling sdbp_batch_depreciation.insert_acct_log and pass the parameters found in Table 25: Create Account Log entries if the Asset Cost Account is changed (1 of 2). and Table 26: Create Account Log entries if the Asset Cost account is changed (2 of 2)..
4. If the Accum Depreciation Account is changed, call sdbp_batch_depreciation.insert_acct_log to create Account Log entries and pass the parameters found in Table 27: Create Account Log entries if the Accum Depreciation account is changed (1 of 2). and Table 28: Create Account Log entries if the Accum Depreciation account is changed (2 of 2)..
5. If there is a change in Acquisition Cost, Salvage Value or Useful Life:

   5.1 If the accounting year and period of the Effective Date is less than or equal to the Last Period Processed key value in the Asset Depreciation Business Rule, then the Effective Date is in the past.

   5.2 If the Effective Date is in the past, perform Reversals:

   5.2.1 Retrieve all posted forecast records of the asset where the year and period are between the year and period of the Effective Date and the value in the Last Period Processed key in the Asset Depreciation Business Rule.

   5.2.2 Reverse the posted depreciation forecast's transaction amounts by calling sdbp_batch_depreciation.insert_acct_log to create Account Log entries and pass the parameters found in Table 29: Create Account Log entries to reverse the posted depreciation forecast's transaction amounts (1 of 2). and Table 30: Create Account Log entries to reverse the posted depreciation forecast's transaction amounts (2 of 2)..

5.2.3 Update the Asset Forecast (**SA_ASSET_DEP_FORECAST**) and unpost the record (ie set the Post to Account Log Indicator field to N). Update the Last Update Date and User fields as well.

5.2.4 Update Fixed Asset. Refer to Table 31: Fields to update in Fixed Asset. for the list of fields that need to be modified. Use Asset Record Type, ID and Plant as qualifiers. Take note that as forecast records are being unposted, the Remaining Life Year and Period of the asset increases, hence, the fields are included in the update.

5.3 Update **SA_ASSET_FIXED_ASSET** plant, asset_id and asset_record_type as qualifiers. See Table 32: Update SA_ASSET_FIXED_ASSET. for the fields to be updated.

5.4 Create a record in **SA_CHANGE_REQUEST_ASSET_LOG**. See Table 33: Create a record in SA_CHANGE_REQUEST_ASSET_LOG. for the fields to populate.

5.5 Recreate the depreciation forecast based on the modifications. Refer to Custom Depreciation Phase 2 Excel Worksheet for a list of sample changes.

5.6 Update the Useful Life, Remaining Life Year, Remaining Life Period and Salvage Value. The Acquisition Cost is not updated once the asset is posted.

5.7 If the Effective Date is in the past and the Post to Prior Year indicator is Y, perform Post to Prior Year:

5.7.1 Retrieve the reversed forecast records in Section 5.2 from the Effective Date up to the end of the prior year.

5.7.2 Post the forecast record by calling sdbp_batch_depreciation.insert_acct_log to create Account Log entries and pass the parameters found in Table 34: Create Account Log entries to post the forecast record (1 of 2). and Table 35: Create Account Log entries to post the forecast record (2 of 2)..

5.7.3 Update the Asset Forecast. Set the Posted to Account Log Indicator to Y and the Posted Date to the current date. Update the Last Update Date and User fields as well.

5.7.4 Update the Fixed Asset table. Refer to Table 36: Fields to update in Fixed Asset. for the list of fields that need to be modified. Use Asset Record Type, ID and Plant as qualifiers. Take note that as forecast records are being posted, the Remaining Life Year and Period of the asset decreases.

Disposal

1. Derive the Period Year, Period Month and Transaction Date of the Retirement Date:

1.1 If the Post to Prior Year Indicator is **checked**:

1.1.1 Period Year is the Prior Year.

1.1.2 Period Month is the Last Period of 1.1.1.

1.1.3 Transaction Date is the End Date of 1.1.1 and 1.1.2.

1.2 Otherwise:

1.2.1 Period Year is the year of the Effective Date from the Accounting Period table

1.2.2 Period Month is the month of 1.2.1.

1.2.3 Transaction Date is the Change Request Status Date.

2. Derive Accumulated Depreciation and Written Down Value:

   2.1 Retrieve the Year, Period, Dep Expense, Accumulated Dep and the Net Book Value of the last Asset Forecast record.

   2.2 If the asset is fully depreciated (ie last forecast record is posted) or the year and period of the last forecast record is less than the year and period of the Retirement Date

      2.2.1 The Written Down Value is equal to the Fixed Asset's Salvage Value.

      2.2.2 The Accumulated Depreciation is equal to the Accumulated Dep.

   2.3 Otherwise:

      2.3.1 Retrieve the Dep Expense, Accumulated Dep and the Net Book Value from the Asset Forecast for the year and period of the Retirement Date.

      2.3.2 Compute Partial Percentage. Refer to Note 2.

      2.3.3 The Written Down Value is the sum of Net Book Value and Dep Expense less the product of Partial Percentage and Dep Expense.

      2.3.4 The Accumulated Depreciation is the difference of Accumulated Dep and Dep Expense plus the product of Partial Percentage and Dep Expense.

3. Insert Asset Cost Write-Off Account Log entry:

   3.1 Compute Asset Cost Write-Off:

      3.1.1 Get the sum of all the Asset Cost Adjustments from the Asset Forecast

      3.1.2 Add the Acquisition Cost and Change Request Cost Adjustment to 3.1.1.

      3.1.3 Get the absolute value in 3.1.2 and multiply by –1.

   3.2 Call sdbp_batch_depreciation.insert_acct_log to create an Account Log entry and pass the parameters found in Table 37: Create an Account Log entry to insert Asset Cost Write-Off..

4. Insert Asset Gain/Loss Account Log entry:

   4.1 Compute Asset Gain/Loss Amount:

      4.1.1 Get the sum of the derived Written Down Value from Item 2 above and Cost Adjustment from the Change Request.

      4.1.2 Deduct the value from Item 4.1.1 above from the Change Request's Proceeds Received.

      4.1.3 The Asset Gain/Loss amount is the value in Item 4.1.2 above multiplied by –1.

   4.2 Call sdbp_batch_depreciation.insert_acct_log to create an Account Log entry and pass the parameters found in Table 38: Create an Account Log entry to compute Asset Gain/Loss Amount..

5. Insert Asset Accumulated Depreciation Account Log entry:

   5.1 The Asset Accumulated Depreciation Amount is the derived Accumulated Depreciation value from Item 2 above.

   5.2 Call sdbp_batch_depreciation.insert_acct_log to create an Account Log entry and pass the parameters found in Table 39: Create an Account Log entry to insert Asset Accumulated Depreciation Account..

6. If there is an Asset Cost Adjustment in the Change Request, create account log entries by calling sdbp_batch_depreciation.insert_acct_log and pass the parameters found in

Table 40: Create Account Log entries if there is an Asset Cost Adjustment in the change Request (1 of 2). and Table 41: Create Account Log entries if there is an Asset Cost Adjustment in the change Request (2 of 2)..

7. If the accounting year and period of the Retirement Date is less than or equal to the Last Period Processed key value in the Asset Depreciation Business Rule, then the Retirement Date is in the past.

   7.1 If the Retirement Date is in the past, perform Reversals:

      7.1.1 Retrieve all posted forecast records of the asset where the year and period are between the year and period of the Retirement Date and the value in the Last Period Processed key in the Asset Depreciation Business Rule.

      7.1.2 Reverse the posted depreciation forecast's transaction amounts by calling sdbp_batch_depreciation.insert_acct_log to create Account Log entries and pass the parameters found in Table 29: Create Account Log entries to reverse the posted depreciation forecast's transaction amounts (1 of 2). and Table 30: Create Account Log entries to reverse the posted depreciation forecast's transaction amounts (2 of 2)..

      7.1.3 Update the Asset Forecast and unpost the record (ie set the Post to Account Log Indicator field to N). Update the Last Update Date and User fields as well.

      7.1.4 Update Fixed Asset. Refer to Table 31: Fields to update in Fixed Asset. for the list of fields that need to be modified. Use Asset Record Type, ID and Plant as qualifiers. Take note that as forecast records are being unposted, the Remaining Life Year and Period of the asset increases, hence, the fields are included in the update.

      7.1.5 Update the Asset Forecast record where the year and period of the forecast is equal to the year and period of the Retirement Date. Refer to Table 42: Fields to update in Asset Forecast..

         7.1.5.1 Written Down Value is the sum of Net Book Value, Dep Expense and Cost Adjustment from the Change Request less the product of the Dep Expense and Partial Percentage (refer to Note 2).

         7.1.5.2 Accumulated Depreciation is the Accumulated Dep less the Dep Expense plus the product of DepExpense and Partial Percentage.

         7.1.5.3 Depreciation Expense is the product Dep Expense and Partial Percentage.

         7.1.5.4 Value Adjustment is the Cost Adjustment from the Change Request.

8. Delete all Asset Forecast where the year and period are greater that the year and period of the Retirement Date.

9. If the Effective Date is in the past and the Post to Prior Year indicator is Y, perform Post to Prior Year:

   9.1 Update Fixed Asset. Set the Retirement Date to the Effective Date and the Retirement Indicator to Y. Update the Last Update Date and User as well.

   9.2 There will only be one record remaining that is unposted after Items 7 and 8 above. Post the forecast record by calling sdbp_batch_depreciation.insert_acct_log to create Account Log entries and pass the parameters found in Table 34: Create Account Log entries to post the forecast record (1 of 2). and Table 35: Create Account Log entries to post the forecast record (2 of 2)..

9.3 Update the Asset Forecast. Set the Posted to Account Log Indicator to Y and the Posted Date to the current date. Update the Last Update Date and User fields as well.

9.4 Update the Fixed Asset table. Refer to Table 43: Fields to update in Fixed Asset. for the list of fields that need to be modified. Use Asset Record Type, ID and Plant as qualifiers.

10. Update the Asset's status to Retired. Also, update the Last Update Date and User.

Notes

1. Reversal and Post to Prior Year are extensions of Asset Change. The custom procedure created in Phase 2: Creating a Custom Change Request Procedure for Asset Change should be extended to accommodate transactions mentioned above.
2. To compute for the Partial Percentage:

   2.1 Retrieve the Accounting Year, Period, Start Date and End Date of the Retirement Date.

   2.2 Compute dividend by deducting Start Date from the Retirement Date and add 1.

   2.3 Compute divisor by deducting Start Date from End Date and add 1.

   2.4 Partial Percentage is the quotient of 2.2 and 2.3.
3. When disposing or retiring an asset, Effective Date is the same as Retirement Date.

Appendix A: Reference Tables

Use this section to find the fields and parameters needed to populate the various tables referenced in the steps to create, update and modify asset information.

Table 1: Fields to populate in SA_ASSET_DEP_FORECAST

Column Name	Data Type	Comments
PLANT	VARCHAR2 (3)	
ASSET_RECORD_TYPE	VARCHAR2 (1)	
ASSET_ID	VARCHAR2 (15)	
YEAR	VARCHAR2 (4)	
PERIOD	VARCHAR2 (3)	
DEPRECIATION_METHOD	VARCHAR2 (20)	Custom depreciation method
ACQUIRED_COST	NUMBER (15,2)	Acquisition cost of the asset
USEFUL_LIFE	NUMBER (8)	
DEPRECIATION_RATE	NUMBER	
SALVAGE_VALUE	NUMBER (15,2)	
DEPRECIATION_EXPENSE	NUMBER (15,2)	Depreciation amount for the year and period
ACCUM_DEPRECIATION	NUMBER (15,2)	First record is 0
WRITTEN_DOWN_VALUE	NUMBER (15,2)	First record equals Acquisition Cost
ASSET_CLASS	VARCHAR2 (20)	
PROPERTY_UNIT_NO	VARCHAR2 (15)	
POSTED_TO_ACCT_LOG_IND	VARCHAR2 (1)	Default to 'N'
CREATED_DATE	DATE	Default to SYSDATE
CREATED_BY	VARCHAR2 (30)	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2 (30)	Default to USER

Table 2: Fields to update for SA_ASSET_FIXED_ASSET

Column Name	Data Type	Comments
DEPRECIATION_RATE	NUMBER	
RATE_DENOMINATOR	NUMBER (8)	Equate to Useful Life

Table 2: Fields to update for SA_ASSET_FIXED_ASSET

Column Name	Data Type	Comments
CURRENT_BOOK_VALUE	NUMBER (8)	Equate to Acquisition Cost
ACCUM_DEPRECIATION	NUMBER (15,2)	Default to 0
REMAINING_LIFE_YEAR	NUMBER (8)	Equate to Useful Life
REMAINING_LIFE_PERIOD	NUMBER (3)	Default to 0
TOTAL_VALUE_ADJUSTMENT	NUMBER (15,2)	Default to 0
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2 (30)	Default to USER

Table 3: Create a record in SA_ASSET

Column Name	Data Type	Comments
PLANT	VARCHAR2	
ASSET_RECORD_TYPE	VARCHAR2	
ASSET_ID	VARCHAR2	Verify whether SA_ASSET has a sequence number in the Sequence module. If it exists, use the next sequence number as Asset ID. Otherwise, user must supply the Asset ID. If the user did not supply an Asset ID, display error message SYN-02083.
ASSET_DESC	VARCHAR2	
ACCOUNT_NO	VARCHAR2	
AREA	VARCHAR2	
DEPARTMENT	VARCHAR2	
BUILDING	VARCHAR2	
LOCATION	VARCHAR2	
POSITION	VARCHAR2	
ASSET_TYPE	VARCHAR2	
ASSET_CLASS	VARCHAR2	
CRITICALITY	VARCHAR2	
PARENT_ASSET_RECORD_TYPE	VARCHAR2	

Table 3: Create a record in SA_ASSET

Column Name	Data Type	Comments
PARENT_ASSET_ID	VARCHAR2	
PROCESS_NO	VARCHAR2	
BOM_ID	VARCHAR2	
SPECIFICATION_NO	VARCHAR2	
SPECIFICATION_CATEGORY	VARCHAR2	
SPECIFICATION_TYPE	VARCHAR2	
LOCATION_BASIS	VARCHAR2	
ROOM	VARCHAR2	
BREAKER_NO	VARCHAR2	
POINT_ID	VARCHAR2	
NUMBER_PREFIX	VARCHAR2	
STREET_NUMBER	NUMBER	
STREET_NUMBER_CHAR	VARCHAR2	
NUMBER_SUFFIX	VARCHAR2	
STREET_NAME	VARCHAR2	
STREET_DIRECTION	VARCHAR2	
CROSS_STREET	VARCHAR2	
CITY	VARCHAR2	
STATE_PROVINCE	VARCHAR2	
POSTAL_CODE	VARCHAR2	
OFFSET	VARCHAR2	
DIRECTION	VARCHAR2	
FROM_ASSET_RECORD_TYPE	VARCHAR2	
TO_ASSET_RECORD_TYPE	VARCHAR2	
FROM_ASSET_ID	VARCHAR2	
TO_ASSET_ID	VARCHAR2	
PLANNER	VARCHAR2	
ROUTING_LIST_ID	VARCHAR2	
BACKLOG_GROUP	VARCHAR2	
SAFETY_CRITICAL_IND	CHAR	Default to N
ISO_IND	CHAR	Default to N

Table 3: Create a record in SA_ASSET

Column Name	Data Type	Comments
HEALTH_IND	CHAR	Default to N
ENVIRONMENTAL_IND	CHAR	Default to N
RUN_TO_FAILURE_IND	CHAR	Default to N
SUITE	VARCHAR2	
ASSET_STATUS	VARCHAR2	ACTIVE
CLASS	VARCHAR2	
BREAKER_ASSET_RECORD_TYPE	VARCHAR2	
BREAKER_ASSET_ID	VARCHAR2	
GIS_GPS_LONGITUDE	VARCHAR2	
GIS_GPS_LATITUDE	VARCHAR2	
CREATION_DATE	DATE	Default to SYSDATE
CREATED_BY	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 4: Create a record in SA_ASSET_FIXED_ASSET if the depreciation section of the Change Request is populated

Column Name	Data Type	Comments
PLANT	VARCHAR2	
ASSET_RECORD_TYPE	VARCHAR2	
ASSET_ID	VARCHAR2	
PROPERTY_UNIT_NO	VARCHAR2	
DEPRECIATION_METHOD	VARCHAR2	
ACQUIRED_COST	NUMBER	
SALVAGE_VALUE	NUMBER	
REPLACEMENT_VALUE	NUMBER	
USEFUL_LIFE	NUMBER	
USEFUL_LIFE_UNITS	VARCHAR2	
ACQUISITION_DATE	DATE	
CONTRIBUTED_IND	CHAR	Set to N if unchecked
IN_SERVICE_DATE	DATE	

Table 4: Create a record in SA_ASSET_FIXED_ASSET if the depreciation section of the Change Request is populated

Column Name	Data Type	Comments
CURRENT_BOOK_VALUE	NUMBER	Equal to Acquired Cost
REMAINING_LIFE_YEAR	NUMBER	Equal to Useful Life
PO_NO	VARCHAR2	
DEP_ACCOUNT_NO	VARCHAR2	
DEP_EXPENSE_CODE	VARCHAR2	
ACCUM_DEP_ACCOUNT	VARCHAR2	
ACCUM_DEP_EXPENSE_CODE	VARCHAR2	
ASSET_COST_ACCOUNT	VARCHAR2	
ASSET_COST_EXPENSE_CODE	VARCHAR2	
ASSET_GAIN_LOSS_ACCOUNT	VARCHAR2	
ASSET_GAIN_LOSS_EXPENSE	VARCHAR2	
FUTURE_RETIREMENT_DATE	DATE	
ACQUISITION_COST_POSTED_IND	CHAR	Set to Y
CREATED_DATE	DATE	Default to SYSDATE
CREATED_BY	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 5: Fields to update in SA_ASSET_FIXED_ASSET

Column Name	Data Type	Comments
DEPRECIATION_RATE	NUMBER	
RATE_DENOMINATOR	NUMBER	Equal to Useful Life
CURRENT_BOOK_VALUE	NUMBER	Equate to Acquisition Cost
ACCUM_DEPRECIATION	NUMBER	Default to 0
REMAINING LIFE YEAR	NUMBER	Equal to Useful Life
REMAINING_LIFE_PERIOD	NUMBER	Default to 0
TOTAL_VALUE_ADJUSTMENT	NUMBER	Set to 0
LAST_UPDATE_USER	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE

Table 6: Create an Account Log entry

Call sdbp_batch_depreciation.insert_acct_log and pass the following parameters:

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Asset Cost Account
EXPENSE_CODE_IN	VARCHAR2	Asset Cost Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	NULL
ACCUMULATED_DEPRECIATION_IN	NUMBER	NULL
TRANSACTION_TYPE_IN	VARCHAR2	‘AN’
TRANSACTION_AMOUNT_IN	NUMBER	Acquisition Cost
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	In Service Date. If it is null, use Acquisition Date. If Acquisition Date is null, use Effective Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 7: Create record(s) in SA_ASSET_MFR_WARRANTY

Column Name	Data Type	Comments
PLANT	VARCHAR2	
ASSET_RECORD_TYPE	VARCHAR2	
ASSET_ID	VARCHAR2	
WARRANTY_ID	VARCHAR2	
ASSET_WTY_EXPIRATION_STATUS	VARCHAR2	
START_DATE	NUMBER	
EXPIRATION_ALERT_SENT_DATE	NUMBER	Set to NULL

Table 8: Fields to update for SA_ASSET_FIXED_ASSET

Column Name	Data Type	Comments
REPLACEMENT_VALUE	NUMBER	
LAST_UPDATE_DATE	DATE	SYSDATE
LAST_UPDATE_USER	VARCHAR2	USER

Table 9: Field to update for SA_CHANGE_REQUEST_ASSET_DATA

Column Name	Data Type	Comments
CHANGES_APPLY_DATE	NUMBER	SYSDATE
LAST_UPDATE_DATE	DATE	SYSDATE
LAST_UPDATE_USER	VARCHAR2	USER

Table 10: Fields to update for SA_ASSET

Column Name	Data Type	Comments
ASSET_DESC	VARCHAR2	
ACCOUNT_NO	VARCHAR2	
AREA	VARCHAR2	
DEPARTMENT	VARCHAR2	
BUILDING	VARCHAR2	
LOCATION	VARCHAR2	
POSITION	VARCHAR2	
ASSET_TYPE	VARCHAR2	
ASSET_CLASS	VARCHAR2	
CLASS	VARCHAR2	Equal to Asset Class

Table 10: Fields to update for SA_ASSET

Column Name	Data Type	Comments
CRITICALITY	VARCHAR2	
PARENT_ASSET_RECORD_TYPE	VARCHAR2	
PARENT_ASSET_ID	VARCHAR2	
PROCESS_NO	VARCHAR2	
BOM_ID	VARCHAR2	
SPECIFICATION_NO	VARCHAR2	
SPECIFICATION_CATEGORY	VARCHAR2	
SPECIFICATION_TYPE	VARCHAR2	
LOCATION_BASIS	VARCHAR2	
ROOM	VARCHAR2	
BREAKER_NO	VARCHAR2	
BREAKER_ASSET_ID	VARCHAR2	
GIS_GPS_LONGITUDE	VARCHAR2	
GIS_GPS_LATITUDE	VARCHAR2	
POINT_ID	VARCHAR2	
NUMBER_PREFIX	VARCHAR2	
STREET_NUMBER	NUMBER	
STREET_NUMBER_CHAR	VARCHAR2	
NUMBER_SUFFIX	VARCHAR2	
SUITE	VARCHAR2	
STREET_NAME	VARCHAR2	
STREET_DIRECTION	VARCHAR2	
CROSS_STREET	VARCHAR2	
CITY	VARCHAR2	
STATE_PROVINCE	VARCHAR2	
POSTAL_CODE	VARCHAR2	
OFFSET	VARCHAR2	
DIRECTION	VARCHAR2	
FROM_ASSET_RECORD_TYPE	VARCHAR2	
TO_ASSET_RECORD_TYPE	VARCHAR2	
FROM_ASSET_ID	VARCHAR2	

Table 10: Fields to update for SA_ASSET

Column Name	Data Type	Comments
TO_ASSET_ID	VARCHAR2	
ROUTING_LIST_ID	VARCHAR2	
PLANNER	VARCHAR2	
BACKLOG_GROUP	VARCHAR2	
SAFETY_CRITICAL_IND	CHAR	
ISO_IND	CHAR	
ENVIRONMENTAL_IND	CHAR	
RUN_TO_FAILURE_IND	CHAR	
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 11: Create a record in SA_ASSET_FIXED_ASSET

Column Name	Data Type	Comments
PLANT	VARCHAR2	
ASSET_RECORD_TYPE	VARCHAR2	
ASSET_ID	VARCHAR2	
PROPERTY_UNIT_NO	VARCHAR2	
DEPRECIATION_METHOD	VARCHAR2	
ACQUIRED_COST	NUMBER	
SALVAGE_VALUE	NUMBER	
REPLACEMENT_VALUE	NUMBER	
USEFUL_LIFE	NUMBER	
USEFUL_LIFE_UNITS	VARCHAR2	
ACQUISITION_DATE	DATE	
CONTRIBUTED_IND	CHAR	Set to N if unchecked
IN_SERVICE_DATE	DATE	
CURRENT_BOOK_VALUE	NUMBER	Equal to Acquired Cost
REMAINING_LIFE_YEAR	NUMBER	Equal to Useful Life
DEP_ACCOUNT_NO	VARCHAR2	
DEP_EXPENSE_CODE	VARCHAR2	

Table 11: Create a record in SA_ASSET_FIXED_ASSET

Column Name	Data Type	Comments
ACCUM_DEP_ACCOUNT	VARCHAR2	
ACCUM_DEP_EXPENSE_CODE	VARCHAR2	
ASSET_COST_ACCOUNT	VARCHAR2	
ASSET_COST_EXPENSE_CODE	VARCHAR2	
ASSET_GAIN_LOSS_ACCOUNT	VARCHAR2	
ASSET_GAIN_LOSS_EXPENSE	VARCHAR2	
FUTURE_RETIREMENT_DATE	DATE	
ACQUISITION_COST_POSTED_ IND	CHAR	Set to Y
CREATED_DATE	DATE	Default to SYSDATE
CREATED_BY	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 12: Update SA_ASSET_FIXED_ASSET once the depreciation forecast is created
The fields that need to be updated are listed here:

Column Name	Data Type	Comments
DEPRECIATION_RATE	NUMBER	
RATE_DENOMINATOR	NUMBER	Equal to Useful Life
CURRENT_BOOK_VALUE	NUMBER	Equate to Acquisition Cost
ACCUM_DEPRECIATION	NUMBER	Default to 0
REMAINING LIFE YEAR	NUMBER	Equal to Useful Life
REMAINING_LIFE_PERIOD	NUMBER	Default to 0
TOTAL_VALUE_ADJUSTMENT	NUMBER	Set to 0
LAST_UPDATE_USER	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE

Table 13: Fields to update if the Posted Indicator (acquisition_cost_posted_ind) is **unchecked**

Column Name	Data Type	Comments
PROPERTY_UNIT_NO	VARCHAR2	
DEPRECIATION_METHOD	VARCHAR2	
ACQUIRED_COST	NUMBER	
SALVAGE_VALUE	NUMBER	
REPLACEMENT_VALUE	NUMBER	
USEFUL_LIFE	NUMBER	
USEFUL_LIFE_UNITS	VARCHAR2	
IN_SERVICE_DATE	DATE	Set to Effective Date (Transaction Date)
CURRENT_BOOK_VALUE	NUMBER	Equal to Acquired Cost
REMAINING_LIFE_YEAR	NUMBER	Equal to Useful Life
DEP_ACCOUNT_NO	VARCHAR2	
DEP_EXPENSE_CODE	VARCHAR2	
ACCUM_DEP_ACCOUNT	VARCHAR2	
ACCUM_DEP_EXPENSE_CODE	VARCHAR2	
ASSET_COST_ACCOUNT	VARCHAR2	
ASSET_COST_EXPENSE_CODE	VARCHAR2	
ASSET_GAIN_LOSS_ACCOUNT	VARCHAR2	
ASSET_GAIN_LOSS_EXPENSE	VARCHAR2	
ACQUISITION_COST_POSTED_IND	CHAR	Set to Y
RATE_DENOMINATOR	NUMBER	Equal to Useful Life
DEPRECIATION_RATE	NUMBER	
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 14: Create an Account Log entry

Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here:

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Asset Cost Account
EXPENSE_CODE_IN	VARCHAR2	Asset Cost Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	NULL
ACCUMULATED_DEPRECIATION_IN	NUMBER	NULL
TRANSACTION_TYPE_IN	VARCHAR2	'AN'
TRANSACTION_AMOUNT_IN	NUMBER	Acquisition Cost
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	In Service Date. If it is null, use Acquisition Date. If Acquisition Date is null, use Effective Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 15: Create an Account Log entry if the Posted Indicator (acquisition_cost_posted_ind) is checked
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here:

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Asset Cost Account
EXPENSE_CODE_IN	VARCHAR2	Asset Cost Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'AC'
TRANSACTION_AMOUNT_IN	NUMBER	Difference between Old Acquisition Cost and New Acquisition Cost
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message

Table 15: Create an Account Log entry if the Posted Indicator (acquisition_cost_posted_ind) is checked
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here:

Column Name	Data Type	Comments
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 16: Create an Account Log entry if the Record Loss Indicator is checked
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here:

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Asset Gain/Loss Account
EXPENSE_CODE_IN	VARCHAR2	Asset Gain/Loss Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	‘AG’
TRANSACTION_AMOUNT_IN	NUMBER	Absolute Value of the Difference between Old Acquisition Cost and New Acquisition Cost
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number

Table 16: Create an Account Log entry if the Record Loss Indicator is checked
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here:

Column Name	Data Type	Comments
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 17: Create Account Log entries if the Asset Cost Account is changed
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here:

(1 of 2)

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Old Asset Cost Account
EXPENSE_CODE_IN	VARCHAR2	Old Asset Cost Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset

Table 17: Create Account Log entries if the Asset Cost Account is changed
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here:

(1 of 2)

Column Name	Data Type	Comments
TRANSACTION_TYPE_IN	VARCHAR2	'AC'
TRANSACTION_AMOUNT_IN	NUMBER	(Acquisition Cost + Total Value Adjustment from Fixed Asset table) * -1
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 18: Create Account Log entries if the Asset Cost Account is changed
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here:

(2 of 2)

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	New Asset Cost Account, if null, use Old Asset Cost Account
EXPENSE_CODE_IN	VARCHAR2	New Asset Cost Expense Code, if null, use Old Asset Cost Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table

Table 18: Create Account Log entries if the Asset Cost Account is changed
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here:

(2 of 2)

Column Name	Data Type	Comments
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'AC'
TRANSACTION_AMOUNT_IN	NUMBER	Acquisition Cost + Total Value Adjustment from Fixed Asset table
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 19: Create Account Log entries if the Accum Depreciation Account is changed
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here

(1 of 2):

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Old Accum Dep. Account

Table 19: Create Account Log entries if the Accum Depreciation Account is changed
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here (1 of 2):

Column Name	Data Type	Comments
EXPENSE_CODE_IN	VARCHAR2	Old Accum Dep. Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'DA'
TRANSACTION_AMOUNT_IN	NUMBER	Accumulated Depreciation of the asset
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 20: Create Account Log entries if the Accum Depreciation Account is changed
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here
(2 of 2):

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	New Accum Dep. Account, if null, use Old Accum Dep. Account
EXPENSE_CODE_IN	VARCHAR2	New Accum Dep. Expense Code, if null, use Old Accum Dep. Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'DA'
TRANSACTION_AMOUNT_IN	NUMBER	(Accumulated Depreciation of the asset) * -1
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message

Table 20: Create Account Log entries if the Accum Depreciation Account is changed
Call sdbp_batch_depreciation.insert_acct_log and pass the parameters listed here

(2 of 2):

Column Name	Data Type	Comments
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 21: Update SA_ASSET_FIXED_ASSET plant, asset_id and asset_record_type as qualifiers.
Update the following fields:

Column Name	Data Type	Comments
DEP_ACCOUNT_NO	VARCHAR2	
DEP_EXPENSE_CODE	VARCHAR2	
ACCUM_DEP_ACCOUNT	VARCHAR2	
ACCUM_DEP_EXPENSE_CODE	VARCHAR2	
ASSET_COST_ACCOUNT	VARCHAR2	
ASSET_COST_EXPENSE_CODE	VARCHAR2	
ASSET_GAIN_LOSS_ACCOUNT	VARCHAR2	
ASSET_GAIN_LOSS_EXPENSE	VARCHAR2	
TOTAL_VALUE_ADJUSTMENT	NUMBER	Null
PROPERTY_UNIT_NO	VARCHAR2	
REPLACEMENT_VALUE	NUMBER	
USEFUL_LIFE_UNITS	VARCHAR2	
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 22: Create a record in SA_CHANGE_REQUEST_ASSET_LOG if there is a change in Acquisition Cost, Useful Life and/or Salvage Value.
Populate the following:

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
EFFECTIVE_DATE	DATE	Effective Date of the Change Request
PERIOD_YEAR	VARCHAR2	Accounting Period Year of Effective Date
PERIOD_MONTH	VARCHAR2	Accounting Period Month of Effective Date
VALUE_ADJUSTMENT	NUMBER	Difference between the New Asset Cost and Old Asset Cost
SALVAGE_VALUE	NUMBER	New Salvage Value
USEFUL_LIFE	NUMBER	New Useful Life
USEFUL_LIFE_ADJUSTMENT	NUMBER	Difference between the New Useful Life and Old Useful Life
CREATED_DATE	DATE	Default to SYSDATE
CREATED_BY	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 23: Create an Account Log entry.

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Asset Cost Account
EXPENSE_CODE_IN	VARCHAR2	Asset Cost Expense Code

Table 23: Create an Account Log entry.

Column Name	Data Type	Comments
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'AC'
TRANSACTION_AMOUNT_IN	NUMBER	Difference between Old Acquisition Cost and New Acquisition Cost
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL

Table 24: Create an Account Log entry if the Record Loss Indicator is **checked**.

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Asset Gain/Loss Account
EXPENSE_CODE_IN	VARCHAR2	Asset Gain/Loss Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset

Table 24: Create an Account Log entry if the Record Loss Indicator is **checked**.

Column Name	Data Type	Comments
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'AG'
TRANSACTION_AMOUNT_IN	NUMBER	Absolute Value of the difference between Old Acquisition Cost and New Acquisition Cost
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 25: Create Account Log entries if the Asset Cost Account is changed (1 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Old Asset Cost Account
EXPENSE_CODE_IN	VARCHAR2	Old Asset Cost Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table

Table 25: Create Account Log entries if the Asset Cost Account is changed (1 of 2).

Column Name	Data Type	Comments
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'AC'
TRANSACTION_AMOUNT_IN	NUMBER	(Acquisition Cost + Total Value Adjustment from Fixed Asset table) * -1
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 26: Create Account Log entries if the Asset Cost account is changed (2 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	New Asset Cost Account, if null, use Old Asset Cost Account
EXPENSE_CODE_IN	VARCHAR2	New Asset Cost Expense Code, if null, use Old Asset Cost Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table

Table 26: Create Account Log entries if the Asset Cost account is changed (2 of 2).

Column Name	Data Type	Comments
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	‘AC’
TRANSACTION_AMOUNT_IN	NUMBER	Acquisition Cost + Total Value Adjustment from Fixed Asset table
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 27: Create Account Log entries if the Accum Depreciation account is changed (1 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Old Accum Dep. Account
EXPENSE_CODE_IN	VARCHAR2	Old Accum Dep. Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table

Table 27: Create Account Log entries if the Accum Depreciation account is changed (1 of 2).

Column Name	Data Type	Comments
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	‘DA’
TRANSACTION_AMOUNT_IN	NUMBER	Accumulated Depreciation of the asset
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 28: Create Account Log entries if the Accum Depreciation account is changed (2 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	New Accum Dep. Account, if null, use Old Accum Dep. Account
EXPENSE_CODE_IN	VARCHAR2	New Accum Dep. Expense Code, if null, use Old Accum Dep. Expense Code

Table 28: Create Account Log entries if the Accum Depreciation account is changed (2 of 2).

Column Name	Data Type	Comments
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'DA'
TRANSACTION_AMOUNT_IN	NUMBER	(Accumulated Depreciation of the asset) * -1
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Change Request Status Date
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 29: Create Account Log entries to reverse the posted depreciation forecast's transaction amounts (1 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Dep. Expense Account
EXPENSE_CODE_IN	VARCHAR2	Dep. Expense Code

Table 29: Create Account Log entries to reverse the posted depreciation forecast's transaction amounts (1 of 2).

Column Name	Data Type	Comments
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Written Down Value for the Year and Month being reversed
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation for the Year and Month being reversed
TRANSACTION_TYPE_IN	VARCHAR2	'DP'
TRANSACTION_AMOUNT_IN	NUMBER	Positive value of Accumulated Depreciation for the Year and Month being reversed
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Accounting Period End Date of the Year and Month being reversed
TRANSACTION_DATE_IN	DATE	SYSDATE
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 30: Create Account Log entries to reverse the posted depreciation forecast's transaction amounts (2 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	

Table 30: Create Account Log entries to reverse the posted depreciation forecast's transaction amounts (2 of 2).

Column Name	Data Type	Comments
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Dep. Expense Account
EXPENSE_CODE_IN	VARCHAR2	Dep. Expense Code
PERIOD_YEAR_IN	VARCHAR2	Year of Change Request Status Date from Accounting Periods table
PERIOD_MONTH_IN	VARCHAR2	Month of Change Request Status Date from Accounting Periods table
WRITTEN_DOWN_VALUE_IN	NUMBER	Written Down Value for the Year and Month being reversed
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation for the Year and Month being reversed
TRANSACTION_TYPE_IN	VARCHAR2	'DA'
TRANSACTION_AMOUNT_IN	NUMBER	Negative value of Accumulated Depreciation for the Year and Month being reversed
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Accounting Period End Date of the Year and Month being reversed
TRANSACTION_DATE_IN	DATE	SYSDATE
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 31: Fields to update in Fixed Asset.

Column Name	Data Type	Comments
REMAINING_LIFE_YEAR	NUMBER	
REMAINING_LIFE_PERIOD	NUMBER	
CURRENT_BOOK_VALUE	NUMBER	Net Book Value of the asset for the last posted Year and Month. Retrieve from Asset Forecast table.
ACCUM_DEPRECIATION	NUMBER	Accumulated Dep of the asset for the last posted Year and Month. Retrieve from Asset Forecast table.
LAST_UPDATE_USER	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE

Table 32: Update SA_ASSET_FIXED_ASSET.

Column Name	Data Type	Comments
DEP_ACCOUNT_NO	VARCHAR2	
DEP_EXPENSE_CODE	VARCHAR2	
ACCUM_DEP_ACCOUNT	VARCHAR2	
ACCUM_DEP_EXPENSE_CODE	VARCHAR2	
ASSET_COST_ACCOUNT	VARCHAR2	
ASSET_COST_EXPENSE_CODE	VARCHAR2	
ASSET_GAIN_LOSS_ACCOUNT	VARCHAR2	
ASSET_GAIN_LOSS_EXPENSE	VARCHAR2	
PROPERTY_UNIT_NO	VARCHAR2	
REPLACEMENT_VALUE	NUMBER	
USEFUL_LIFE_UNITS	VARCHAR2	
LAST_UPDATE_DATE	DATE	Default to SYSDATE

Table 32: Update SA_ASSET_FIXED_ASSET.

Column Name	Data Type	Comments
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 33: Create a record in SA_CHANGE_REQUEST_ASSET_LOG.

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
EFFECTIVE_DATE	DATE	Effective Date of the Change Request
PERIOD_YEAR	VARCHAR2	Accounting Period Year of Effective Date
PERIOD_MONTH	VARCHAR2	Accounting Period Month of Effective Date
VALUE_ADJUSTMENT	NUMBER	Difference between the New Asset Cost and Old Asset Cost
SALVAGE_VALUE	NUMBER	New Salvage Value
USEFUL_LIFE	NUMBER	New Useful Life
USEFUL_LIFE_ADJUSTMENT	NUMBER	Difference between the New Useful Life and Old Useful Life
CREATED_DATE	DATE	Default to SYSDATE
CREATED_BY	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 34: Create Account Log entries to post the forecast record (1 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Dep. Expense Account

Table 34: Create Account Log entries to post the forecast record (1 of 2).

Column Name	Data Type	Comments
EXPENSE_CODE_IN	VARCHAR2	Dep. Expense Code
PERIOD_YEAR_IN	VARCHAR2	Prior Year
PERIOD_MONTH_IN	VARCHAR2	Period of the forecast that is being posted
WRITTEN_DOWN_VALUE_IN	NUMBER	Written Down Value for the Year and Month being posted
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation for the Year and Month being posted
TRANSACTION_TYPE_IN	VARCHAR2	'DP'
TRANSACTION_AMOUNT_IN	NUMBER	Positive value of Accumulated Depreciation for the Year and Month being posted
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Accounting Period End Date of the Year and Month being posted
TRANSACTION_DATE_IN	DATE	SYSDATE
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 35: Create Account Log entries to post the forecast record (2 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Dep. Expense Account
EXPENSE_CODE_IN	VARCHAR2	Dep. Expense Code

Table 35: Create Account Log entries to post the forecast record (2 of 2).

Column Name	Data Type	Comments
PERIOD_YEAR_IN	VARCHAR2	Prior Year
PERIOD_MONTH_IN	VARCHAR2	Period of the forecast that is being posted
WRITTEN_DOWN_VALUE_IN	NUMBER	Written Down Value for the Year and Month being posted
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation for the Year and Month being posted
TRANSACTION_TYPE_IN	VARCHAR2	'DA'
TRANSACTION_AMOUNT_IN	NUMBER	Negative value of Accumulated Depreciation for the Year and Month being posted
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Accounting Period End Date of the Year and Month being posted
TRANSACTION_DATE_IN	DATE	SYSDATE
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 36: Fields to update in Fixed Asset.

Column Name	Data Type	Comments
CURRENT_BOOK_VALUE	NUMBER	New Written Down Value of the Year and Month being posted
ACCUM_DEPRECIATION	NUMBER	New Accumulated Depreciation amount of the Year and Month being posted
DEPRECIATION_RATE	NUMBER	New Depreciation Rate

Table 36: Fields to update in Fixed Asset.

Column Name	Data Type	Comments
REMAINING_LIFE_YEAR	NUMBER	
REMAINING_LIFE_PERIOD	NUMBER	
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 37: Create an Account Log entry to insert Asset Cost Write-Off.

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Asset Cost Account
EXPENSE_CODE_IN	VARCHAR2	Asset Cost Expense Code
PERIOD_YEAR_IN	VARCHAR2	Period Year from Item 1 Disposal section
PERIOD_MONTH_IN	VARCHAR2	Period Month from Item 1 Disposal section
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'AW'
TRANSACTION_AMOUNT_IN	NUMBER	Item 3.1.3 Disposal section
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Transaction Date from Item 1 Disposal section

Table 37: Create an Account Log entry to insert Asset Cost Write-Off.

Column Name	Data Type	Comments
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 38: Create an Account Log entry to compute Asset Gain/Loss Amount.

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Asset Gain/Loss Account
EXPENSE_CODE_IN	VARCHAR2	Asset Gain/Loss Expense Code
PERIOD_YEAR_IN	VARCHAR2	Period Year from Item 1 Disposal section
PERIOD_MONTH_IN	VARCHAR2	Period Month from Item 1 Disposal section
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'AG'
TRANSACTION_AMOUNT_IN	NUMBER	Item 4.1.3 Disposal section
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Transaction Date from Item 1 Disposal section
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message

Table 38: Create an Account Log entry to compute Asset Gain/Loss Amount.

Column Name	Data Type	Comments
ERROR_NO	NUMBER	Out parameter, error number
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 39: Create an Account Log entry to insert Asset Accumulated Depreciation Account.

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Accum Depreciation Account
EXPENSE_CODE_IN	VARCHAR2	Accum Depreciation Expense Code
PERIOD_YEAR_IN	VARCHAR2	Period Year from Item 1 Disposal section
PERIOD_MONTH_IN	VARCHAR2	Period Month from Item 1 Disposal section
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'DA'
TRANSACTION_AMOUNT_IN	NUMBER	Item 5.1 Disposal section
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Transaction Date from Item1 Disposal section
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number

Table 39: Create an Account Log entry to insert Asset Accumulated Depreciation Account.

Column Name	Data Type	Comments
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 40: Create Account Log entries if there is an Asset Cost Adjustment in the change Request (1 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Accum Depreciation Account
EXPENSE_CODE_IN	VARCHAR2	Accum Depreciation Expense Code
PERIOD_YEAR_IN	VARCHAR2	Period Year from Item 1 Disposal section
PERIOD_MONTH_IN	VARCHAR2	Period Month from Item 1 Disposal section
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	‘AC’
TRANSACTION_AMOUNT_IN	NUMBER	Asset Cost Adjustment from the Change Request.
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Transaction Date from Item1 Disposal section
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number

Table 40: Create Account Log entries if there is an Asset Cost Adjustment in the change Request (1 of 2).

Column Name	Data Type	Comments
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 41: Create Account Log entries if there is an Asset Cost Adjustment in the change Request (2 of 2).

Column Name	Data Type	Comments
PLANT_IN	VARCHAR2	
ASSET_ID_IN	VARCHAR2	
ASSET_RECORD_TYPE_IN	VARCHAR2	
ACCOUNT_NO_IN	VARCHAR2	Accum Depreciation Account
EXPENSE_CODE_IN	VARCHAR2	Accum Depreciation Expense Code
PERIOD_YEAR_IN	VARCHAR2	Period Year from Item 1 Disposal section
PERIOD_MONTH_IN	VARCHAR2	Period Month from Item 1 Disposal section
WRITTEN_DOWN_VALUE_IN	NUMBER	Current Book Value of the asset
ACCUMULATED_DEPRECIATION_IN	NUMBER	Accumulated Depreciation of the asset
TRANSACTION_TYPE_IN	VARCHAR2	'AG'
TRANSACTION_AMOUNT_IN	NUMBER	Asset Cost Adjustment from the Change Request * -1
ASSET_CHANGE_REQUEST_NO_IN	VARCHAR2	Change Request Number
ASSET_CHANGE_REQUEST_SEQ_NO_IN	NUMBER	Asset Change Sequence Number
ASSET_CHANGE_TRANS_TYPE_IN	VARCHAR2	NULL
ASSET_CHANGE_EFFECTIVE_DATE_IN	DATE	Effective Date. If it is null, use Transaction Date.
TRANSACTION_DATE_IN	DATE	Transaction Date from Item1 Disposal section
DBMS_ACTIVITY	VARCHAR2	Out parameter, tracing message
ERROR_NO	NUMBER	Out parameter, error number

Table 41: Create Account Log entries if there is an Asset Cost Adjustment in the change Request (2 of 2).

Column Name	Data Type	Comments
ERROR_MSG	VARCHAR2	Out parameter, error message

Table 42: Fields to update in Asset Forecast.

Column Name	Data Type	Comments
WRITTE_DOWN_VALUE	NUMBER	
ACCUM_DEPRECIATION	NUMBER	
DEPRECIATION_EXPENSE	NUMBER	
VALUE_ADJSUTMENT	NUMBER	
LAST_UPDATE_USER	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE

Table 43: Fields to update in Fixed Asset.

Column Name	Data Type	Comments
CURRENT_BOOK_VALUE	NUMBER	0
ACCUM_DEPRECIATION	NUMBER	0
TOTAL_VALUE_ADJUSTMENT	NUMBER	Sum of Total Value Adjustment and Cost Adjustment from Change Request
REMAINING_LIFE_YEAR	NUMBER	0
REMAINING_LIFE_PERIOD	NUMBER	0
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

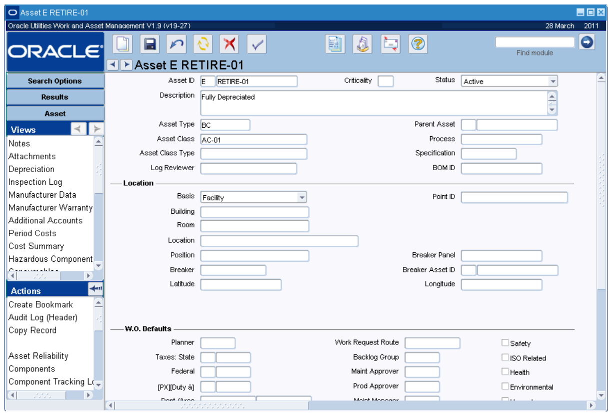
Appendix B: Performing Custom Depreciation Phase 3

Disposal: Fully Depreciated Asset

Follow these steps to dispose of a fully depreciated asset.

Create the Asset and Set Depreciation Settings

1. Create an asset. Make sure that the Asset Class and its Depreciation Parameters are set-up.



Asset E RETIRE-01

Asset ID	E RETIRE-01
Criticality	
Status	Active
Description	Fully Depreciated
Asset Type	BC
Parent Asset	
Asset Class	AC-01
Process	
Asset Class Type	
Specification	
Log Reviewer	
BOM ID	

Location

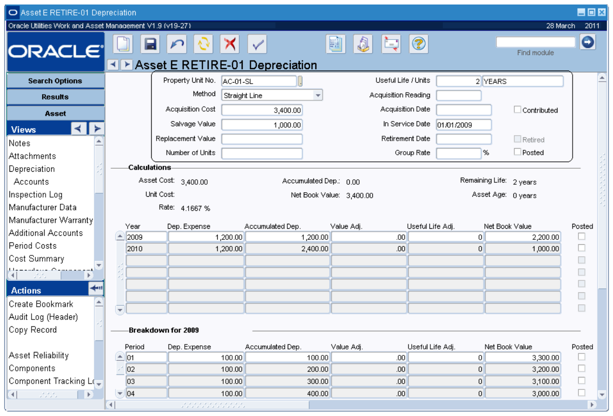
Basis	Facility
Point ID	
Building	
Room	
Location	
Position	
Breaker Panel	
Breaker	
Breaker Asset ID	
Latitude	
Longitude	

W.O. Defaults

Planner	
Work Request Route	
Taxes: State	
Backlog Group	
Federal	
Maint Approver	
[PX][Duty ä]	
Prod Approver	

☐ Safety ☐ ISO Related ☐ Health ☐ Environmental

2. Navigate to the Depreciation view to create a Fixed Asset record. Populate the Fixed Asset based on the screen below. Upon Save, the Calculations and Breakdown sections are automatically populated.



Asset E RETIRE-01 Depreciation

Property Unit No.	AC-01-SL
Useful Life / Units	2 YEARS
Method	Straight Line
Acquisition Reading	
Acquisition Cost	3,400.00
Acquisition Date	
Salvage Value	1,000.00
In Service Date	01/01/2009
Replacement Value	
Retirement Date	
Number of Units	
Group Rate	%

☐ Contributed ☐ Retired ☐ Posted

Calculations

Asset Cost:	3,400.00
Accumulated Dep.:	0.00
Remaining Life:	2 years
Unit Cost:	
Net Book Value:	3,400.00
Asset Age:	0 years
Rate:	4.1667 %

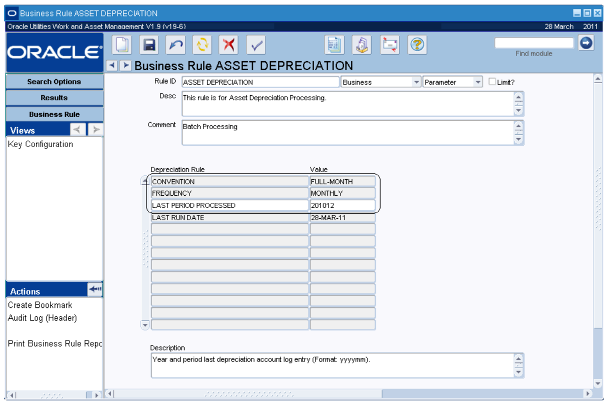
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2009	1,200.00	1,200.00	.00	0	2,200.00	☐
2010	1,200.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2009

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	100.00	.00	0	3,300.00	☐
02	100.00	200.00	.00	0	3,200.00	☐
03	100.00	300.00	.00	0	3,100.00	☐
04	100.00	400.00	.00	0	3,000.00	☐

Modify the Asset Depreciation Business Rule and Run Batch

1. Change the ASSET DEPRECIATION business rule. Set LAST PERIOD PROCESSED, CONVENTION and FREQUENCY based on the screen below.



Business Rule ASSET DEPRECIATION

Rule ID: ASSET DEPRECIATION | Business | Parameter | Limit? ☐

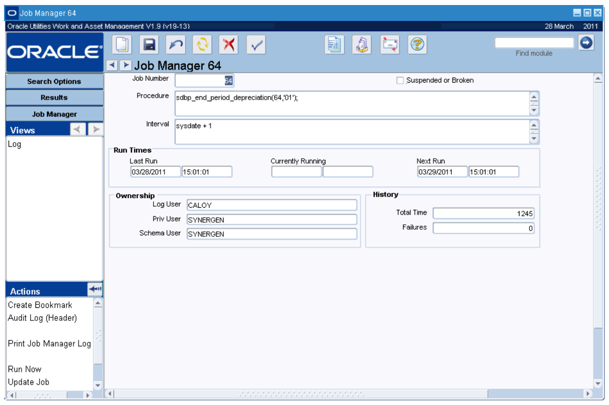
Desc: This rule is for Asset Depreciation Processing.

Comment: Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201012
LAST RUN DATE	28-MAR-11

Description: Year and period last depreciation account log entry (Format: yyyymm).

2. Run the batch job sdbp_end_period_depreciation. The batch job picks-up the LAST PERIOD PROCESSED value and processes the next period, which is Year 2011 Period.



Job Manager 64

Job Number: 64 | Suspended or Broken ☐

Procedure: sdbp_end_period_depreciation(64,'01');

Interval: sysdate + 1

Run Times

Last Run		Currently Running		Next Run	
03/28/2011	15:01:01			03/29/2011	15:01:01

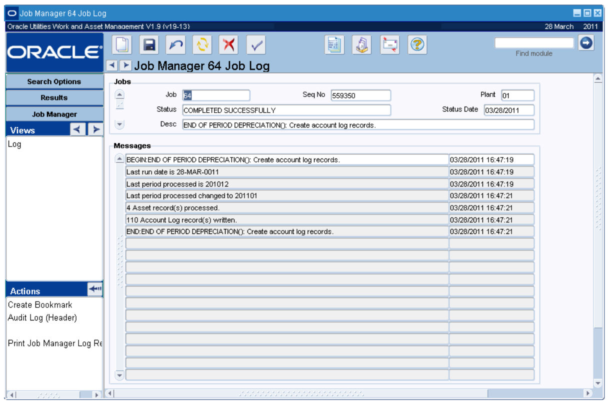
Ownership

Log User: CALOY
Priv User: SYNERGEN
Schema User: SYNERGEN

History

Total Time: 1245
Failures: 0

3. Make sure that the batch job ran successfully.



Job Manager 64 Job Log

Jobs

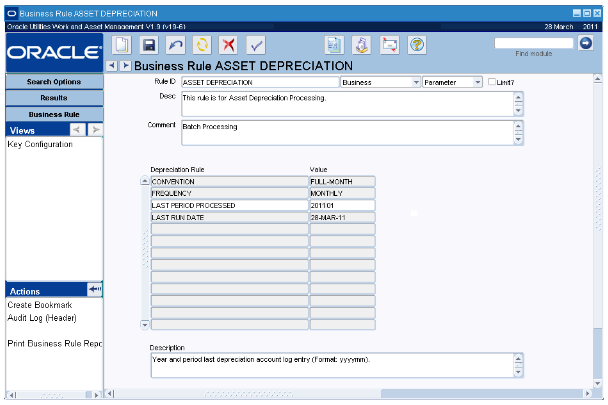
Job: 64 | Seq No: 559350 | Plant: 01
Status: COMPLETED SUCCESSFULLY | Status Date: 03/28/2011
Desc: END OF PERIOD DEPRECIATION(): Create account log records.

Messages

BEGIN:END OF PERIOD DEPRECIATION(): Create account log records.	03/28/2011 16:47:19
Last run date is 28-MAR-0011	03/28/2011 16:47:19
Last period processed is 201012	03/28/2011 16:47:19
Last period processed changed to 201101	03/28/2011 16:47:21
4 Asset record(s) processed.	03/28/2011 16:47:21
110 Account Log record(s) written.	03/28/2011 16:47:21
END:END OF PERIOD DEPRECIATION(): Create account log records.	03/28/2011 16:47:21

Verify that the Asset was Processed Correctly

1. Navigate back to the ASSET DEPRECIATION Business Rule.
   Make sure that the LAST PERIOD PROCESSED is incremented by one period and the LAST RUN DATE is updated with Sysdate.



Business Rule ASSET DEPRECIATION

Rule ID: ASSET DEPRECIATION | Business | Parameter | Limit? ☐

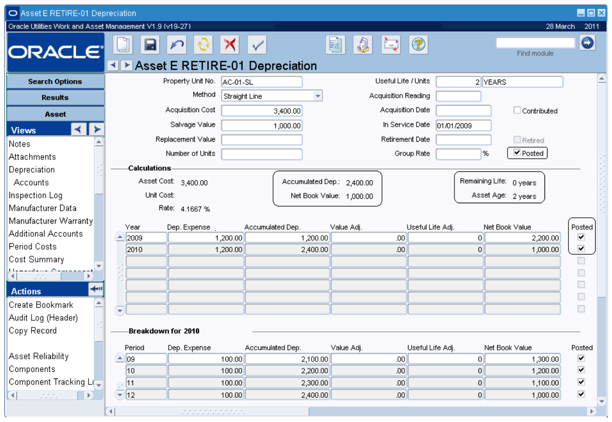
Desc: This rule is for Asset Depreciation Processing.

Comment: Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201101
LAST RUN DATE	28-MAR-11

Description: Year and period last depreciation account log entry (Format: yyyymm).

2. Navigate back to the Depreciation view of the Asset module.
   The Posted boxes for both years are checked. The Remaining Life is zero and the Asset Age equals Useful Life. The Accumulated Dep and Net Book Value are the same as the values for Year 2010 Period 12. The asset is fully depreciated.



Asset E RETIRE-01 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/01/2009
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ✓ Posted

Calculations

Asset Cost: 3,400.00
Unit Cost:
Rate: 4.1667 %

Accumulated Dep.: 2,400.00
Net Book Value: 1,000.00

Remaining Life: 0 years
Asset Age: 2 years

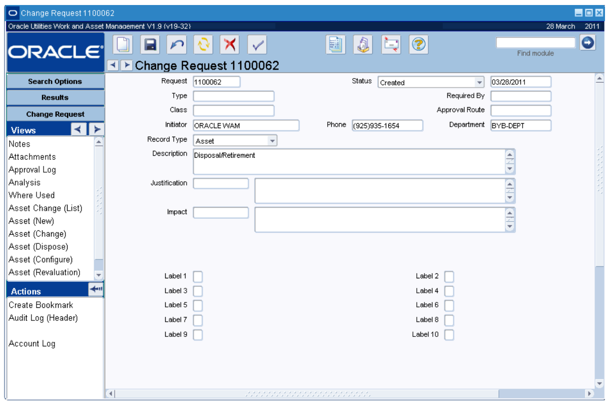
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2009	1,200.00	1,200.00	.00	0	2,200.00	✓
2010	1,200.00	2,400.00	.00	0	1,000.00	✓

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	2,100.00	.00	0	1,300.00	✓
10	100.00	2,200.00	.00	0	1,200.00	✓
11	100.00	2,300.00	.00	0	1,100.00	✓
12	100.00	2,400.00	.00	0	1,000.00	✓

Create a Change Request

1. Create a Change Request with Asset as Record Type.

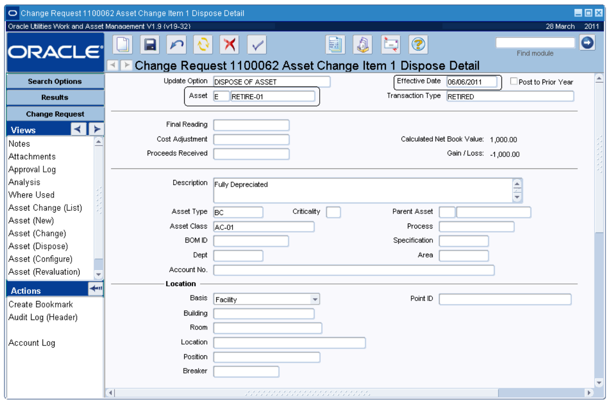

Change Request 1100062

Oracle Utilities Work and Asset Management V1.9 {v19-32} 28 March 2011

Change Request 1100062

Request: 1100062
Status: Created 03/28/2011
Initiator: ORACLE WAM
Phone: (925)935-1654
Department: BYB-DEPT
Record Type: Asset
Description: Disposal/Retirement

2. Navigate to the Asset (Dispose) view and enter the asset that was created in the first step in this procedure above.
Create a record based on the screen below.

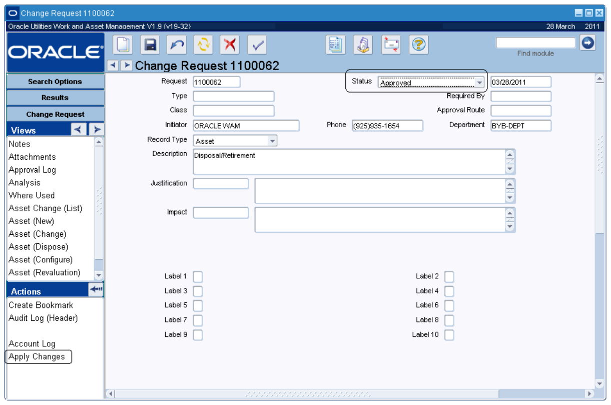

Change Request 1100062 Asset Change Item 1 Dispose Detail

Oracle Utilities Work and Asset Management V1.9 {v19-32} 28 March 2011

Change Request 1100062 Asset Change Item 1 Dispose Detail

Update Option: DISPOSE OF ASSET
Asset: E RETIRE-01
Effective Date: 06/06/2011
Post to Prior Year: ☐
Transaction Type: RETIRED
Calculated Net Book Value: 1,000.00
Gain / Loss: -1,000.00
Description: Fully Depreciated
Asset Type: BC
Asset Class: AC-01
Location
Basis: Facility

3. Navigate back to the Change Request header and change the status from Created to Approved.
The Apply Changes action opens.


Change Request 1100062

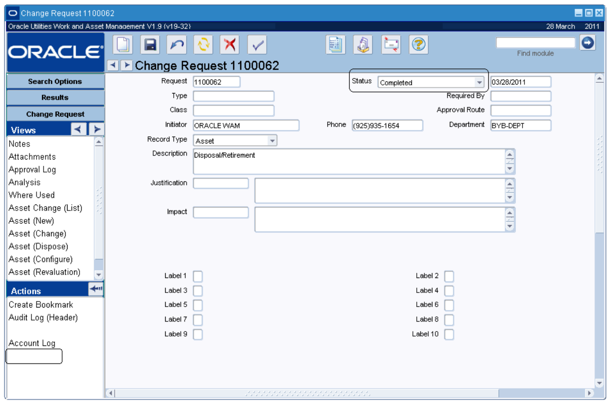
Oracle Utilities Work and Asset Management V1.9 {v19-32} 28 March 2011

Change Request 1100062

Request: 1100062
Status: Approved 03/28/2011
Initiator: ORACLE WAM
Phone: (925)935-1654
Department: BYB-DEPT
Record Type: Asset
Description: Disposal/Retirement

4. Click on the Apply Changes action.

The status changes from Approved to Completed and Apply Changes action disappears.



Change Request 1100062
Oracle Utilities Work and Asset Management V1.9 (v19-32) 28 March 2011

Change Request 1100062

Request	1100062	Status	Completed		03/28/2011
Type				Required By	
Class				Approval Route	
Initiator	ORACLE WAM	Phone	(925)935-1654	Department	BYB-DEPT
Record Type	Asset				
Description	Disposal/Retirement				
Justification					
Impact					

5. Click on the Account Log action from the Change Request header screen. The Account Log module is displayed.

   5.1 Transaction Date is the Change Request Status Date.

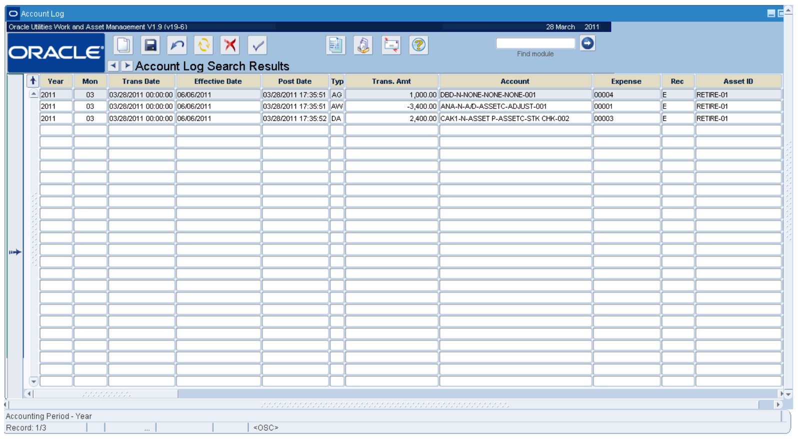
   5.2 Year and Month is the Year and Period of the Transaction Date in Accounting Period table.

   5.3 Effective Date is the Change Request item Effective Date.

   5.4 Post Date is Sysdate.

   5.5 Since there are no cost adjustments to the asset, the Trans. Amt for the Asset Cost Write-off (AW) is the negative value of the asset's Acquisition Cost.

   5.6 Since the asset is fully depreciated and there are no cost adjustments, the derived Written Down value of the asset is -$1,000. Deducting the derived value from the Proceeds Received, which is zero, the Trans.Amt for AG (Asset Gain/Loss) is +$1.000. The Accumulated Depreciation (DA) is $2,400, equal to the Accumulated Dep. of the last forecast record of the asset.



Account Log
Oracle Utilities Work and Asset Management V1.9 (v19-6) 28 March 2011

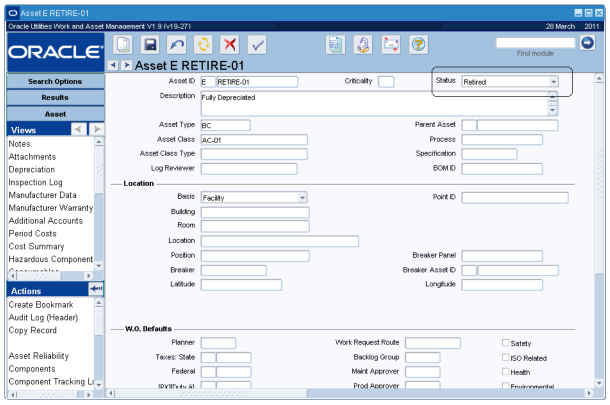
Account Log Search Results

Year	Mon	Trans Date	Effective Date	Post Date	Typ	Trans. Amt	Account	Expense	Rec	Asset ID
2011	03	03/28/2011 00:00:00	06/06/2011	03/28/2011 17:35:51	AG	1,000.00	DBD-N-NONE-NONE-NONE-001	00004	E	RETIRE-01
2011	03	03/28/2011 00:00:00	06/06/2011	03/28/2011 17:35:51	AW	-3,400.00	ANA-N-A/D-ASSETC-ADJUST-001	00001	E	RETIRE-01
2011	03	03/28/2011 00:00:00	06/06/2011	03/28/2011 17:35:52	DA	2,400.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	00003	E	RETIRE-01

Accounting Period - Year
Record: 1/3 ... <OSC>

Verify Revised Cost Calculations

1. Navigate back to the Asset module and retrieve the asset created in the first step in this procedure. The asset’s status is now Retired.



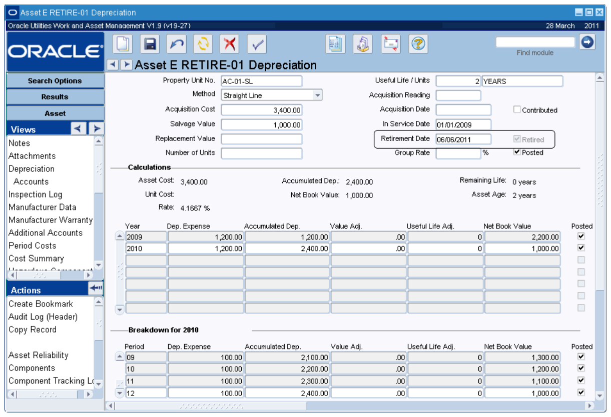
Asset E RETIRE-01

Asset ID: E RETIRE-01
Criticality:
Status: Retired
Description: Fully Depreciated
Asset Type: BC
Parent Asset:
Asset Class: AC-01
Process:
Asset Class Type:
Specification:
Log Reviewer:
BOM ID:

Location
Basis: Facility
Point ID:
Building:
Room:
Location:
Position:
Breaker Panel:
Breaker:
Breaker Asset ID:
Latitude:
Longitude:

W.O. Defaults
Planner:
Work Request Route:
Safety
Taxes: State:
Backlog Group:
ISO Related
Federal:
Maint Approver:
Health
[PX][Duty Ä]:
Prod Approver:
Environmental

2. Navigate to the Depreciation view. The retirement Date is now populated with the Effective Date from the Change Request. The Retired box is also checked.



Asset E RETIRE-01 Depreciation

Property Unit No.: AC-01-SL
Useful Life / Units: 2 YEARS
Method: Straight Line
Acquisition Reading:
Acquisition Cost: 3,400.00
Acquisition Date:
Contributed
Salvage Value: 1,000.00
In Service Date: 01/01/2009
Replacement Value:
Retirement Date: 06/06/2011
✓ Retired
Number of Units:
Group Rate: %
✓ Posted

Calculations
Asset Cost: 3,400.00
Accumulated Dep.: 2,400.00
Remaining Life: 0 years
Unit Cost:
Net Book Value: 1,000.00
Asset Age: 2 years
Rate: 4.1667 %

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2009	1,200.00	1,200.00	.00	0	2,200.00	✓
2010	1,200.00	2,400.00	.00	0	1,000.00	✓

Breakdown for 2010

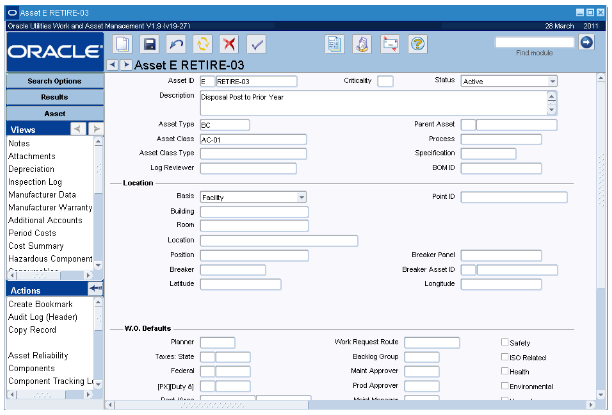
Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	2,100.00	.00	0	1,300.00	✓
10	100.00	2,200.00	.00	0	1,200.00	✓
11	100.00	2,300.00	.00	0	1,100.00	✓
12	100.00	2,400.00	.00	0	1,000.00	✓

Disposal: Post to Prior Year

Follow these steps to dispose of an asset and post costs to the prior year.

Create the Asset and Set Depreciation Settings

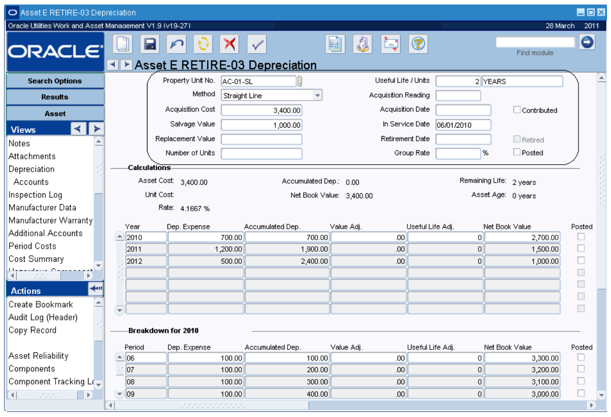
1. Create an asset. Make sure that the Asset Class and its Depreciation Parameters are set-up.



Asset E RETIRE-03

Asset ID: E RETIRE-03
Status: Active
Description: Disposal Post to Prior Year
Asset Type: BC
Asset Class: AC-01
Basis: Facility

2. Navigate to the Depreciation view to create a Fixed Asset record. Populate Fixed Asset based on the screen below. Upon Save, the Calculations and Breakdown sections are automatically populated.



Asset E RETIRE-03 Depreciation

Property Unit No.: AC-01-SL
Method: Straight Line
Acquisition Cost: 3,400.00
Salvage Value: 1,000.00
Useful Life / Units: 2 YEARS
In Service Date: 06/01/2010

Calculations

Asset Cost: 3,400.00
Unit Cost:
Rate: 4.1667 %
Accumulated Dep.: 0.00
Net Book Value: 3,400.00
Remaining Life: 2 years
Asset Age: 0 years

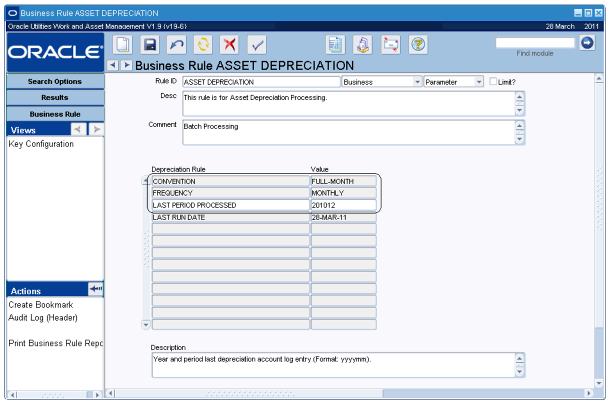
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	700.00	700.00	.00	0	2,700.00	☐
2011	1,200.00	1,900.00	.00	0	1,500.00	☐
2012	500.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
06	100.00	100.00	.00	0	3,300.00	☐
07	100.00	200.00	.00	0	3,200.00	☐
08	100.00	300.00	.00	0	3,100.00	☐
09	100.00	400.00	.00	0	3,000.00	☐

Modify the Asset Depreciation Business Rule and Run Batch

1. Change the Business Rule ASSET DEPRECIATION. Set LAST PERIOD PROCESSED, CONVENTION and FREQUENCY based on the screen below.



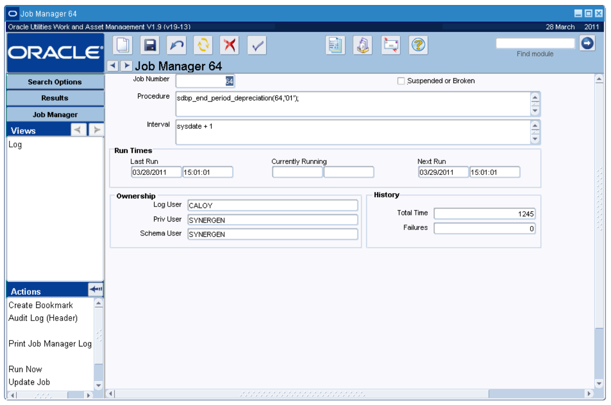
Business Rule ASSET DEPRECIATION

Rule ID: ASSET DEPRECIATION Business Parameter
Desc: This rule is for Asset Depreciation Processing.
Comment: Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201012
LAST RUN DATE	28-MAR-11

Description: Year and period last depreciation account log entry (Format: yyyymm).

2. Run the batch job sdbp_end_period_depreciation. The batch job picks-up the LAST PERIOD PROCESSED value and processes the next period, which is Year 2011 Period.



Job Manager 64

Job Number 64 | Suspended or Broken
Procedure sdbp_end_period_depreciation(64,'01');
Interval sysdate + 1

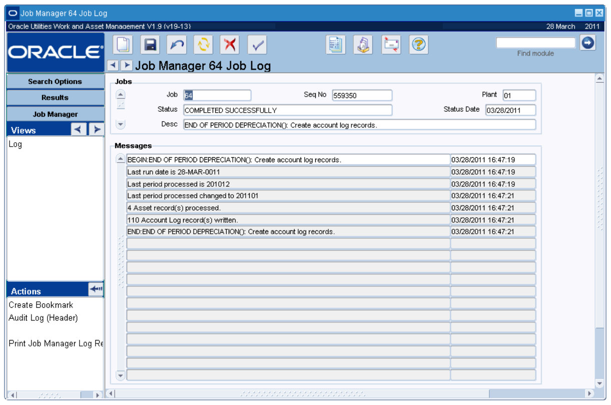
Run Times

Last Run		Currently Running		Next Run	
03/28/2011	15:01:01			03/29/2011	15:01:01

Ownership
Log User CALOY
Priv User SYNERGEN
Schema User SYNERGEN

History
Total Time 1245
Failures 0

3. Make sure that the batch job ran successfully.



Job Manager 64 Job Log

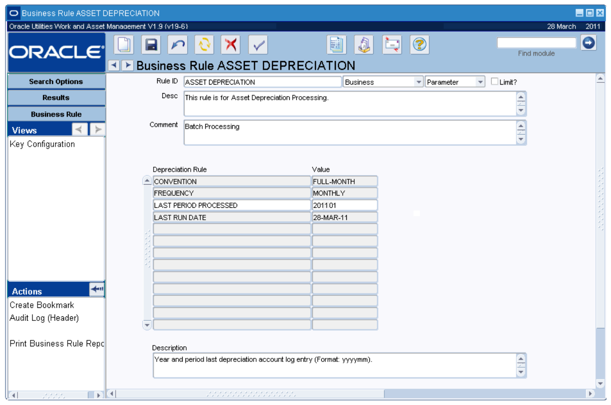
Jobs
Job 64 | Seq No 559350 | Plant 01
Status COMPLETED SUCCESSFULLY | Status Date 03/28/2011
Desc END OF PERIOD DEPRECIATION(): Create account log records.

Messages

BEGIN:END OF PERIOD DEPRECIATION(): Create account log records.	03/28/2011 16:47:19
Last run date is 28-MAR-0011	03/28/2011 16:47:19
Last period processed is 201012	03/28/2011 16:47:19
Last period processed changed to 201101	03/28/2011 16:47:21
4 Asset record(s) processed.	03/28/2011 16:47:21
110 Account Log record(s) written.	03/28/2011 16:47:21
END:END OF PERIOD DEPRECIATION(): Create account log records.	03/28/2011 16:47:21

Verify that the Asset was Processed Correctly

1. Navigate back to the ASSET DEPRECIATION Business Rule.
   Make sure that the LAST PERIOD PROCESSED is incremented by one period and the LAST RUN DATE is updated with Sysdate.



Business Rule ASSET DEPRECIATION

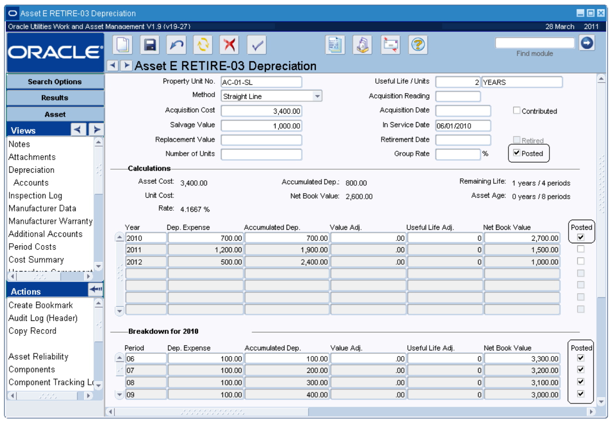
Rule ID ASSET DEPRECIATION | Business | Parameter | Limit?
Desc This rule is for Asset Depreciation Processing.
Comment Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201101
LAST RUN DATE	28-MAR-11

Description
Year and period last depreciation account log entry (Format: yyyymm).

2. Navigate back to the Depreciation view of the Asset module.

Note the Posted check boxes. If it is checked, it means that the asset has been processed for the end of period and Account Log entries have been written against the asset. All the forecast records for the Year 2010 are posted.



Asset E RETIRE-03 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	06/01/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ☑ Posted

Calculations

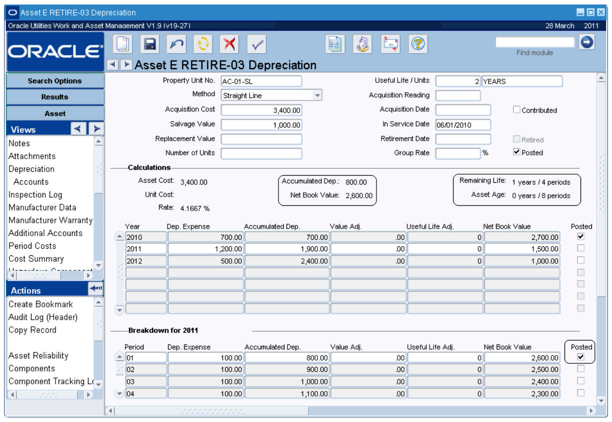
Asset Cost: 3,400.00 Accumulated Dep.: 800.00 Remaining Life: 1 years / 4 periods
Unit Cost: Net Book Value: 2,600.00 Asset Age: 0 years / 8 periods
Rate: 4.1667 %

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	700.00	700.00	.00	0	2,700.00	✓
2011	1,200.00	1,900.00	.00	0	1,500.00	☐
2012	500.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
06	100.00	100.00	.00	0	3,300.00	✓
07	100.00	200.00	.00	0	3,200.00	✓
08	100.00	300.00	.00	0	3,100.00	✓
09	100.00	400.00	.00	0	3,000.00	✓

3. Note that the Year 2011, Period 01 is also posted. The information right below the Calculations heading are updated as well.



Asset E RETIRE-03 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	06/01/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ☑ Posted

Calculations

Asset Cost: 3,400.00 Accumulated Dep.: 800.00 Remaining Life: 1 years / 4 periods
Unit Cost: Net Book Value: 2,600.00 Asset Age: 0 years / 8 periods
Rate: 4.1667 %

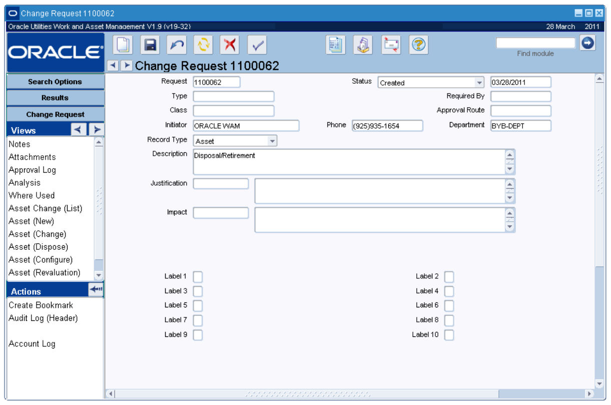
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	700.00	700.00	.00	0	2,700.00	✓
2011	1,200.00	1,900.00	.00	0	1,500.00	☐
2012	500.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	800.00	.00	0	2,600.00	✓
02	100.00	900.00	.00	0	2,500.00	☐
03	100.00	1,000.00	.00	0	2,400.00	☐
04	100.00	1,100.00	.00	0	2,300.00	☐

Create a Change Request

1. Create a Change Request with Asset as Record Type.



Change Request 1100062

Request	1100062	Status	Created		03/28/2011
Type				Required By	
Class				Approval Route	
Initiator	ORACLE WAM	Phone	(925)935-1654	Department	BYB-DEPT
Record Type	Asset				
Description	Disposal/Retirement				
Justification					
Impact					

☐ Label 1 ☐ Label 2 ☐ Label 3 ☐ Label 4 ☐ Label 5 ☐ Label 6 ☐ Label 7 ☐ Label 8 ☐ Label 9 ☐ Label 10

2. Navigate to the Asset (Dispose) view and enter the asset that was created in the first step in this procedure above.
   Create a record based on the screen below. Make sure that the Post to Prior Year box is checked. Save the record.

2.1 The Calculated Net Book Value was set to $2,761.29 after the Effective Date was entered.

2.1.1 The Net Book Value for the Year and Period of the Retirement Date (i.e. Year 2011, Period 12) from the forecast is $2,700. The Depreciation Expense for the same Year and Period is $100.

2.1.2 Compute Partial Percentage.

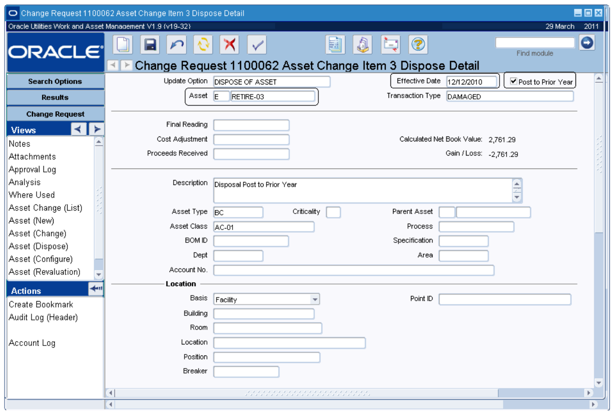
2.1.2.1 Retirement Date – Start Date + 1 = Dec. 12, 2011 – Dec. 1, 2011 + 1 = 12

2.1.2.2 End Date – Start Date + 1 = Dec. 31, 2011 – Dec. 1, 2011 + 1 = 31

2.1.2.3 Partial Percentage = 12/31 * 100 = 38.71%

2.1.3 Calculated Net Book Value = Net Book Value + Depreciation Expense – (Depreciation Expense * Partial Percentage /100) = $2,700 + $100 – ($100 * 38.71/100) = $2,761.29.

2.2 The Gain/Loss is the same as the Calculated Net Book Value.



Change Request 1100062 Asset Change Item 3 Dispose Detail

Oracle Utilities Work and Asset Management V1.9 (v19-32) 29 March 2011

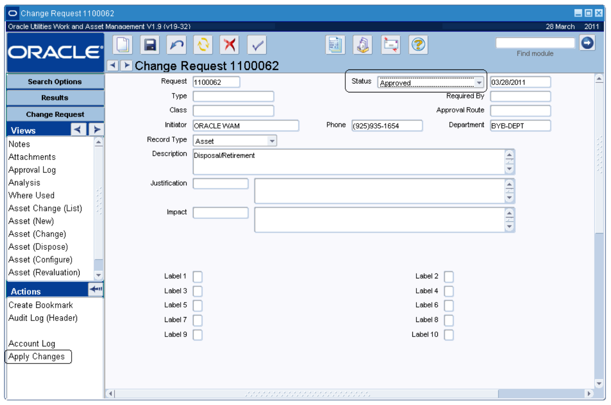
Change Request 1100062 Asset Change Item 3 Dispose Detail

Update Option	DISPOSE OF ASSET	Effective Date	12/12/2010
		✓ Post to Prior Year	
Asset	E RETIRE-03	Transaction Type	DAMAGED
Final Reading			
Cost Adjustment		Calculated Net Book Value:	2,761.29
Proceeds Received		Gain / Loss:	-2,761.29
Description	Disposal Post to Prior Year		
Asset Type	BC	Criticality	
Asset Class	AC-01	Parent Asset	
BOM ID		Process	
Dept		Specification	
Account No.		Area	

Location

Basis	Facility	Point ID	
Building			
Room			
Location			
Position			
Breaker			

3. Navigate back to the Change Request header and change the status from Created to Approved. The Apply Changes action appears.



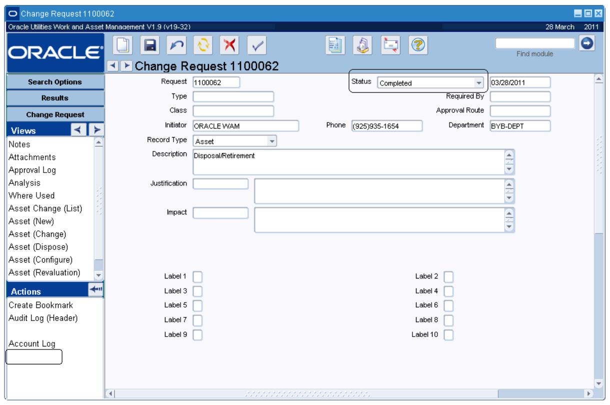
Change Request 1100062

Oracle Utilities Work and Asset Management V1.9 (v19-32) 28 March 2011

Change Request 1100062

Request	1100062	Status	Approved		03/28/2011
Type				Required By	
Class				Approval Route	
Initiator	ORACLE WAM	Phone	(925)935-1654	Department	BYB-DEPT
Record Type	Asset				
Description	Disposal/Retirement				
Justification					
Impact					
Label 1	☐			Label 2	☐
Label 3	☐			Label 4	☐
Label 5	☐			Label 6	☐
Label 7	☐			Label 8	☐
Label 9	☐			Label 10	☐

4. Click on the Apply Changes action. The status changes from Approved to Completed and Apply Changes action disappears.



Request: 1100062
Status: Completed 03/28/2011
Type:
Required By:
Class:
Approval Route:
Initiator: ORACLE WAM
Phone: (925)935-1654
Department: BYB-DEPT
Record Type: Asset
Description: Disposal/Retirement
Justification:
Impact:

5. Click the Account Log action. The Account Log module is displayed.

5.1 Asset Gain/Loss (AG)

5.1.1 The Transaction Date is the End Date of the last Period of the Prior Year in Accounting Period table.

5.1.2 Year and Mon are the Year and Period of the Transaction Date.

5.1.3 Effective Date is the Retirement Date.

5.1.4 Transaction Amount is the Calculated Net Book Value in 10.1 above.

5.2 Asset Cost Write Off (AW)

5.2.1 The Transaction Date is the End Date of the last Period of the Prior Year in Accounting Period table.

5.2.2 Year and Mon are the Year and Period of the Transaction Date.

5.2.3 Effective Date is the Retirement Date.

5.2.4 Add the Acquisition Cost to the sum of all value adjustments in the forecast. Multiply the sum by –1. The product is the Transaction Amount.

5.3 Asset Accumulated Depreciation (DA)

5.3.1 The Transaction Date is the End Date of the last Period of the Prior Year in Accounting Period table.

5.3.2 Year and Mon are the Year and Period of the Transaction Date.

5.3.3 Effective Date is the Retirement Date.

5.3.4 Transaction Amount is $638.71.

5.3.4.1 The Accumulated Dep. and Dep. Expense of the asset for the Year 2010 Period 12 before the Change Request was applied were $700 and $100, respectively.

5.3.4.2 The Partial Percentage is 38.71%. Refer to 10.1.2.3 above.

5.3.4.3 The DA Transaction Amount is Accumulated Dep. less Dep. Expense plus product of Dep. Expense and Partial Percentage. So that is $700 - $100 + ($100 * 38.71 /100) = $638.71.

5.4 Asset Depreciation/ Asset Accumulated Depreciation (DP/DA) in **Yellow**

5.4.1 The reversed transaction for Year 2010 Period 12.

5.4.2 Transaction Date is the End Date of the last Period of the Prior Year in Accounting Period table.

5.4.3 Year and Mon are the Year and Period of the Transaction Date.

5.4.4 Effective Date is the End Date of the Year and Period being reversed.

5.4.5 Note that the DA is positive and the DP is negative.

5.5 Asset Depreciation/ Asset Accumulated Depreciation (DP/DA) in **Blue**

5.5.1 The reversed transaction for Year 2011 Period 01.

5.5.2 Transaction Date is the Change Request Status Date.

5.5.3 Year and Mon are the Year and Period of the Transaction Date.

5.5.4 Effective Date is the End Date of the Year and Period being reversed.

5.5.5 Note that the DA is positive and the DP is negative.

5.5.6 Note that the Transaction Date is different from the date in 13.4.2.

5.6 Asset Depreciation/ Asset Accumulated Depreciation (DP/DA) in **Green**

5.6.1 The actual retirement transaction for the Year and Period of Retirement Date.

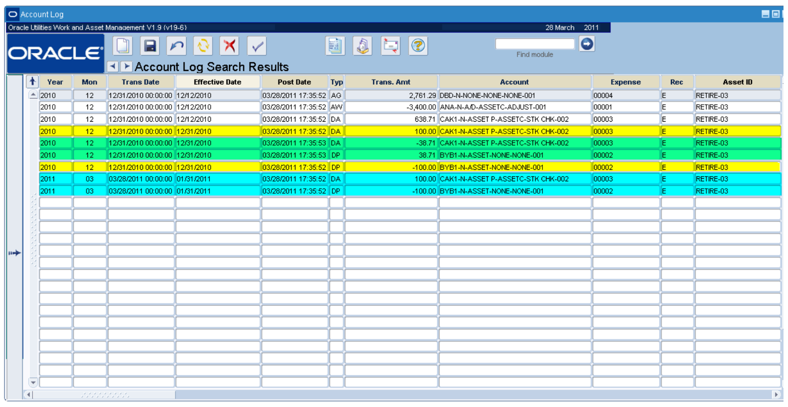
5.6.2 Transaction Date is the End Date of the last Period of the Prior Year in Accounting Period table.

5.6.3 Year and Mon are the Year and Period of the Transaction Date.

5.6.4 Effective Date is the End Date of the Year and Period posted.

5.6.5 Note that the DP is positive and the DA is negative.

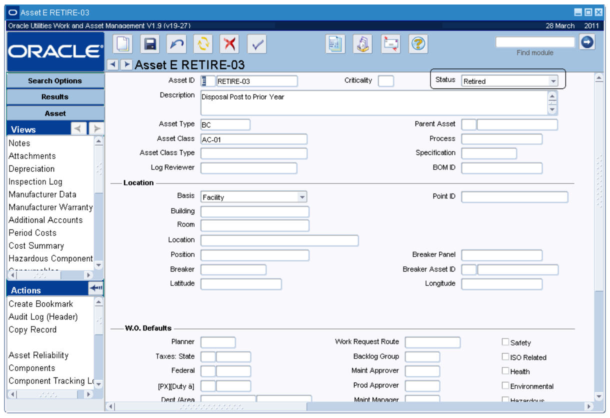
5.7 Post Date is Sysdate.


Account Log
Oracle Utilities Work and Asset Management V1.9 (v19-6)
28 March 2011
ORACLE
Find module
Account Log Search Results

Year	Mon	Trans Date	Effective Date	Post Date	Typ	Trans. Amt	Account	Expense	Rec	Asset ID
2010	12	12/31/2010 00:00:00	12/12/2010	03/28/2011 17:35:52	AG	2,761.29	DBD-N-NONE-NONE-NONE-001	00004	E	RETIRE-03
2010	12	12/31/2010 00:00:00	12/12/2010	03/28/2011 17:35:52	AW	-3,400.00	ANA-N-A/D-ASSETC-ADJUST-001	00001	E	RETIRE-03
2010	12	12/31/2010 00:00:00	12/12/2010	03/28/2011 17:35:52	DA	638.71	CAK1-N-ASSET P-ASSETC-STK CHK-002	00003	E	RETIRE-03
2010	12	12/31/2010 00:00:00	12/31/2010	03/28/2011 17:35:52	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	00003	E	RETIRE-03
2010	12	12/31/2010 00:00:00	12/31/2010	03/28/2011 17:35:53	DA	-38.71	CAK1-N-ASSET P-ASSETC-STK CHK-002	00003	E	RETIRE-03
2010	12	12/31/2010 00:00:00	12/31/2010	03/28/2011 17:35:53	DP	38.71	BYB1-N-ASSET-NONE-NONE-001	00002	E	RETIRE-03
2010	12	12/31/2010 00:00:00	12/31/2010	03/28/2011 17:35:52	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	00002	E	RETIRE-03
2011	03	03/28/2011 00:00:00	01/31/2011	03/28/2011 17:35:52	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	00003	E	RETIRE-03
2011	03	03/28/2011 00:00:00	01/31/2011	03/28/2011 17:35:52	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	00002	E	RETIRE-03

Verify Revised Cost Calculations

1. Navigate back to the Asset module and retrieve the asset that was created in the first step in this procedure above . The asset is now Retired.



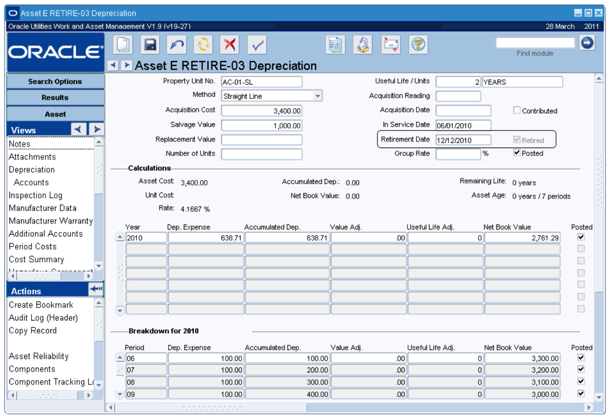
Asset E RETIRE-03

Asset ID: E RETIRE-03
Criticality:
Status: Retired
Description: Disposal Post to Prior Year
Asset Type: BC
Asset Class: AC-01
Asset Class Type:
Log Reviewer:
Parent Asset:
Process:
Specification:
BOM ID:

Location
Basis: Facility
Building:
Room:
Location:
Position:
Breaker:
Latitude:
Point ID:
Breaker Panel:
Breaker Asset ID:
Longitude:

W.O. Defaults
Planner:
Taxes: State
Federal
[PX][Duty ä]
Dept /Area
Work Request Route:
Backlog Group:
Maint Approver:
Prod Approver:
Maint Manager:
☐ Safety
☐ ISO Related
☐ Health
☐ Environmental
☐ Hazardous

2. Navigate to the Depreciation view. The retirement Date is now populated with the Effective Date from the Change Request. The Retired box is checked.



Asset E RETIRE-03 Depreciation

Property Unit No.: AC-01-SL
Method: Straight Line
Acquisition Cost: 3,400.00
Salvage Value: 1,000.00
Replacement Value:
Number of Units:
Useful Life / Units: 2 YEARS
Acquisition Reading:
Acquisition Date:
In Service Date: 06/01/2010
Retirement Date: 12/12/2010
Group Rate: %
☐ Contributed
☑ Retired
☑ Posted

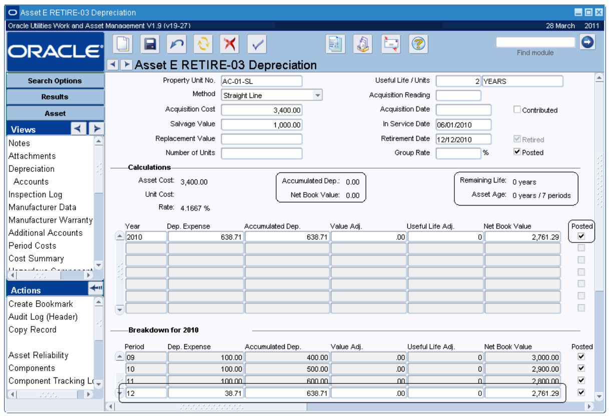
Calculations
Asset Cost: 3,400.00
Unit Cost:
Rate: 4.1667 %
Accumulated Dep.: 0.00
Net Book Value: 0.00
Remaining Life: 0 years
Asset Age: 0 years / 7 periods

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	638.71	638.71	.00	0	2,761.29	✓

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
06	100.00	100.00	.00	0	3,300.00	✓
07	100.00	200.00	.00	0	3,200.00	✓
08	100.00	300.00	.00	0	3,100.00	✓
09	100.00	400.00	.00	0	3,000.00	✓

3. Scroll down to the last Breakdown record.
The Remaining Life, Accumulated Dep. and Net Book Value are now zero. Asset Age is seven periods for Period 06 to 12. All the forecast records for Year 2010 are posted. All the forecast records after Year 2010 Period 12 are deleted. The values in the last forecast record are only partial of the original amount. The asset is now fully depreciated and retired.



Asset E RETIRE-03 Depreciation

Property Unit No.: AC-01-SL
Method: Straight Line
Acquisition Cost: 3,400.00
Salvage Value: 1,000.00
Replacement Value:
Number of Units:
Useful Life / Units: 2 YEARS
Acquisition Reading:
Acquisition Date:
In Service Date: 06/01/2010
Retirement Date: 12/12/2010
Group Rate: %
☐ Contributed
☑ Retired
☑ Posted

Calculations
Asset Cost: 3,400.00
Unit Cost:
Rate: 4.1667 %
Accumulated Dep.: 0.00
Net Book Value: 0.00
Remaining Life: 0 years
Asset Age: 0 years / 7 periods

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	638.71	638.71	.00	0	2,761.29	✓

Breakdown for 2010

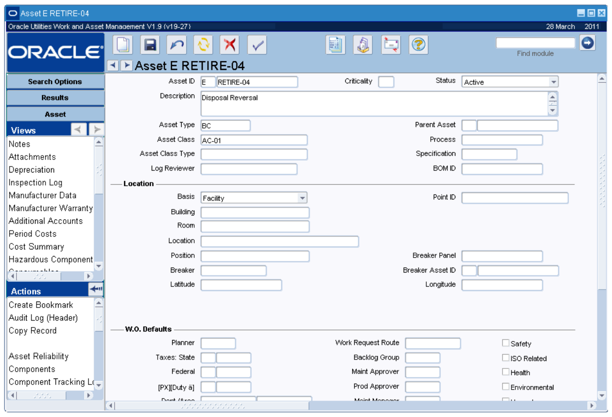
Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	400.00	.00	0	3,000.00	✓
10	100.00	500.00	.00	0	2,900.00	✓
11	100.00	600.00	.00	0	2,800.00	✓
12	38.71	638.71	.00	0	2,761.29	✓

Disposal Reversal

Follow these steps to dispose of an asset and reverse costs associated with the asset.

Create the Asset and Set Depreciation Settings

1. Create an asset. Make sure that the Asset Class and its Depreciation Parameters are set-up.



Asset E RETIRE-04

Asset ID	E RETIRE-04	Criticality	
Description	Disposal Reversal	Status	Active
Asset Type	BC	Parent Asset	
Asset Class	AC-01	Process	
Asset Class Type		Specification	
Log Reviewer		BOM ID	

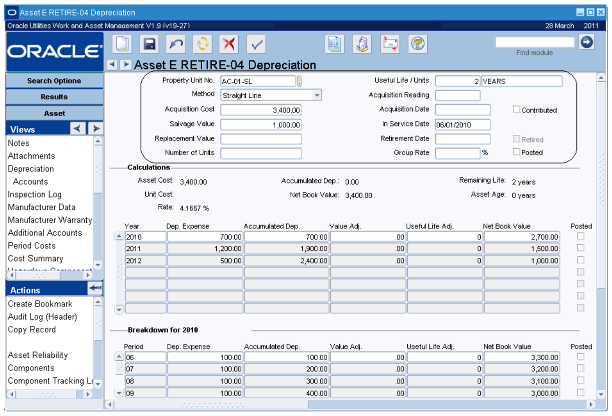
Location

Basis	Facility	Point ID	
Building			
Room			
Location			
Position		Breaker Panel	
Breaker		Breaker Asset ID	
Latitude		Longitude	

W.O. Defaults

Planner		Work Request Route		☐ Safety
Taxes: State		Backlog Group		☐ ISO Related
Federal		Maint Approver		☐ Health
[PX][Duty ä]		Prod Approver		☐ Environmental

2. Navigate to Depreciation view to create a Fixed Asset record. Populate Fixed Asset based on the screen below. Upon Save, the Calculations and Breakdown sections are automatically populated.



Asset E RETIRE-04 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS	
Method	Straight Line	Acquisition Reading		
Acquisition Cost	3,400.00	Acquisition Date		☐ Contributed
Salvage Value	1,000.00	In Service Date	06/01/2010	
Replacement Value		Retirement Date		☐ Retired
Number of Units		Group Rate	%	☐ Posted

Calculations

Asset Cost:	3,400.00	Accumulated Dep.:	0.00	Remaining Life:	2 years
Unit Cost:		Net Book Value:	3,400.00	Asset Age:	0 years
Rate:	4.1667 %				

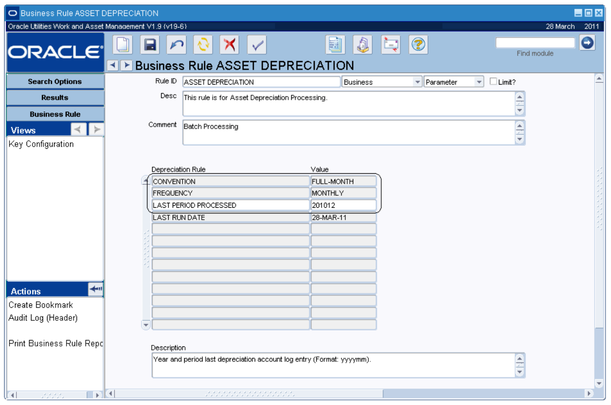
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	700.00	700.00	.00	0	2,700.00	☐
2011	1,200.00	1,900.00	.00	0	1,500.00	☐
2012	500.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
06	100.00	100.00	.00	0	3,300.00	☐
07	100.00	200.00	.00	0	3,200.00	☐
08	100.00	300.00	.00	0	3,100.00	☐
09	100.00	400.00	.00	0	3,000.00	☐

Modify the Asset Depreciation Business Rule and Run Batch

1. Change the Business Rule ASSET DEPRECIATION. Set LAST PERIOD PROCESSED, CONVENTION and FREQUENCY based on the screen below.



Business Rule ASSET DEPRECIATION

Rule ID: ASSET DEPRECIATION | Business | Parameter | ☐ Limit?

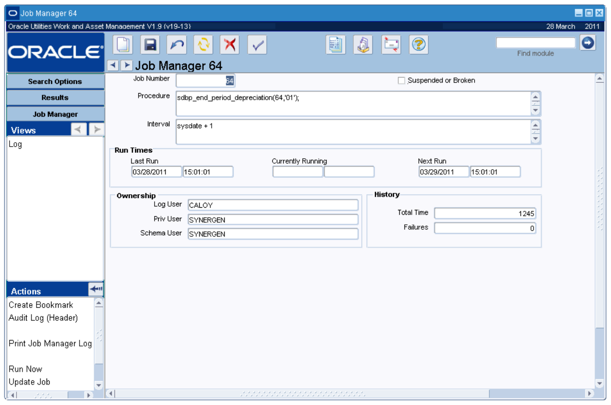
Desc: This rule is for Asset Depreciation Processing.

Comment: Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201012
LAST RUN DATE	28-MAR-11

Description: Year and period last depreciation account log entry (Format: yyyymm).

2. Run the batch job sdbp_end_period_depreciation. The batch job picks-up the LAST PERIOD PROCESSED value and processes the next period, which is Year 2011 Period.



Job Manager 64

Job Number	64
Suspended or Broken	☐
Procedure	sdbp_end_period_depreciation(64,'01');
Interval	sysdate + 1

Run Times

Last Run		Currently Running		Next Run	
03/28/2011	15:01:01			03/29/2011	15:01:01

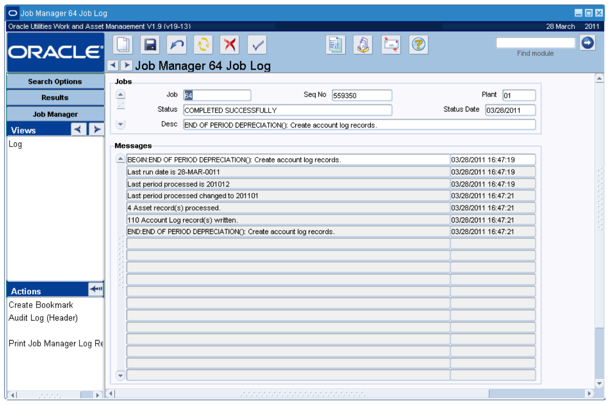
Ownership

Log User	CALOY
Priv User	SYNERGEN
Schema User	SYNERGEN

History

Total Time	1245
Failures	0

3. Make sure that the batch job ran successfully.



Job Manager 64 Job Log

Jobs

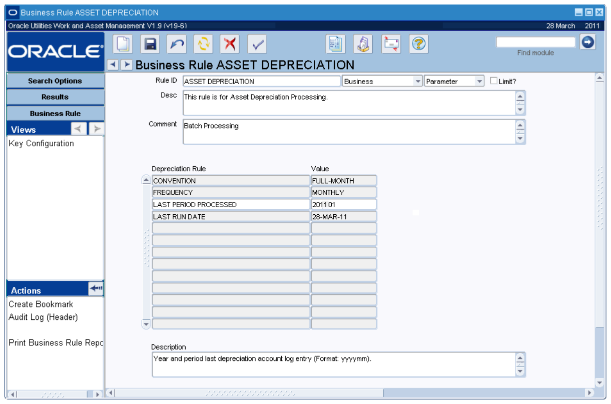
Job	64
Seq No	559350
Plant	01
Status	COMPLETED SUCCESSFULLY
Status Date	03/28/2011
Desc	END OF PERIOD DEPRECIATION(): Create account log records.

Messages

```
BEGIN:END OF PERIOD DEPRECIATION(): Create account log records.   03/28/2011 16:47:19
Last run date is 28-MAR-0011                                      03/28/2011 16:47:19
Last period processed is 201012                                   03/28/2011 16:47:19
Last period processed changed to 201101                           03/28/2011 16:47:21
4 Asset record(s) processed.                                      03/28/2011 16:47:21
110 Account Log record(s) written.                                03/28/2011 16:47:21
END:END OF PERIOD DEPRECIATION(): Create account log records.     03/28/2011 16:47:21
```

Verify that the Asset was Processed Correctly

1. Navigate back to the ASSET DEPRECIATION Business Rule. Make sure that the LAST PERIOD PROCESSED is incremented by one period and the LAST RUN DATE is updated with Sysdate.



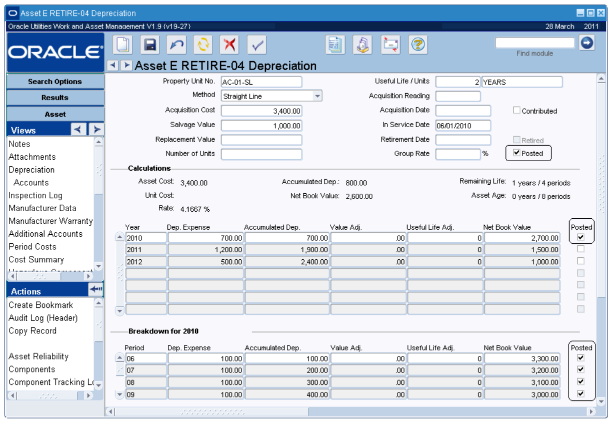
Business Rule ASSET DEPRECIATION

Rule ID	ASSET DEPRECIATION
	Business
	Parameter
Limit?	☐
Desc	This rule is for Asset Depreciation Processing.
Comment	Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201101
LAST RUN DATE	28-MAR-11

Description: Year and period last depreciation account log entry (Format: yyyymm).

2. Navigate back to the Depreciation view of the Asset module. Note the Posted check boxes. If it is **checked**, it means that the asset has been processed for the end of period and Account Log entries have been written against the asset.



Asset E RETIRE-04 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	06/01/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ☑ Posted

Calculations

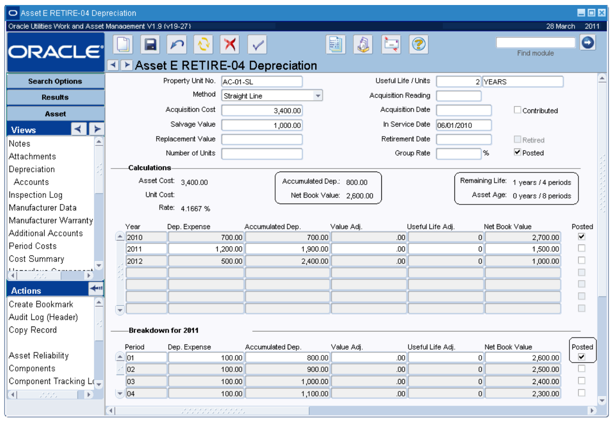
Asset Cost: 3,400.00 Accumulated Dep.: 800.00 Remaining Life: 1 years / 4 periods
Unit Cost: Net Book Value: 2,600.00 Asset Age: 0 years / 8 periods
Rate: 4.1667 %

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	700.00	700.00	.00	0	2,700.00	✓
2011	1,200.00	1,900.00	.00	0	1,500.00	☐
2012	500.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
06	100.00	100.00	.00	0	3,300.00	✓
07	100.00	200.00	.00	0	3,200.00	✓
08	100.00	300.00	.00	0	3,100.00	✓
09	100.00	400.00	.00	0	3,000.00	✓

3. Note that the Year 2011, Period 01 is also posted. The information right below the Calculations heading is updated as well.



Asset E RETIRE-04 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	06/01/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ☑ Posted

Calculations

Asset Cost: 3,400.00 Accumulated Dep.: 800.00 Remaining Life: 1 years / 4 periods
Unit Cost: Net Book Value: 2,600.00 Asset Age: 0 years / 8 periods
Rate: 4.1667 %

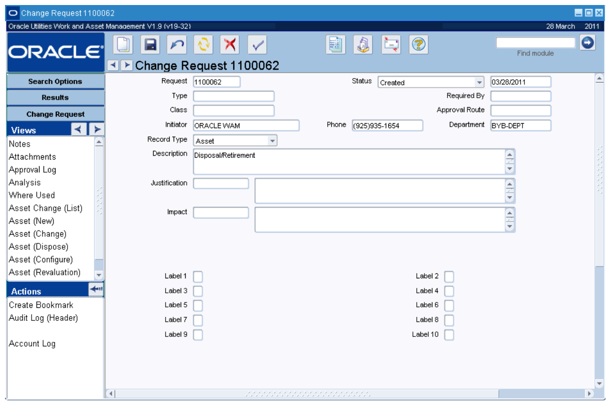
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	700.00	700.00	.00	0	2,700.00	✓
2011	1,200.00	1,900.00	.00	0	1,500.00	☐
2012	500.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	800.00	.00	0	2,600.00	✓
02	100.00	900.00	.00	0	2,500.00	☐
03	100.00	1,000.00	.00	0	2,400.00	☐
04	100.00	1,100.00	.00	0	2,300.00	☐

Create a Change Request

1. Create a Change Request with Asset as Record Type.



Change Request 1100062

Request	1100062	Status	Created 03/28/2011
Type		Required By	
Class		Approval Route	
Initiator	ORACLE WAM	Phone	(925)935-1654
Department	BYB-DEPT		
Record Type	Asset		
Description	Disposal/Retirement		
Justification			
Impact			

2. Navigate to Asset (Change) view and enter the asset that was created in the first step in this procedure above.
Create a record based on the screen below. Save the record.

2.1 The Calculated Net Book Value was set to \$2,761.29 after the Effective Date was entered.

2.1.1 The Net Book Value for the Year and Period of the Retirement Date (i.e. Year 2011, Period 12) from the forecast is \$2,700. The Depreciation Expense for the same Year and Period is \$100.

2.1.2 Compute Partial Percentage.

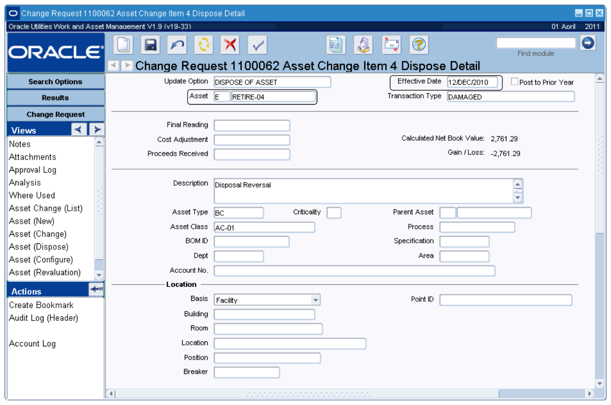
2.1.2.1 Retirement Date – Start Date + 1 = Dec. 12, 2011 – Dec. 1, 2011 + 1 = 12

2.1.2.2 End Date – Start Date + 1 = Dec. 31, 2011 – Dec. 1, 2011 + 1 = 31

2.1.2.3 Partial Percentage = 12/31 * 100 = 38.71%

2.1.3 Calculated Net Book Value = Net Book Value + Depreciation Expense – (Depreciation Expense * Partial Percentage /100) = \$2,700 + \$100 – (\$100 * 38.71/100) = \$2,761.29.

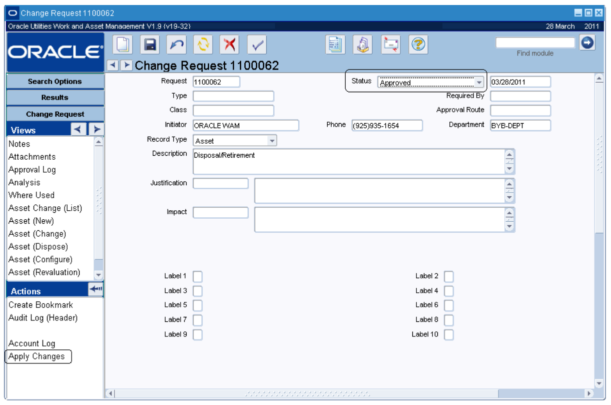
2.2 The Gain/Loss is the same as the Calculated Net Book Value.



Change Request 1100062 Asset Change Item 4 Dispose Detail
Oracle Utilities Work and Asset Management V1.9 (v19-33) 01 April 2011

Update Option	DISPOSE OF ASSET	Effective Date	12/DEC/2010
Asset	E RETIRE-04	Transaction Type	DAMAGED
Final Reading			
Cost Adjustment		Calculated Net Book Value:	2,761.29
Proceeds Received		Gain / Loss:	-2,761.29
Description	Disposal Reversal		
Asset Type	BC	Parent Asset	
Asset Class	AC-01	Process	
BOM ID		Specification	
Dept		Area	
Account No.			
Basis	Facility	Point ID	

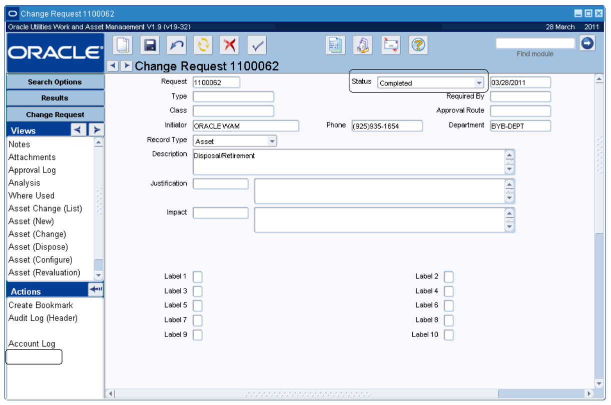
3. Navigate back to the Change Request header and change the status from Created to Approved. The Apply Changes action appears.



Change Request 1100062
Oracle Utilities Work and Asset Management V1.9 (v19-32) 28 March 2011

Request	1100062	Status	Approved 03/28/2011
Type		Required By	
Class		Approval Route	
Initiator	ORACLE WAM	Phone	(925)935-1654
Department	BYB-DEPT		
Record Type	Asset		
Description	Disposal/Retirement		

4. Click on the Apply Changes action. The status changed from Approved to Completed and Apply Changes action disappears.



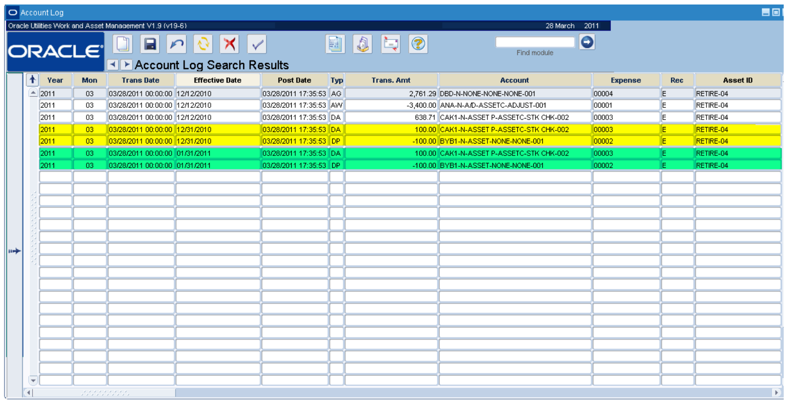
Change Request 1100062

Request	1100062	Status	Completed, 03/28/2011
Type		Required By	
Class		Approval Route	
Initiator	ORACLE WAM	Phone	(925)935-1654
Department	BYB-DEPT	Record Type	Asset
Description	Disposal/Retirement	Justification	
Impact			

5. Click the Account Log action. The Account Log module is displayed.
   - 5.1 Transaction Date is the Change Request Status Date.
   - 5.2 Year and Mon are the Year and Period of the Transaction Date from the Accounting Period table.
   - 5.3 Post Date is Sysdate.
   - 5.4 Asset Gain/Loss (AG)
     - 5.4.1 Effective Date is the Retirement Date.
     - 5.4.2 Trans. Amt is the Calculated Net Book Value from 10.1.
   - 5.5 Asset Cost Write Off (AW)
     - 5.5.1 Effective Date is the Retirement Date.
     - 5.5.2 Add the Acquisition Cost to the sum of all value adjustments in the forecast. Multiply the sum by –1. The product is the Tran. Amt.
   - 5.6 Asset Accumulated Depreciation (DA)
     - 5.6.1 Effective Date is the Retirement Date.
     - 5.6.2 Trans. Amt is $638.71
       - 5.6.2.1 The Accumulated Dep. and Dep. Expense of the asset for the Year 2010 Period 12 before the Change Request was applied were $700 and $100, respectively.
       - 5.6.2.2 The Partial Percentage is 38.71%. Refer to 10.1.2.3 above.
       - 5.6.2.3 The DA Transaction Amount is Accumulated Dep. less Dep. Expense plus product of Dep. Expense and Partial Percentage. So that is $700 - $100 + ($100 * 38.71 /100) = $638.71.
   - 5.7 Asset Depreciation/ Asset Accumulated Depreciation (DP/DA) in Yellow
     - 5.7.1 The reversed transaction for Year 2010 Period 12.
     - 5.7.2 Effective Date is the End Date of Year 2010 Period 12 from Accounting Period table.
     - 5.7.3 Trans. Amt is the reversed Depreciation Expense.
   - 5.8 Asset Depreciation/ Asset Accumulated Depreciation (DP/DA) in Green
     - 5.8.1 The reversed transaction for Year 2011 Period 01.

5.8.2 Effective Date is the End Date of Year 2011 Period 01 from Accounting Period table.

5.8.3 Trans. Amt is the reversed Depreciation Expense.



Account Log

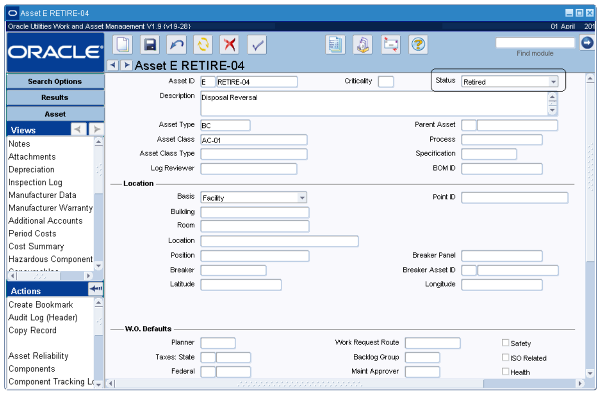
Oracle Utilities Work and Asset Management V1.9 (v19-6) 28 March 2011

Account Log Search Results

Year	Mon	Trans Date	Effective Date	Post Date	Typ	Trans. Amt	Account	Expense	Rec	Asset ID
2011	03	03/28/2011 00:00:00	12/12/2010	03/28/2011 17:35:53	AG	2,761.29	DBD-N-NONE-NONE-NONE-001	00004	E	RETIRE-04
2011	03	03/28/2011 00:00:00	12/12/2010	03/28/2011 17:35:53	AW	-3,400.00	ANA-N-A/D-ASSETC-ADJUST-001	00001	E	RETIRE-04
2011	03	03/28/2011 00:00:00	12/12/2010	03/28/2011 17:35:53	DA	638.71	CAK1-N-ASSET P-ASSETC-STK CHK-002	00003	E	RETIRE-04
2011	03	03/28/2011 00:00:00	12/31/2010	03/28/2011 17:35:53	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	00003	E	RETIRE-04
2011	03	03/28/2011 00:00:00	12/31/2010	03/28/2011 17:35:53	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	00002	E	RETIRE-04
2011	03	03/28/2011 00:00:00	01/31/2011	03/28/2011 17:35:53	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	00003	E	RETIRE-04
2011	03	03/28/2011 00:00:00	01/31/2011	03/28/2011 17:35:53	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	00002	E	RETIRE-04

Verify Revised Cost Calculations

1. Navigate back to the Asset (Change) screen. The Asset Cost field should now be the same for both Current Value and Requested Change columns.



Asset E RETIRE-04

Oracle Utilities Work and Asset Management V1.9 (v19-28) 01 April 201

Asset E RETIRE-04

Asset ID	E RETIRE-04	Criticality	
Status	Retired		
Description	Disposal Reversal		
Asset Type	BC	Parent Asset	
Asset Class	AC-01	Process	
Asset Class Type		Specification	
Log Reviewer		BOM ID	

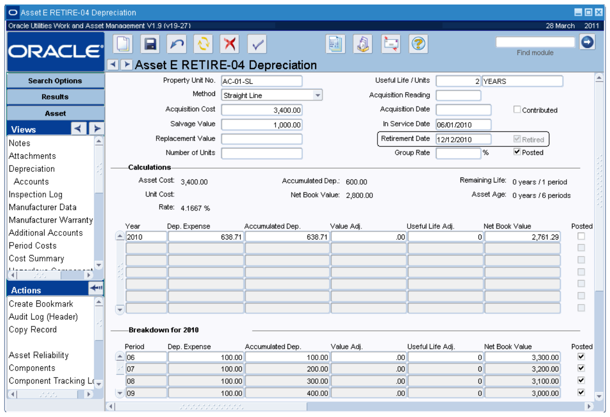
Location

Basis	Facility	Point ID	
Building			
Room			
Location			
Position		Breaker Panel	
Breaker		Breaker Asset ID	
Latitude		Longitude	

W.O. Defaults

Planner		Work Request Route		☐ Safety
Taxes: State		Backlog Group		☐ ISO Related
Federal		Maint Approver		☐ Health

2. Navigate to the Depreciation view. The retirement Date is now populated with the Effective Date from the Change Request. The Retired box is checked.



Asset E RETIRE-04 Depreciation

Oracle Utilities Work and Asset Management V1.9 (v19-27) 28 March 2011

Asset E RETIRE-04 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS	
Method	Straight Line	Acquisition Reading		
Acquisition Cost	3,400.00	Acquisition Date		☐ Contributed
Salvage Value	1,000.00	In Service Date	06/01/2010	
Replacement Value		Retirement Date	12/12/2010	✓ Retired
Number of Units		Group Rate	%	✓ Posted

Calculations

Asset Cost:	3,400.00	Accumulated Dep.:	600.00	Remaining Life:	0 years / 1 period
Unit Cost:		Net Book Value:	2,800.00	Asset Age:	0 years / 6 periods
Rate:	4.1667 %				

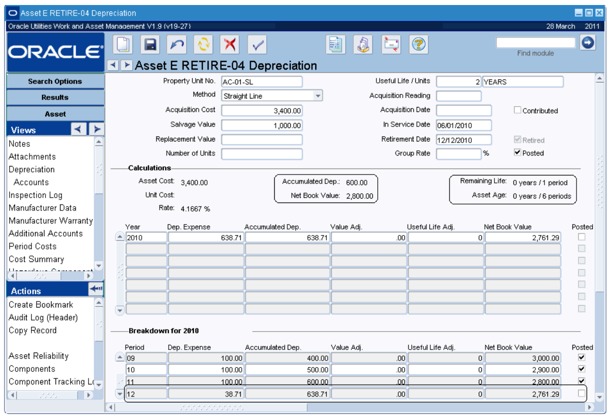
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	638.71	638.71	.00	0	2,761.29	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
06	100.00	100.00	.00	0	3,300.00	✓
07	100.00	200.00	.00	0	3,200.00	✓
08	100.00	300.00	.00	0	3,100.00	✓
09	100.00	400.00	.00	0	3,000.00	✓

3. Scroll down to the last Breakdown record.Only one unposted record remains. The forecast records after the Retirement Date were deleted. The Accumulated Dep and Net Book Value reflect the values of the last posted record. The Remaining Life and Asset

Age were also updated. Only one unposted record remains. The values in the last forecast record are only partial of the original amount.



Asset E RETIRE-04 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	06/01/2010
Replacement Value		Retirement Date	12/12/2010
Number of Units		Group Rate	%

☐ Contributed
☑ Retired
☑ Posted

Calculations

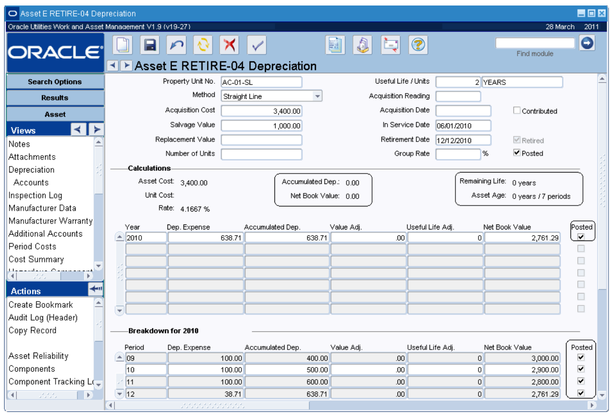
Asset Cost: 3,400.00
Unit Cost:
Rate: 4.1667 %
Accumulated Dep.: 600.00
Net Book Value: 2,800.00
Remaining Life: 0 years / 1 period
Asset Age: 0 years / 6 periods

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	638.71	638.71	.00	0	2,761.29	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	400.00	.00	0	3,000.00	✓
10	100.00	500.00	.00	0	2,900.00	✓
11	100.00	600.00	.00	0	2,800.00	✓
12	38.71	638.71	.00	0	2,761.29	☐

4. Run sdbp_end_period_depreciation. Navigate back to the Asset's Depreciation view. Year 2010 Period 12 is now posted. The Accumulated Dep and Net Book Value are 0. Remaining Life is zero and the Asset Age equals the number of posted periods. The asset is now fully depreciated and retired.



Asset E RETIRE-04 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	06/01/2010
Replacement Value		Retirement Date	12/12/2010
Number of Units		Group Rate	%

☐ Contributed
☑ Retired
☑ Posted

Calculations

Asset Cost: 3,400.00
Unit Cost:
Rate: 4.1667 %
Accumulated Dep.: 0.00
Net Book Value: 0.00
Remaining Life: 0 years
Asset Age: 0 years / 7 periods

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	638.71	638.71	.00	0	2,761.29	✓

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	400.00	.00	0	3,000.00	✓
10	100.00	500.00	.00	0	2,900.00	✓
11	100.00	600.00	.00	0	2,800.00	✓
12	38.71	638.71	.00	0	2,761.29	✓

5. Rerun the batch job and make sure that it ran successfully.
   Make sure that the LAST PERIOD PROCESSED is incremented by one period (i.e. 201102)and the LAST RUN DATE is updated with Sysdate. Navigate back to the Depreciation view of the Asset module. Note that all period for the year 2010 are now posted and the Posted indicator for year 2010 is also checked.

6. Highlight the Year 2011.
   The Rate, Accumulated Dep., Net Book Value, Remaining Life and Asset Age were updated. The system 'caught-up' by posting the three reversed forecast records and the next period as well. Periods 01 and 02 for the Year 2011 were posted.

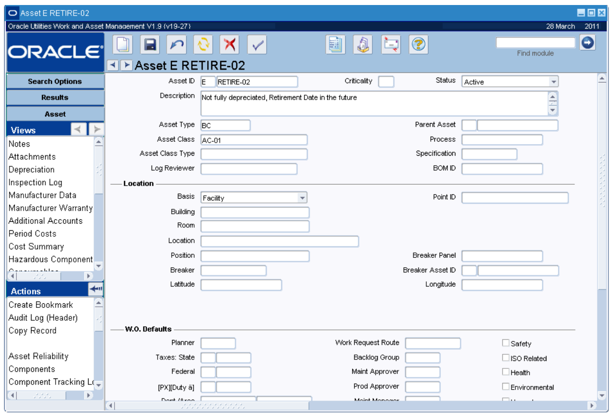
7. Navigate back to the Account Log and retrieve the records related to the asset.
   The records in yellow are the entries created by the Change Request. The entries in blue are the records created by the batch job the second time it was ran. The log entries in green are the records created the first time the batch job was executed. The AN and log entries for the Year 2010 Periods 09 and 10 were purposely not included.

Disposal

Follow these steps to simply dispose of an asset.

Create the Asset and Set Depreciation Settings

1. Create an asset. Make sure that the Asset Class and its Depreciation Parameters are set-up.



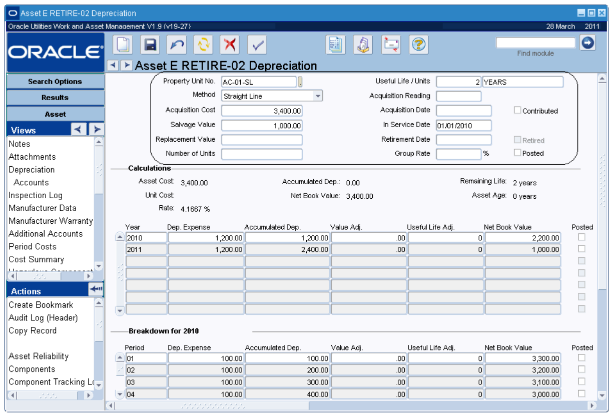
Asset E RETIRE-02

Oracle Utilities Work and Asset Management V1.9 (v19-27) 28 March 2011

Asset E RETIRE-02

Asset ID	E RETIRE-02	Status	Active
Description	Not fully depreciated, Retirement Date in the future		
Asset Type	BC	Parent Asset	
Asset Class	AC-01	Process	
Location Basis	Facility	Point ID	

2. Navigate to Depreciation view to create a Fixed Asset record. Populate Fixed Asset based on the screen below. Upon Save, the Calculations and Breakdown sections are automatically populated.



Asset E RETIRE-02 Depreciation

Oracle Utilities Work and Asset Management V1.9 (v19-27) 28 March 2011

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/01/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

Calculations

Asset Cost:	3,400.00	Accumulated Dep.:	0.00	Remaining Life:	2 years
Unit Cost:		Net Book Value:	3,400.00	Asset Age:	0 years
Rate:	4.1667 %				

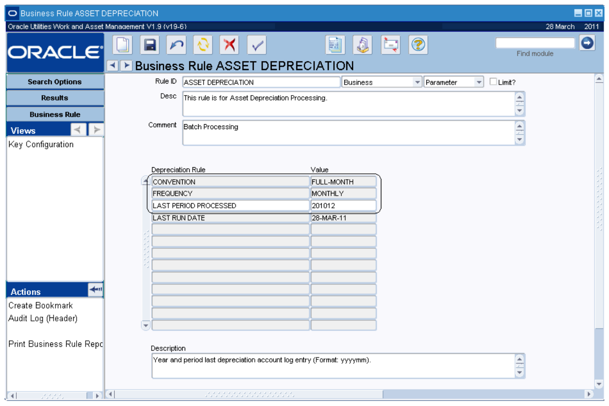
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	1,200.00	1,200.00	.00	0	2,200.00	☐
2011	1,200.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	100.00	.00	0	3,300.00	☐
02	100.00	200.00	.00	0	3,200.00	☐
03	100.00	300.00	.00	0	3,100.00	☐
04	100.00	400.00	.00	0	3,000.00	☐

Modify the Asset Depreciation Business Rule and Run Batch

1. Change the ASSET DEPRECIATION business rule. Set LAST PERIOD PROCESSED, CONVENTION and FREQUENCY based on the screen below.



Business Rule ASSET DEPRECIATION

Oracle Utilities Work and Asset Management V1.9 (v19-6) 28 March 2011

Rule ID: ASSET DEPRECIATION Business Parameter Limit? ☐

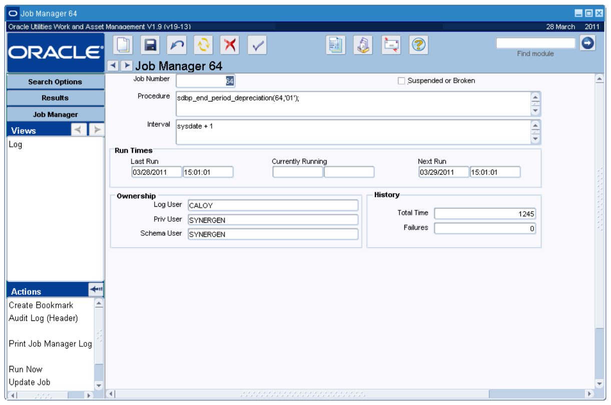
Desc: This rule is for Asset Depreciation Processing.

Comment: Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201012
LAST RUN DATE	28-MAR-11

Description: Year and period last depreciation account log entry (Format: yyyymm).

2. Run the batch job sdbp_end_period_depreciation. The batch job picks-up the LAST PERIOD PROCESSED value and processes the next period, which is Year 2011 Period.



Job Manager 64

Job Number	64
Suspended or Broken	☐
Procedure	sdbp_end_period_depreciation(64,'01');
Interval	sysdate + 1

Run Times

Last Run		Currently Running		Next Run	
03/28/2011	15:01:01			03/29/2011	15:01:01

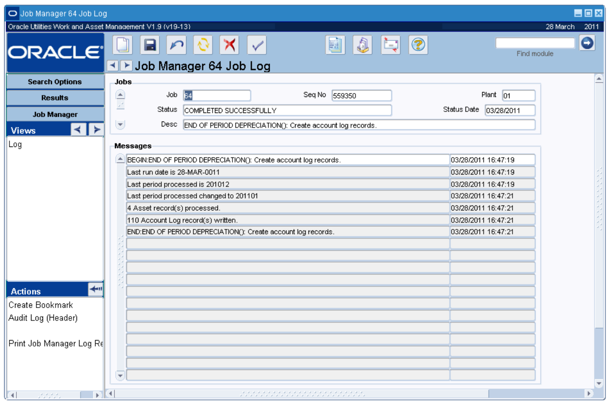
Ownership

Log User	CALOY
Priv User	SYNERGEN
Schema User	SYNERGEN

History

Total Time	1245
Failures	0

3. Make sure that the batch job ran successfully.



Job Manager 64 Job Log

Jobs

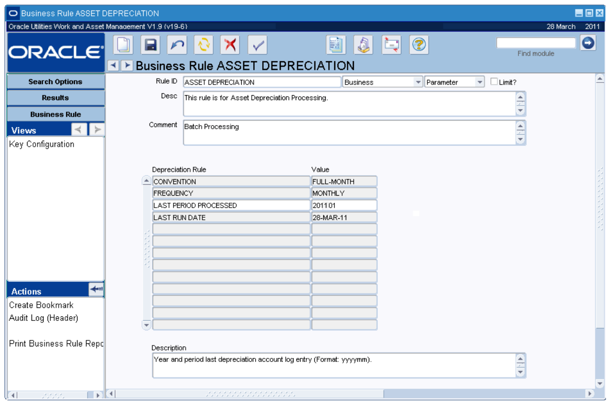
Job	64	Seq No	559350	Plant	01
Status	COMPLETED SUCCESSFULLY			Status Date	03/28/2011
Desc	END OF PERIOD DEPRECIATION(): Create account log records.				

Messages

```
BEGIN:END OF PERIOD DEPRECIATION(): Create account log records.   03/28/2011 16:47:19
Last run date is 28-MAR-0011                                      03/28/2011 16:47:19
Last period processed is 201012                                   03/28/2011 16:47:19
Last period processed changed to 201101                           03/28/2011 16:47:21
4 Asset record(s) processed.                                      03/28/2011 16:47:21
110 Account Log record(s) written.                                03/28/2011 16:47:21
END:END OF PERIOD DEPRECIATION(): Create account log records.     03/28/2011 16:47:21
```

Verify that the Asset was Processed Correctly

1. Navigate back to the ASSET DEPRECIATION Business Rule. Make sure that the LAST PERIOD PROCESSED is incremented by one period and the LAST RUN DATE is updated with Sysdate.



Business Rule ASSET DEPRECIATION

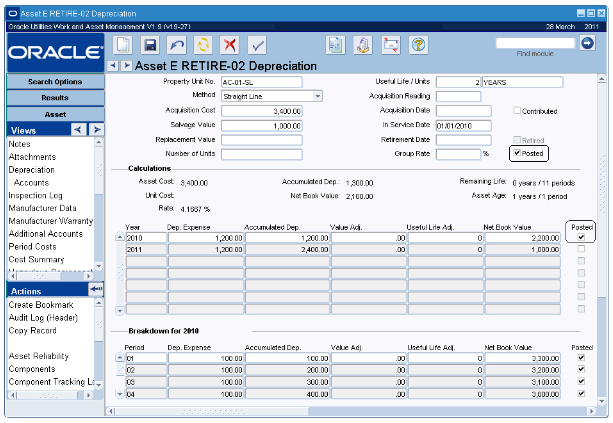
Rule ID	ASSET DEPRECIATION
	Business
	Parameter
Limit?	☐
Desc	This rule is for Asset Depreciation Processing.
Comment	Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201101
LAST RUN DATE	28-MAR-11

Description

Year and period last depreciation account log entry (Format: yyyymm).

2. Navigate back to the Depreciation view of the Asset module. The Posted box in the Fixed Asset section is checked. The Year 2010 in the Calculations section is also checked, meaning that all the forecast records for the year is posted.



Asset E RETIRE-02 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/01/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

Contributed ☐ Retired ☐ Posted ✓

Calculations

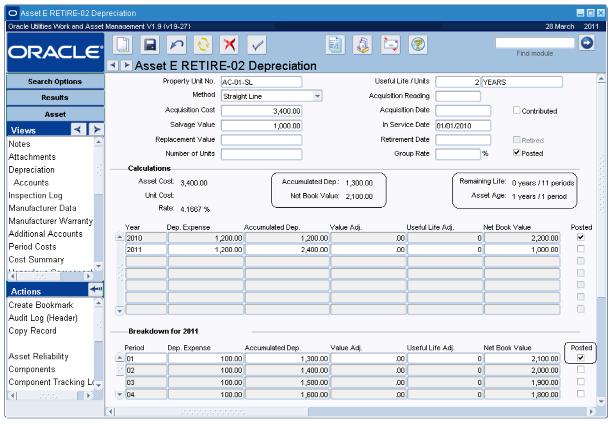
Asset Cost: 3,400.00 Accumulated Dep.: 1,300.00 Remaining Life: 0 years / 11 periods
Unit Cost: Net Book Value: 2,100.00 Asset Age: 1 years / 1 period
Rate: 4.1667 %

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	1,200.00	1,200.00	.00	0	2,200.00	✓
2011	1,200.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	100.00	.00	0	3,300.00	✓
02	100.00	200.00	.00	0	3,200.00	✓
03	100.00	300.00	.00	0	3,100.00	✓
04	100.00	400.00	.00	0	3,000.00	✓

3. Highlight the year 2004. The forecast for Year 2011 Period 01 is posted as a result of running the batch job above. Note that the Remaining Life, Asset Age, Accumulated Dep and Net Book Value are updated.



Asset E RETIRE-02 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	3,400.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/01/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

Contributed ☐ Retired ☐ Posted ✓

Calculations

Asset Cost: 3,400.00 Accumulated Dep.: 1,300.00 Remaining Life: 0 years / 11 periods
Unit Cost: Net Book Value: 2,100.00 Asset Age: 1 years / 1 period
Rate: 4.1667 %

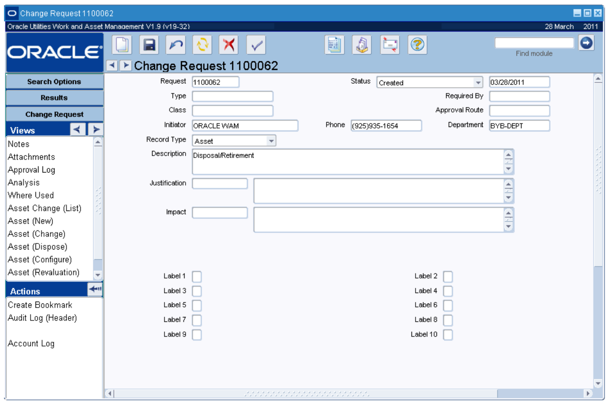
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	1,200.00	1,200.00	.00	0	2,200.00	✓
2011	1,200.00	2,400.00	.00	0	1,000.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	1,300.00	.00	0	2,100.00	✓
02	100.00	1,400.00	.00	0	2,000.00	☐
03	100.00	1,500.00	.00	0	1,900.00	☐
04	100.00	1,600.00	.00	0	1,800.00	☐

Create a Change Request

1. Create a Change Request with Asset as Record Type.



Change Request 1100062

Request	1100062	Status	Created		03/28/2011
Type				Required By	
Class				Approval Route	
Initiator	ORACLE WAM	Phone	(925)935-1654	Department	BYB-DEPT
Record Type	Asset				

Description: Disposal/Retirement
Justification:
Impact:

Label 1 ☐ Label 2 ☐
Label 3 ☐ Label 4 ☐
Label 5 ☐ Label 6 ☐
Label 7 ☐ Label 8 ☐
Label 9 ☐ Label 10 ☐

2. Navigate to Asset (Dispose) view and enter the asset that was created in the first step in this procedure above.
Create a record based on the screen below.

2.1 The Calculated Net Book Value was set $1,680 after entering the Effective Date

2.1.1 The Net Book Value for the Year and Period of the Retirement Date (i.e. Year 2011, Period 06) from the forecast is $1,600. The Depreciation Expense for the same Year and Period is $100.

2.1.2 Partial Percentage is 20%

2.1.2.1 Retirement Date – Start Date + 1 = June 6, 2011 – June 1, 2011 + 1 = 6

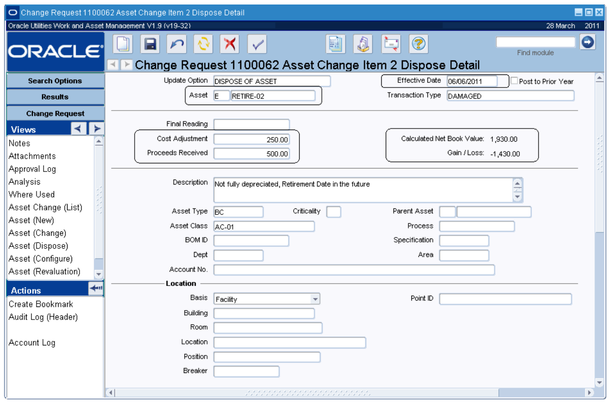
2.1.2.2 End Date – Start Date + 1 = June 30, 2011 – June 1, 2011 + 1 = 30

2.1.2.3 Partial Percentage = 6/30 * 100 = 20%

2.1.3 Calculated Net Book Value = Net Book Value + Depreciation Expense – (Depreciation Expense * Partial Percentage /100) = $1,600 + $100 – ($100 * 20/100) = $1,680

2.2 Note that the Calculated Net Book Value got changed when the Cost Adjustment of $250 was entered. Calculated Net Book Value = $1,680 + $250 = $1,930. The Gain/Loss is the reverse of Calculated Net Book Value.

2.3 The Gain/Loss is the difference of the Proceeds Received and Calculated Net Book Value.

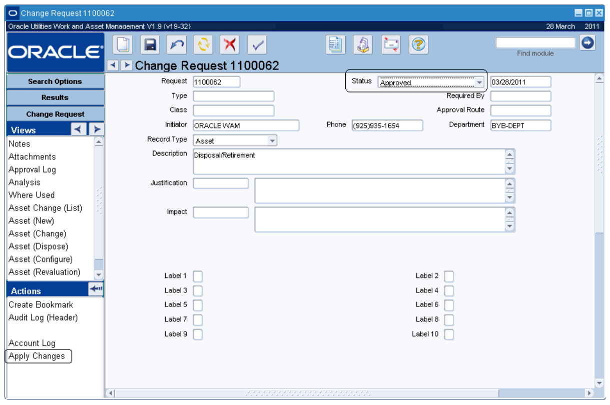

Change Request 1100062 Asset Change Item 2 Dispose Detail

Oracle Utilities Work and Asset Management V1.9 (v19-32) 28 March 2011

Change Request 1100062 Asset Change Item 2 Dispose Detail

Update Option	DISPOSE OF ASSET
Effective Date	06/06/2011
Post to Prior Year	☐
Asset	E RETIRE-02
Transaction Type	DAMAGED
Final Reading	
Cost Adjustment	250.00
Proceeds Received	500.00
Calculated Net Book Value:	1,930.00
Gain / Loss:	-1,430.00
Description	Not fully depreciated, Retirement Date in the future
Asset Type	BC
Criticality	
Parent Asset	
Asset Class	AC-01
Process	
BOM ID	
Specification	
Dept	
Area	
Account No.	
Location	
Basis	Facility
Point ID	
Building	
Room	
Location	
Position	
Breaker	

3. Navigate back to the Change Request header and change the status from Created to Approved. The Apply Changes action appears.

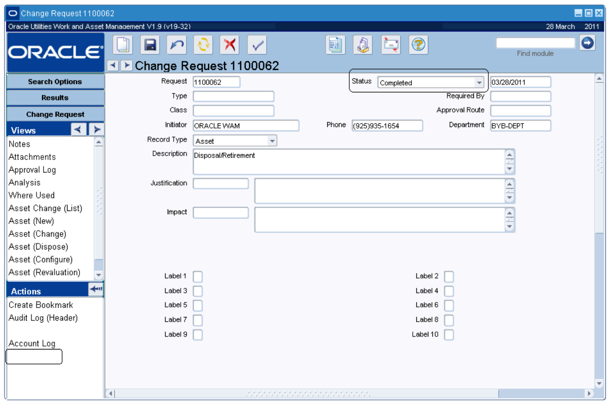

Change Request 1100062

Oracle Utilities Work and Asset Management V1.9 (v19-32) 28 March 2011

Change Request 1100062

Request	1100062
Status	Approved 03/28/2011
Type	
Required By	
Class	
Approval Route	
Initiator	ORACLE WAM
Phone	(925)935-1654
Department	BYB-DEPT
Record Type	Asset
Description	Disposal/Retirement
Justification	
Impact	
Label 1	☐
Label 2	☐
Label 3	☐
Label 4	☐
Label 5	☐
Label 6	☐
Label 7	☐
Label 8	☐
Label 9	☐
Label 10	☐

4. Click on the Apply Changes action. The status changed from Approved to Completed and Apply Changes action disappears.



Change Request 1100062

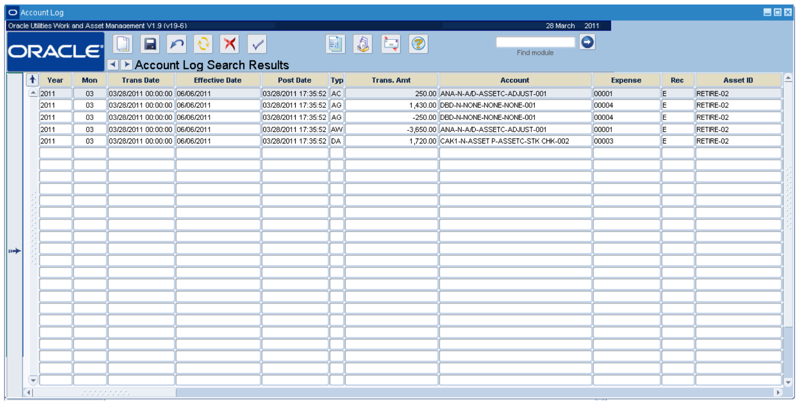
Oracle Utilities Work and Asset Management V1.9 (v19-32) 28 March 2011

Change Request 1100062

Request	1100062	Status	Completed 03/28/2011
Type		Required By	
Class		Approval Route	
Initiator	ORACLE WAM	Phone	(925)935-1654
Department	BYB-DEPT		
Record Type	Asset		
Description	Disposal/Retirement		
Justification			
Impact			
Label 1	☐	Label 2	☐
Label 3	☐	Label 4	☐
Label 5	☐	Label 6	☐
Label 7	☐	Label 8	☐
Label 9	☐	Label 10	☐

5. Click on the Account Log action from the Change Request header screen. The Account Log module is displayed.
   - 5.1 Transaction Date is the Change Request Status Date.
   - 5.2 Year and Month is the Year and Period of the Transaction Date in Accounting Period table.
   - 5.3 Effective Date is the Change Request item Effective Date.
   - 5.4 Post Date is Sysdate.
   - 5.5 The Cost Adjustment is recorded by the AC and AG transactions. The former is the positive value and the latter is the negative.
   - 5.6 The other AG (Asset Gain Loss) transaction records Step 2.3.
   - 5.7 The Accumulated Depreciation (DA) is $2,400, equal to the Accumulated Dep. of the last forecast record of the asset.
   - 5.8 AW (Asset Cost Write-off) is the sum of the Acquisition Cost and the sum of all value adjustments in the forecast. Multiply the sum by –1.
   - 5.9 The Asset Accumulated Depreciation amount is $1,720.
     - 5.9.1 The Accumulated Dep. and Dep. Expense of the asset for the Year 2011 Period 06 before the Change Request was applied were $1,800 and $100, respectively.
     - 5.9.2 The Partial Percentage is 20%, refer to 10.1.2 above.

5.9.3 The DA Transaction Amount is Accumulated Dep. less Dep. Expense plus product of Dep. Expense and Partial Percentage. So that is \$1,800 - \$100 + (\$100 * 20 /100) = \$1,720.


Account Log

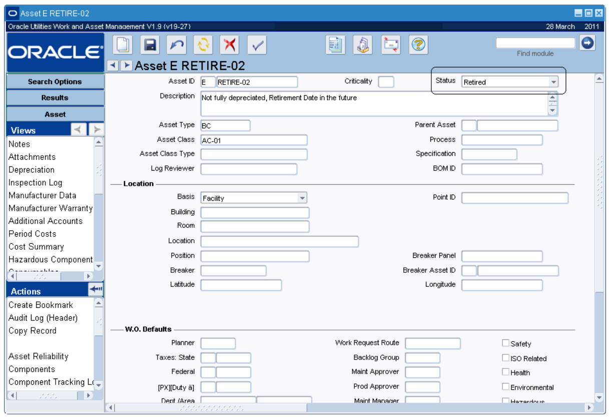
Oracle Utilities Work and Asset Management V1.9 (v19-6) 28 March 2011

Account Log Search Results

Year	Mon	Trans Date	Effective Date	Post Date	Typ	Trans. Amt	Account	Expense	Rec	Asset ID
2011	03	03/28/2011 00:00:00	06/06/2011	03/28/2011 17:35:52	AC	250.00	ANA-N-A/D-ASSETC-ADJUST-001	00001	E	RETIRE-02
2011	03	03/28/2011 00:00:00	06/06/2011	03/28/2011 17:35:52	AG	1,430.00	DBD-N-NONE-NONE-NONE-001	00004	E	RETIRE-02
2011	03	03/28/2011 00:00:00	06/06/2011	03/28/2011 17:35:52	AG	-250.00	DBD-N-NONE-NONE-NONE-001	00004	E	RETIRE-02
2011	03	03/28/2011 00:00:00	06/06/2011	03/28/2011 17:35:52	AW	-3,650.00	ANA-N-A/D-ASSETC-ADJUST-001	00001	E	RETIRE-02
2011	03	03/28/2011 00:00:00	06/06/2011	03/28/2011 17:35:52	DA	1,720.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	00003	E	RETIRE-02

Verify Revised Cost Calculations

1. Navigate back to the Asset module and retrieve the asset created in the first step in this procedure above. The asset's status is now Retired.

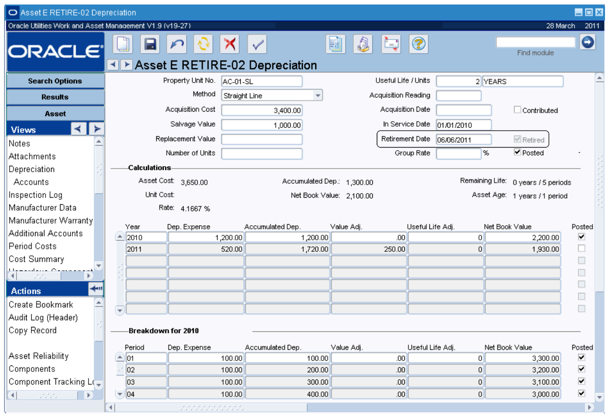

Asset E RETIRE-02

Oracle Utilities Work and Asset Management V1.9 (v19-27) 28 March 2011

Asset E RETIRE-02

Asset ID	E RETIRE-02
Criticality	
Status	Retired
Description	Not fully depreciated, Retirement Date in the future
Asset Type	BC
Asset Class	AC-01
Basis	Facility

2. Navigate to the Depreciation view. The retirement Date is now populated with the Effective Date from the Change Request. The Retired box is checked.


Asset E RETIRE-02 Depreciation

Oracle Utilities Work and Asset Management V1.9 (v19-27) 28 March 2011

Asset E RETIRE-02 Depreciation

Property Unit No.	AC-01-SL
Method	Straight Line
Acquisition Cost	3,400.00
Salvage Value	1,000.00
Useful Life / Units	2 YEARS
In Service Date	01/01/2010
Retirement Date	06/06/2011 ✓ Retired
Group Rate	% ✓ Posted

Calculations

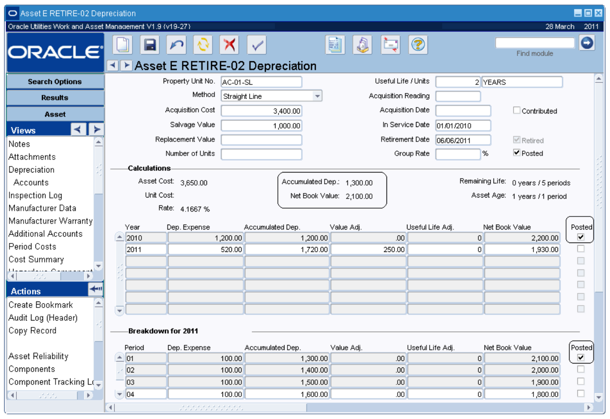
Asset Cost:	3,650.00
Rate:	4.1667 %
Accumulated Dep.:	1,300.00
Net Book Value:	2,100.00
Remaining Life:	0 years / 5 periods
Asset Age:	1 years / 1 period

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	1,200.00	1,200.00	.00	0	2,200.00	✓
2011	520.00	1,720.00	250.00	0	1,930.00	

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	100.00	.00	0	3,300.00	✓
02	100.00	200.00	.00	0	3,200.00	✓
03	100.00	300.00	.00	0	3,100.00	✓
04	100.00	400.00	.00	0	3,000.00	✓

3. All the forecast records for Year 2010 are still posted.
   The forecast record for Year 2011 Period 01 is still posted. The Accumulated Dep. and Net Book Value still reflect the values from the last posted forecast record.



Asset E RETIRE-02 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS	
Method	Straight Line	Acquisition Reading		
Acquisition Cost	3,400.00	Acquisition Date		☐ Contributed
Salvage Value	1,000.00	In Service Date	01/01/2010	
Replacement Value		Retirement Date	06/06/2011	✓ Retired
Number of Units		Group Rate	%	✓ Posted

Calculations

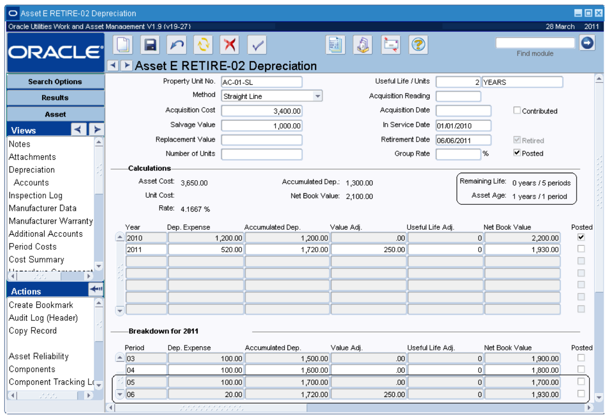
Asset Cost:	3,650.00	Accumulated Dep.:	1,300.00	Remaining Life:	0 years / 5 periods
Unit Cost:		Net Book Value:	2,100.00	Asset Age:	1 years / 1 period
Rate:	4.1667 %				

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	1,200.00	1,200.00	.00	0	2,200.00	✓
2011	520.00	1,720.00	250.00	0	1,930.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	1,300.00	.00	0	2,100.00	✓
02	100.00	1,400.00	.00	0	2,000.00	☐
03	100.00	1,500.00	.00	0	1,900.00	☐
04	100.00	1,600.00	.00	0	1,800.00	☐

4. Highlight 2011 and scroll the Breakdown until the last record is reached.
   Note that the forecast has shortened. All the unposted records after the Retirement Date (i.e. Year 2011 Period 07 onwards) are deleted. Hence the Remaining Life decreased from 11 periods to 5 periods (Periods 06 to 12 were deleted for a total of 6 periods). The Asset Age remained the same.



Asset E RETIRE-02 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS	
Method	Straight Line	Acquisition Reading		
Acquisition Cost	3,400.00	Acquisition Date		☐ Contributed
Salvage Value	1,000.00	In Service Date	01/01/2010	
Replacement Value		Retirement Date	06/06/2011	✓ Retired
Number of Units		Group Rate	%	✓ Posted

Calculations

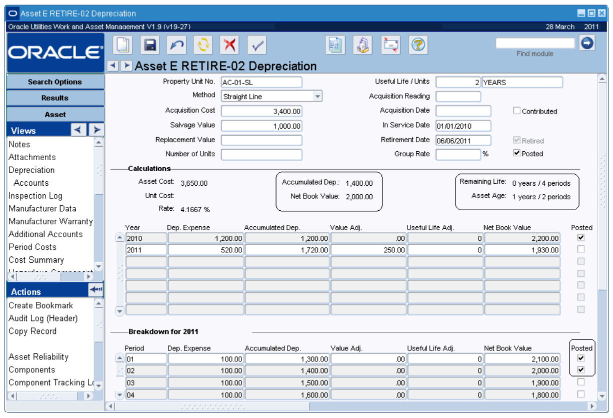
Asset Cost:	3,650.00	Accumulated Dep.:	1,300.00	Remaining Life:	0 years / 5 periods
Unit Cost:		Net Book Value:	2,100.00	Asset Age:	1 years / 1 period
Rate:	4.1667 %				

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	1,200.00	1,200.00	.00	0	2,200.00	✓
2011	520.00	1,720.00	250.00	0	1,930.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
03	100.00	1,500.00	.00	0	1,900.00	☐
04	100.00	1,600.00	.00	0	1,800.00	☐
05	100.00	1,700.00	.00	0	1,700.00	☐
06	20.00	1,720.00	250.00	0	1,930.00	☐

5. Run sdbp_end_period_depreciation. Navigate back to the Asset's Depreciation view. Year 2011 Period 02 is now posted. The Accumulated Dep. and Net Book Value are the same as the values from the last posted forecast record. The Remaining Life decreased by a period. The Asset Age increased by a period.



Asset E RETIRE-02 Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	2 YEARS	
Method	Straight Line	Acquisition Reading		
Acquisition Cost	3,400.00	Acquisition Date		☐ Contributed
Salvage Value	1,000.00	In Service Date	01/01/2010	
Replacement Value		Retirement Date	06/06/2011	✓ Retired
Number of Units		Group Rate	%	✓ Posted

Calculations

Asset Cost:	3,650.00	Accumulated Dep.:	1,400.00	Remaining Life:	0 years / 4 periods
Unit Cost:		Net Book Value:	2,000.00	Asset Age:	1 years / 2 periods
Rate:	4.1667 %				

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	1,200.00	1,200.00	.00	0	2,200.00	✓
2011	520.00	1,720.00	250.00	0	1,930.00	☐

Breakdown for 2011

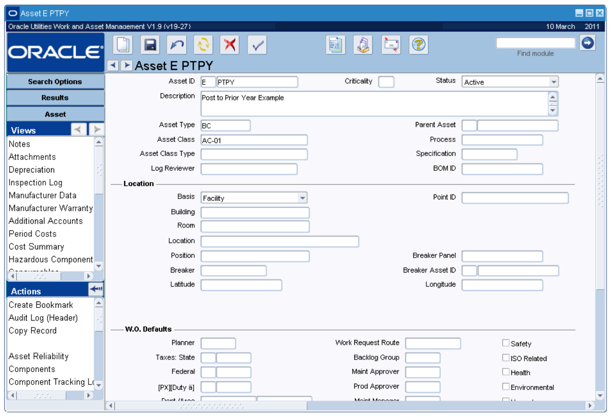
Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	1,300.00	.00	0	2,100.00	✓
02	100.00	1,400.00	.00	0	2,000.00	✓
03	100.00	1,500.00	.00	0	1,900.00	☐
04	100.00	1,600.00	.00	0	1,800.00	☐

Post to Prior Year

Post costs for assets to the prior year by completing the following steps:

Create the Asset and Set Depreciation Settings

1. Create an asset. Make sure that the Asset Class and its Depreciation Parameters are set-up.



Asset E PTPY

Asset ID	E PTPY	Status	Active
Description	Post to Prior Year Example	Criticality	
Asset Type	BC	Parent Asset	
Asset Class	AC-01	Process	
Asset Class Type		Specification	
Log Reviewer		BOM ID	

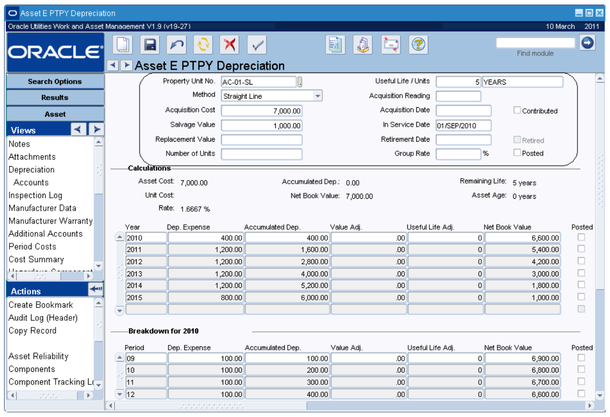
Location

Basis	Facility	Point ID	
Building			
Room			
Location			
Position		Breaker Panel	
Breaker		Breaker Asset ID	
Latitude		Longitude	

W.O. Defaults

Planner		Work Request Route		☐ Safety
Taxes: State		Backlog Group		☐ ISO Related
Federal		Maint Approver		☐ Health
[PX][Duty ä]		Prod Approver		☐ Environmental

2. Navigate to Depreciation view to create a Fixed Asset record. Populate Fixed Asset based on the screen below. Upon Save, the Calculations and Breakdown sections are automatically populated.



Asset E PTPY Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS	
Method	Straight Line	Acquisition Reading		
Acquisition Cost	7,000.00	Acquisition Date		☐ Contributed
Salvage Value	1,000.00	In Service Date	01/SEP/2010	
Replacement Value		Retirement Date		☐ Retired
Number of Units		Group Rate	%	☐ Posted

Calculations

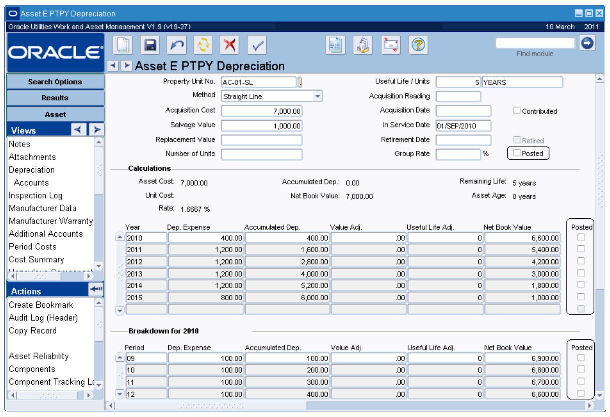
Asset Cost:	7,000.00	Accumulated Dep.:	0.00	Remaining Life:	5 years
Unit Cost:		Net Book Value:	7,000.00	Asset Age:	0 years
Rate:	1.6667 %				

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	400.00	400.00	.00	0	6,600.00	☐
2011	1,200.00	1,600.00	.00	0	5,400.00	☐
2012	1,200.00	2,800.00	.00	0	4,200.00	☐
2013	1,200.00	4,000.00	.00	0	3,000.00	☐
2014	1,200.00	5,200.00	.00	0	1,800.00	☐
2015	800.00	6,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	☐
10	100.00	200.00	.00	0	6,800.00	☐
11	100.00	300.00	.00	0	6,700.00	☐
12	100.00	400.00	.00	0	6,600.00	☐

3. Take note that the Posted indicators are all unchecked.



Asset E PTPY Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS	
Method	Straight Line	Acquisition Reading		
Acquisition Cost	7,000.00	Acquisition Date		☐ Contributed
Salvage Value	1,000.00	In Service Date	01/SEP/2010	
Replacement Value		Retirement Date		☐ Retired
Number of Units		Group Rate	%	☐ Posted

Calculations

Asset Cost:	7,000.00	Accumulated Dep.:	0.00	Remaining Life:	5 years
Unit Cost:		Net Book Value:	7,000.00	Asset Age:	0 years
Rate:	1.6667 %				

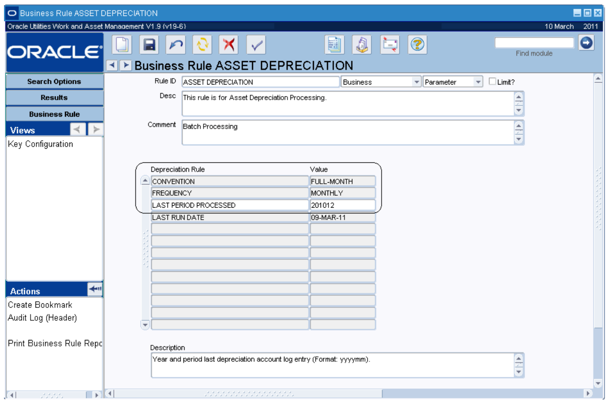
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	400.00	400.00	.00	0	6,600.00	☐
2011	1,200.00	1,600.00	.00	0	5,400.00	☐
2012	1,200.00	2,800.00	.00	0	4,200.00	☐
2013	1,200.00	4,000.00	.00	0	3,000.00	☐
2014	1,200.00	5,200.00	.00	0	1,800.00	☐
2015	800.00	6,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	☐
10	100.00	200.00	.00	0	6,800.00	☐
11	100.00	300.00	.00	0	6,700.00	☐
12	100.00	400.00	.00	0	6,600.00	☐

Modify the Asset Depreciation Business Rule and Run Batch

1. Change the Business Rule ASSET DEPRECIATION. Set LAST PERIOD PROCESSED, CONVENTION and FREQUENCY based on the screen below.



Rule ID: ASSET DEPRECIATION | Business | Parameter | Limit?

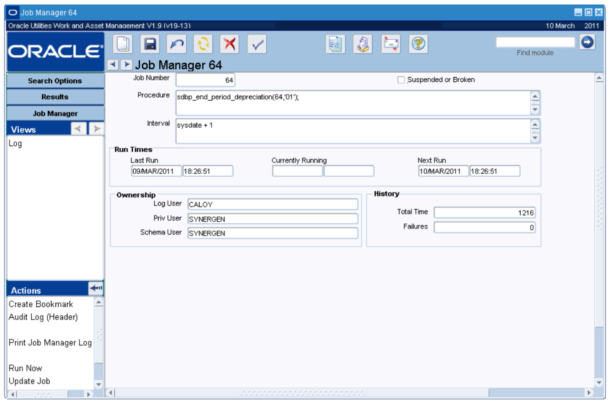
Desc: This rule is for Asset Depreciation Processing.

Comment: Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201012
LAST RUN DATE	09-MAR-11

Description: Year and period last depreciation account log entry (Format: yyyymm).

2. Run the batch job sdbp_end_period_depreciation. The batch job picks-up the LAST PERIOD PROCESSED value and processes the next period, which is Year 2011 Period.



Job Number: 64 | Suspended or Broken

Procedure: sdbp_end_period_depreciation(64,'01');

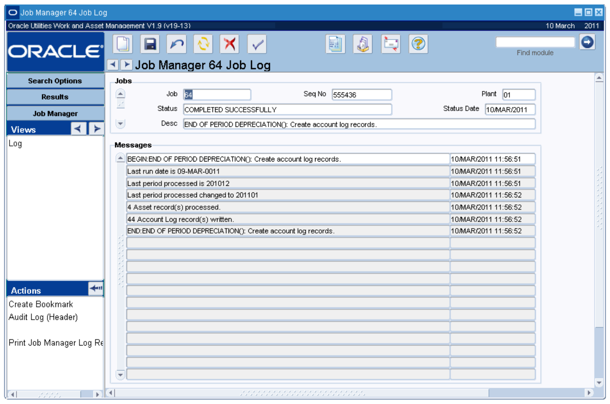
Interval: sysdate + 1

Run Times
Last Run: 09/MAR/2011 18:26:51 | Currently Running: | Next Run: 10/MAR/2011 18:26:51

Ownership
Log User: CALOY
Priv User: SYNERGEN
Schema User: SYNERGEN

History
Total Time: 1216
Failures: 0

3. Make sure that the batch job ran successfully.



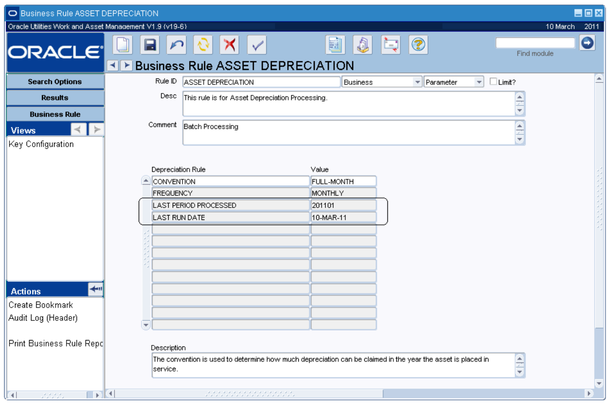
Jobs
Job: 64 | Seq No: 555436 | Plant: 01
Status: COMPLETED SUCCESSFULLY | Status Date: 10/MAR/2011
Desc: END OF PERIOD DEPRECIATION(): Create account log records.

Messages

BEGIN:END OF PERIOD DEPRECIATION(): Create account log records.	10/MAR/2011 11:56:51
Last run date is 09-MAR-0011	10/MAR/2011 11:56:51
Last period processed is 201012	10/MAR/2011 11:56:51
Last period processed changed to 201101	10/MAR/2011 11:56:52
4 Asset record(s) processed.	10/MAR/2011 11:56:52
44 Account Log record(s) written.	10/MAR/2011 11:56:52
END:END OF PERIOD DEPRECIATION(): Create account log records.	10/MAR/2011 11:56:52

Verify that the Asset was Processed Correctly

1. Navigate back to the ASSET DEPRECIATION business rule. Make sure that the LAST PERIOD PROCESSED is incremented by one period and the LAST RUN DATE is updated with Sysdate.



Business Rule ASSET DEPRECIATION

Rule ID: ASSET DEPRECIATION | Business | Parameter | Limit? ☐

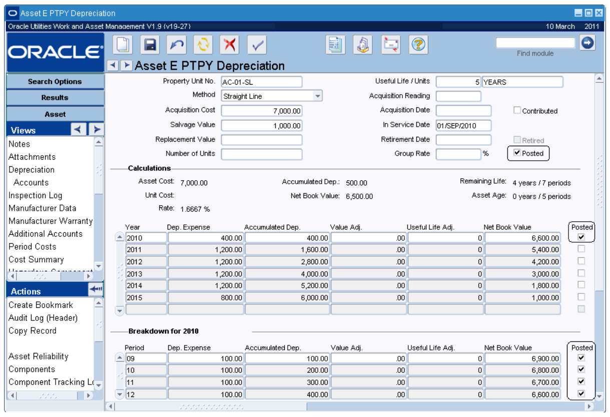
Desc: This rule is for Asset Depreciation Processing.

Comment: Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201101
LAST RUN DATE	10-MAR-11

Description: The convention is used to determine how much depreciation can be claimed in the year the asset is placed in service.

2. Navigate back to the Depreciation view of the Asset module. Note the Posted check boxes. If it is checked, it means that the asset has been processed for the end of period and Account Log entries have been written against the asset.



Asset E PTPY Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ✓ Posted

Calculations

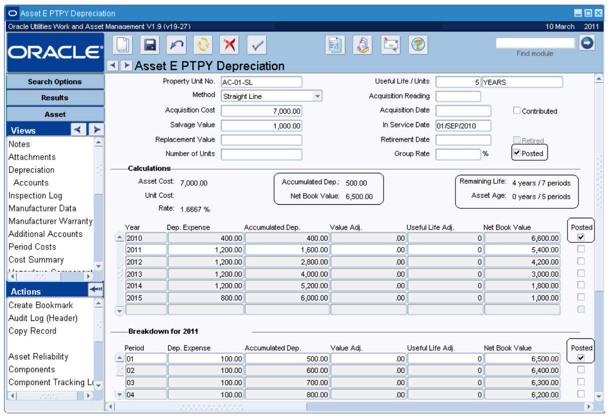
Asset Cost: 7,000.00 Accumulated Dep.: 500.00 Remaining Life: 4 years / 7 periods
Unit Cost: Net Book Value: 6,500.00 Asset Age: 0 years / 5 periods
Rate: 1.6667 %

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	400.00	400.00	.00	0	6,600.00	✓
2011	1,200.00	1,600.00	.00	0	5,400.00	☐
2012	1,200.00	2,800.00	.00	0	4,200.00	☐
2013	1,200.00	4,000.00	.00	0	3,000.00	☐
2014	1,200.00	5,200.00	.00	0	1,800.00	☐
2015	800.00	6,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	✓
10	100.00	200.00	.00	0	6,800.00	✓
11	100.00	300.00	.00	0	6,700.00	✓
12	100.00	400.00	.00	0	6,600.00	✓

3. Note that the Year 2011, Period 01 is also posted. The information right below the Calculations heading are updated as well.



Asset E PTPY Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ✓ Posted

Calculations

Asset Cost: 7,000.00 Accumulated Dep.: 500.00 Remaining Life: 4 years / 7 periods
Unit Cost: Net Book Value: 6,500.00 Asset Age: 0 years / 5 periods
Rate: 1.6667 %

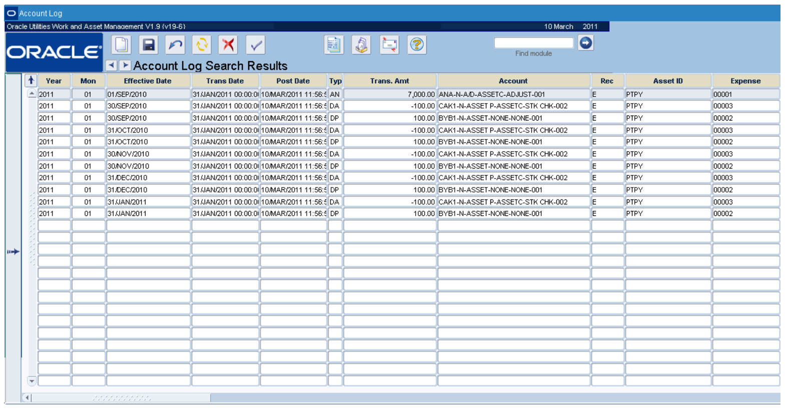
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	400.00	400.00	.00	0	6,600.00	✓
2011	1,200.00	1,600.00	.00	0	5,400.00	☐
2012	1,200.00	2,800.00	.00	0	4,200.00	☐
2013	1,200.00	4,000.00	.00	0	3,000.00	☐
2014	1,200.00	5,200.00	.00	0	1,800.00	☐
2015	800.00	6,000.00	.00	0	1,000.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	500.00	.00	0	6,500.00	✓
02	100.00	600.00	.00	0	6,400.00	☐
03	100.00	700.00	.00	0	6,300.00	☐
04	100.00	800.00	.00	0	6,200.00	☐

4. Navigate to the Account Log and search for the Asset.
The Transaction Date is the End Date of the Year and Period that was processed by the batch job. In this case, the Year and Period that was last processed was 2011 and 01, respectively. The End Date of Year 2011 and Period 01 in the Accounting Period is 31-Jan-2011; hence the Transaction Date is 31/Jan/2011. The Post Date is the Sysdate. The

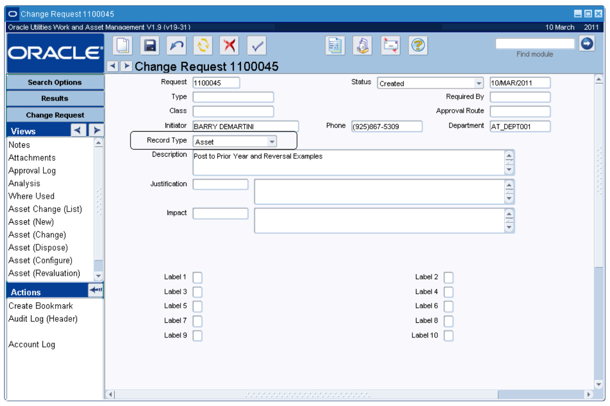
Effective Date is the End Date of the Year and Period of the forecast record that was being posted. The Transaction Type AN (Asset New) was created the first time the asset was posted. The pair of Asset Depreciation/Asset Accumulated Depreciation Transaction Types (DP/DA) with the amount of ?$100 is the Depreciation Expense for the Forecast's Year and Period.


Account Log
Oracle Utilities Work and Asset Management V1.9 (v19-6) 10 March 2011
Account Log Search Results

Year	Mon	Effective Date	Trans Date	Post Date	Typ	Trans. Amt	Account	Rec	Asset ID	Expense
2011	01	01/SEP/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	AN	7,000.00	ANA-N-A/D-ASSETC-ADJUST-001	E	PTPY	00001
2011	01	30/SEP/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	01	30/SEP/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	01	31/OCT/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	01	31/OCT/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	01	30/NOV/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	01	30/NOV/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	01	31/DEC/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	01	31/DEC/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	01	31/JAN/2011	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	01	31/JAN/2011	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002

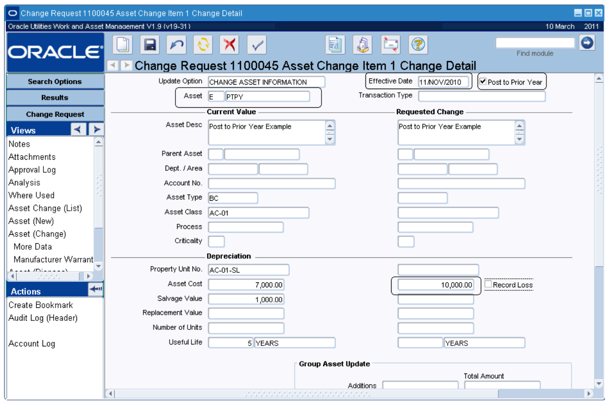
Create a Change Request

1. Create a Change Request with Asset as Record Type.


Change Request 1100045
Oracle Utilities Work and Asset Management V1.9 (v19-31) 10 March 2011
Change Request 1100045

Request	1100045	Status	Created		10/MAR/2011
Type				Required By	
Class				Approval Route	
Initiator	BARRY DEMARTINI	Phone	(925)867-5309	Department	AT_DEPT001
Record Type	Asset				
Description	Post to Prior Year and Reversal Examples				
Justification					
Impact					

2. Navigate to Asset (Change) view and enter the asset that was created in the first step in this procedure above. Create a record based on the screen below. Increase the asset cost from $7,000 to $10,000. Make sure that the Post to Prior Year box is checked. Save the record.


Change Request 1100045 Asset Change Item 1 Change Detail
Oracle Utilities Work and Asset Management V1.9 (v19-31) 10 March 2011
Change Request 1100045 Asset Change Item 1 Change Detail

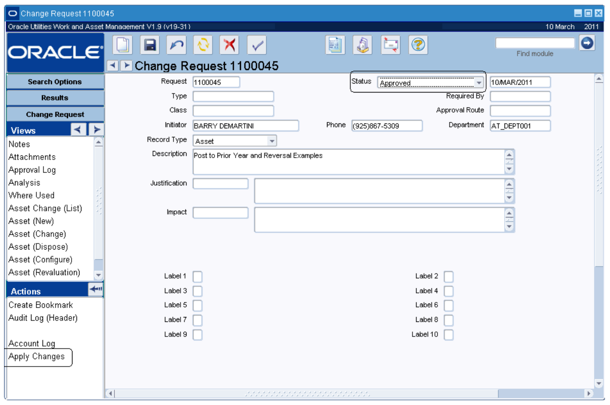
Update Option: CHANGE ASSET INFORMATION
Effective Date: 11/NOV/2010
✓ Post to Prior Year
Asset: E PTPY
Transaction Type:

	Current Value	Requested Change
Asset Desc	Post to Prior Year Example	Post to Prior Year Example
Parent Asset		
Dept. / Area		
Account No.		
Asset Type	BC	
Asset Class	AC-01	
Process		
Criticality		
Depreciation		
Property Unit No.	AC-01-SL	
Asset Cost	7,000.00	10,000.00
Salvage Value	1,000.00	
Replacement Value		
Number of Units		
Useful Life	5 YEARS	YEARS

Record Loss

Group Asset Update
Total Amount
Additions

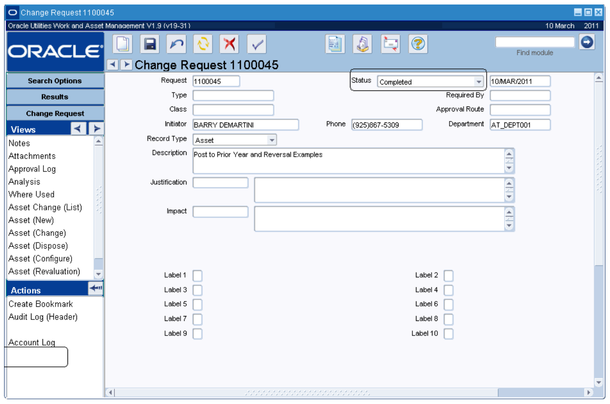
3. Navigate back to the Change Request header and change the status from Created to Approved. The Apply Changes action appears.



Change Request 1100045

Request: 1100045
Status: Approved
10/MAR/2011
Type:
Class:
Required By:
Approval Route:
Initiator: BARRY DEMARTINI
Phone: (925)867-5309
Department: AT_DEPT001
Record Type: Asset
Description: Post to Prior Year and Reversal Examples
Justification:
Impact:
Label 1 ☐ Label 2 ☐ Label 3 ☐ Label 4 ☐ Label 5 ☐ Label 6 ☐ Label 7 ☐ Label 8 ☐ Label 9 ☐ Label 10 ☐

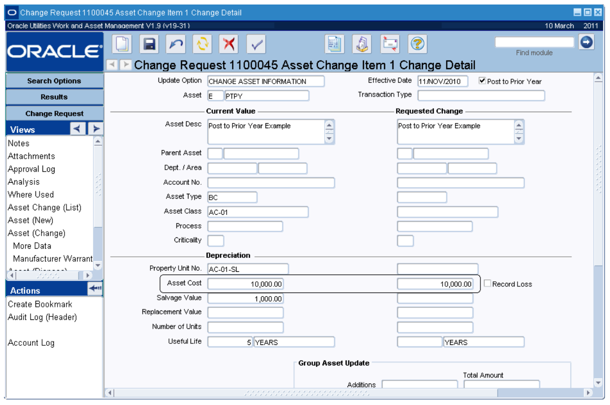
4. Click on the Apply Changes action. The status changed from Approved to Completed and Apply Changes action disappears.



Change Request 1100045

Request: 1100045
Status: Completed
10/MAR/2011
Type:
Class:
Required By:
Approval Route:
Initiator: BARRY DEMARTINI
Phone: (925)867-5309
Department: AT_DEPT001
Record Type: Asset
Description: Post to Prior Year and Reversal Examples
Justification:
Impact:
Label 1 ☐ Label 2 ☐ Label 3 ☐ Label 4 ☐ Label 5 ☐ Label 6 ☐ Label 7 ☐ Label 8 ☐ Label 9 ☐ Label 10 ☐

5. Navigate back to the Asset (Change) screen. The Asset Cost field should now be the same for both Current Value and Requested Change columns.



Change Request 1100045 Asset Change Item 1 Change Detail

Update Option: CHANGE ASSET INFORMATION
Effective Date: 11/NOV/2010
☑ Post to Prior Year
Asset: E PTPY
Transaction Type:

	Current Value	Requested Change
Asset Desc	Post to Prior Year Example	Post to Prior Year Example
Parent Asset		
Dept. / Area		
Account No.		
Asset Type	BC	
Asset Class	AC-01	
Process		
Criticality		
Depreciation		
Property Unit No.	AC-01-SL	
Asset Cost	10,000.00	10,000.00
Salvage Value	1,000.00	
Replacement Value		
Number of Units		
Useful Life	5 YEARS	YEARS

☐ Record Loss

Group Asset Update
Total Amount
Additions

6. Navigate back to the Change Request Header and click on the Account Log action.
   6.1 **Green**:
      6.1.1 Transaction Type AC (Asset Cost Change in Value) reflects the increase or decrease in the value of the asset.
      6.1.2 The Transaction Date is the End Date of the last period of the Prior Year.
      6.1.3 Year and Mon is the Year and Period of the Transaction Date in the Accounting Period.

6.1.4 Effective Date is from the Change Request Asset (Change) Item Detail.

6.2 **Yellow/Purple**:

6.2.1 Reversed records.

6.2.2 Yellow is the reversed DA transaction and Purple is the reversed DP transaction. Compare the reversed records to the records in #10 when the asset was posted for the first time.

6.2.3 The Transaction Date is the End Date of the last period of the Prior Year if the year of the forecast record is the Prior Year. Otherwise, Transaction Date is the Change Request Status Date.

6.2.4 Year and Mon is the Year and Period of the Transaction Date in the Accounting Period.

6.2.5 Effective Date is the End Date for the Year and Period of the reversed Forecast record.

6.3 **Blue/Gray**:

6.3.1 Newly posted records.

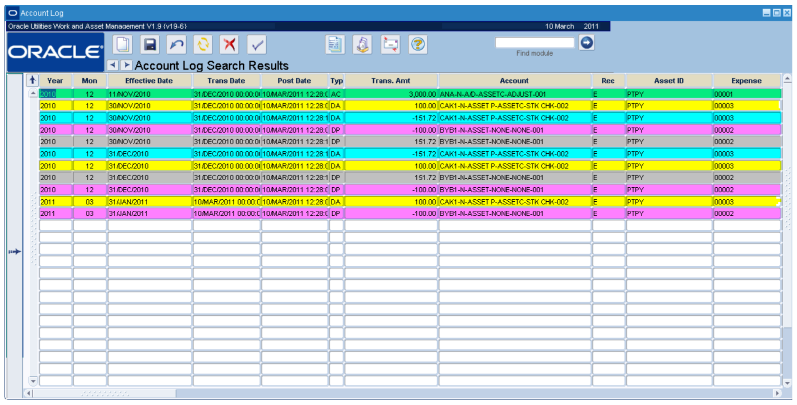
6.3.2 Blue is the DA transaction. Gray is the DP. Note that the DA is negative and DP is positive, the opposite of 16.2.2 above.

6.3.3 The Transaction Date is the End Date of the last period of the Prior Year.

6.3.4 Year and Mon is the Year and Period of the Transaction Date in the Accounting Period.

6.3.5 Effective Date is the End Date of the Year and Period of the newly posted Forecast record.

6.4 Account Log entries for Year 2010 Periods 09 and 10 purposely not included.



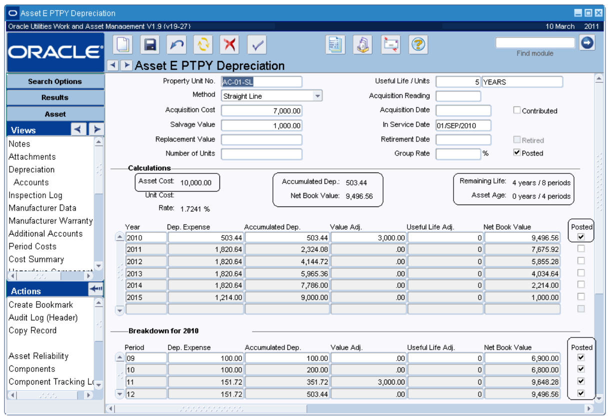
Account Log
Oracle Utilities Work and Asset Management V1.9 (v19-6) 10 March 2011
Account Log Search Results

Year	Mon	Effective Date	Trans Date	Post Date	Typ	Trans. Amt	Account	Rec	Asset ID	Expense
2010	12	11/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	AC	3,000.00	ANA-N-A/D-ASSETC-ADJUST-001	E	PTPY	00001
2010	12	30/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2010	12	30/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:1	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2010	12	30/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2010	12	30/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:1	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2010	12	31/DEC/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:1	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2010	12	31/DEC/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2010	12	31/DEC/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:1	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2010	12	31/DEC/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	03	31/JAN/2011	10/MAR/2011 00:00:0	10/MAR/2011 12:28:0	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	03	31/JAN/2011	10/MAR/2011 00:00:0	10/MAR/2011 12:28:0	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002

Verify Revised Cost Calculations

1. Navigate back to the Asset module and retrieve the asset created in #1. Navigate to the Depreciation view. The Asset Cost reflects the new asset cost. The Acquisition Cost remained at $7,000. The Acquisition Cost does not get updated if the asset has been previously posted. The Accumulated Dep. and Net Book Value reflect the Accum Dep.

And Net Book Value of the last posted forecast record. The Remaining Life increased and the Asset Age decreased by just one period. All the records for the year 2010 are posted.



Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed
☐ Retired
☑ Posted

Calculations

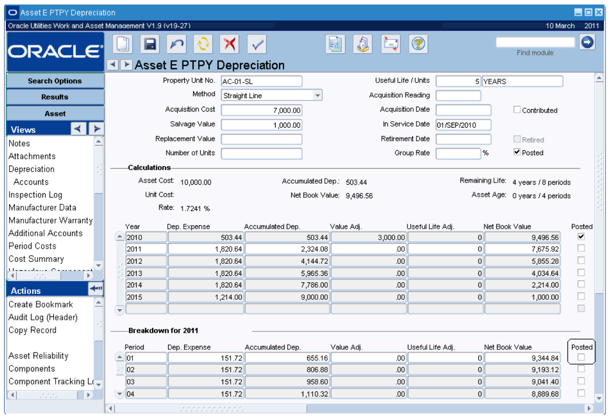
Asset Cost: 10,000.00
Unit Cost:
Rate: 1.7241 %
Accumulated Dep.: 503.44
Net Book Value: 9,496.56
Remaining Life: 4 years / 8 periods
Asset Age: 0 years / 4 periods

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	503.44	503.44	3,000.00	0	9,496.56	✓
2011	1,820.64	2,324.08	.00	0	7,675.92	☐
2012	1,820.64	4,144.72	.00	0	5,855.28	☐
2013	1,820.64	5,965.36	.00	0	4,034.64	☐
2014	1,820.64	7,786.00	.00	0	2,214.00	☐
2015	1,214.00	9,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	✓
10	100.00	200.00	.00	0	6,800.00	✓
11	151.72	351.72	3,000.00	0	9,648.28	✓
12	151.72	503.44	.00	0	9,496.56	✓

2. Highlight Year 2011. Note that Period 01 remains unposted.



Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed
☐ Retired
☑ Posted

Calculations

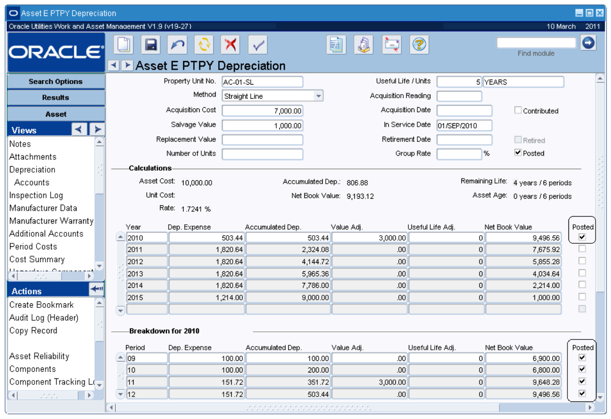
Asset Cost: 10,000.00
Unit Cost:
Rate: 1.7241 %
Accumulated Dep.: 503.44
Net Book Value: 9,496.56
Remaining Life: 4 years / 8 periods
Asset Age: 0 years / 4 periods

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	503.44	503.44	3,000.00	0	9,496.56	✓
2011	1,820.64	2,324.08	.00	0	7,675.92	☐
2012	1,820.64	4,144.72	.00	0	5,855.28	☐
2013	1,820.64	5,965.36	.00	0	4,034.64	☐
2014	1,820.64	7,786.00	.00	0	2,214.00	☐
2015	1,214.00	9,000.00	.00	0	1,000.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	151.72	655.16	.00	0	9,344.84	☐
02	151.72	806.88	.00	0	9,193.12	☐
03	151.72	958.60	.00	0	9,041.40	☐
04	151.72	1,110.32	.00	0	8,889.68	☐

3. Rerun the batch job and make sure that it ran successfully. Make sure that the LAST PERIOD PROCESSED is incremented by one period (i.e. 201102) and the LAST RUN DATE is updated with Sysdate. Navigate back to the Depreciation view of the Asset module. All periods for the year 2010 remain posted.



Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed
☐ Retired
☑ Posted

Calculations

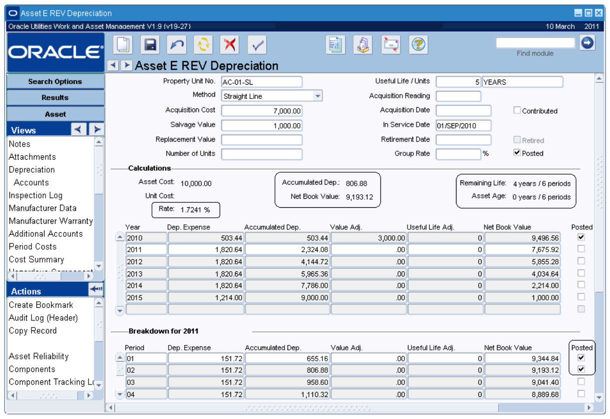
Asset Cost: 10,000.00
Unit Cost:
Rate: 1.7241 %
Accumulated Dep.: 806.88
Net Book Value: 9,193.12
Remaining Life: 4 years / 6 periods
Asset Age: 0 years / 6 periods

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	503.44	503.44	3,000.00	0	9,496.56	✓
2011	1,820.64	2,324.08	.00	0	7,675.92	☐
2012	1,820.64	4,144.72	.00	0	5,855.28	☐
2013	1,820.64	5,965.36	.00	0	4,034.64	☐
2014	1,820.64	7,786.00	.00	0	2,214.00	☐
2015	1,214.00	9,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	✓
10	100.00	200.00	.00	0	6,800.00	✓
11	151.72	351.72	3,000.00	0	9,648.28	✓
12	151.72	503.44	.00	0	9,496.56	✓

4. Highlight the Year 2011. The Rate, Accumulated Dep., Net Book Value, Remaining Life and Asset Age were updated. The system 'caught-up' by posting the Year 2011 Period 01 forecast record and the next period as well.



Asset E REV Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed
☐ Retired
✓ Posted

Calculations

Asset Cost: 10,000.00
Unit Cost:
Rate: 1.7241 %
Accumulated Dep.: 806.88
Net Book Value: 9,193.12
Remaining Life: 4 years / 6 periods
Asset Age: 0 years / 6 periods

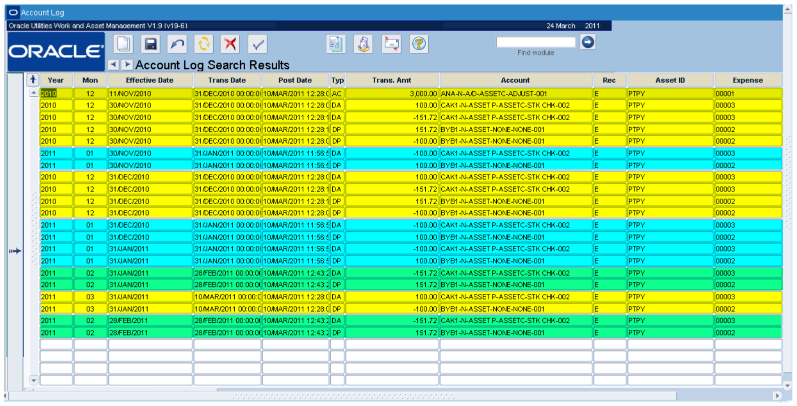
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	503.44	503.44	3,000.00	0	9,496.56	✓
2011	1,820.64	2,324.08	.00	0	7,675.92	☐
2012	1,820.64	4,144.72	.00	0	5,855.28	☐
2013	1,820.64	5,965.36	.00	0	4,034.64	☐
2014	1,820.64	7,786.00	.00	0	2,214.00	☐
2015	1,214.00	9,000.00	.00	0	1,000.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	151.72	655.16	.00	0	9,344.84	✓
02	151.72	806.88	.00	0	9,193.12	✓
03	151.72	958.60	.00	0	9,041.40	☐
04	151.72	1,110.32	.00	0	8,889.68	☐

5. Navigate back to the Account Log and retrieve the records related to the asset.
   5.1 **Blue**:
      5.1.1 Original Account Log entries. Created the first time the batch job was ran.
   5.2 **Yellow**:
      5.2.1 Entries created by the Change Request. The entries are the same as in #16 above.
   5.3 **Green**
      5.3.1 Entries created by the batch job the second time it was ran.
      5.3.2 Transaction Date is the End Date of the Year and Period being processed, in this case 201102.
      5.3.3 Year and Mon is the Year and Period of Transaction Date.
      5.3.4 Effective Date is the End Date from Accounting Period of the Year and Period of the Forecast.
   5.4 Note the entries with Effective Date of 31/JAN/2011.
      5.4.1 The records in blue were created the first time the batch job was run.
      5.4.2 They were then reversed by the Change Request but were NOT posted because their years were not in the prior year.
      5.4.3 They were then reposted the second time the batch job was executed.
   5.5 Note the entries with Effective Date of 30/NOV/2010.
      5.5.1 The first time the batch job was executed the records in blue were created.
      5.5.2 The DA with +$100 and the DP with $100 reversed the transactions created in 21.5.1.
      5.5.3 The DA with -$151.72 and the DP with +$151.72 were created because they belonged to the Prior Year. The posting was immediate. There was no need to wait for the next period for the records to be reposted by running the batch job.
      5.5.4 It is same for entries with Effective Date of 31/DEC/2010.

5.6 Entries for Year 2010 Periods 09 and 10 purposely not included.


Account Log
Oracle Utilities Work and Asset Management V1.9 (v19-6)
24 March 2011
ORACLE
Find module
Account Log Search Results

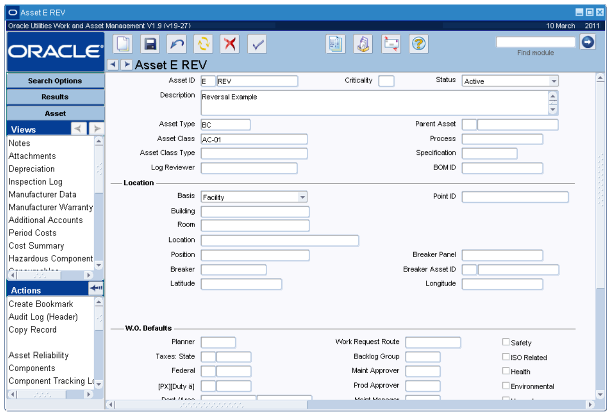
Year	Mon	Effective Date	Trans Date	Post Date	Typ	Trans. Amt	Account	Rec	Asset ID	Expense
2010	12	11/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	AC	3,000.00	ANA-N-A/D-ASSETC-ADJUST-001	E	PTPY	00001
2010	12	30/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2010	12	30/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:1	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2010	12	30/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:1	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2010	12	30/NOV/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	01	30/NOV/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	01	30/NOV/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2010	12	31/DEC/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2010	12	31/DEC/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:1	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2010	12	31/DEC/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:1	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2010	12	31/DEC/2010	31/DEC/2010 00:00:0	10/MAR/2011 12:28:0	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	01	31/DEC/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	01	31/DEC/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	01	31/JAN/2011	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	01	31/JAN/2011	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	02	31/JAN/2011	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	02	31/JAN/2011	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	03	31/JAN/2011	10/MAR/2011 00:00:0	10/MAR/2011 12:28:0	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	03	31/JAN/2011	10/MAR/2011 00:00:0	10/MAR/2011 12:28:0	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002
2011	02	28/FEB/2011	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	PTPY	00003
2011	02	28/FEB/2011	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	PTPY	00002

Reversals

Complete the following steps to reverse costs for an asset.

Create the Asset and Set Depreciation Settings

1. Create an asset. Make sure that the Asset Class and its Depreciation Parameters are set-up.

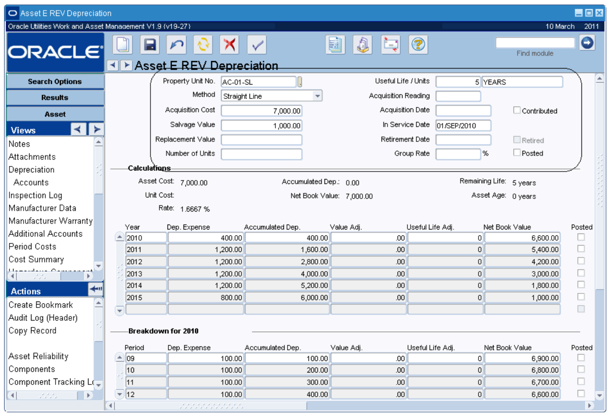


Asset ID: E REV
Criticality:
Status: Active
Description: Reversal Example
Asset Type: BC
Parent Asset:
Asset Class: AC-01
Process:
Asset Class Type:
Specification:
Log Reviewer:
BOM ID:

Location
Basis: Facility
Point ID:
Building:
Room:
Location:
Position:
Breaker Panel:
Breaker:
Breaker Asset ID:
Latitude:
Longitude:

W.O. Defaults
Planner:
Work Request Route:
☐ Safety
Taxes: State
Backlog Group:
☐ ISO Related
Federal
Maint Approver:
☐ Health
[PX][Duty ä]
Prod Approver:
☐ Environmental

2. Navigate to Depreciation view to create a Fixed Asset record. Populate Fixed Asset based on the screen below. Upon Save, the Calculations and Breakdown sections are automatically populated.



Property Unit No.: AC-01-SL
Useful Life / Units: 5 YEARS
Method: Straight Line
Acquisition Reading:
Acquisition Cost: 7,000.00
Acquisition Date:
☐ Contributed
Salvage Value: 1,000.00
In Service Date: 01/SEP/2010
Replacement Value:
Retirement Date:
☐ Retired
Number of Units:
Group Rate: %
☐ Posted

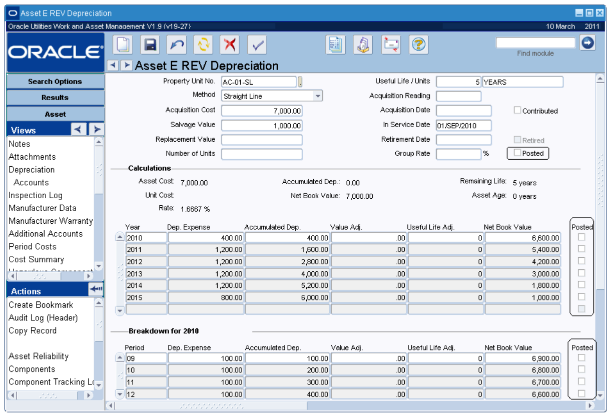
Calculations
Asset Cost: 7,000.00
Accumulated Dep.: 0.00
Remaining Life: 5 years
Unit Cost:
Net Book Value: 7,000.00
Asset Age: 0 years
Rate: 1.6667 %

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	400.00	400.00	.00	0	6,600.00	☐
2011	1,200.00	1,600.00	.00	0	5,400.00	☐
2012	1,200.00	2,800.00	.00	0	4,200.00	☐
2013	1,200.00	4,000.00	.00	0	3,000.00	☐
2014	1,200.00	5,200.00	.00	0	1,800.00	☐
2015	800.00	6,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	☐
10	100.00	200.00	.00	0	6,800.00	☐
11	100.00	300.00	.00	0	6,700.00	☐
12	100.00	400.00	.00	0	6,600.00	☐

3. Take note that the Posted indicators are all unchecked.



Asset E REV Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ☐ Posted

Calculations

Asset Cost:	7,000.00	Accumulated Dep.:	0.00	Remaining Life:	5 years
Unit Cost:		Net Book Value:	7,000.00	Asset Age:	0 years
Rate:	1.6667 %				

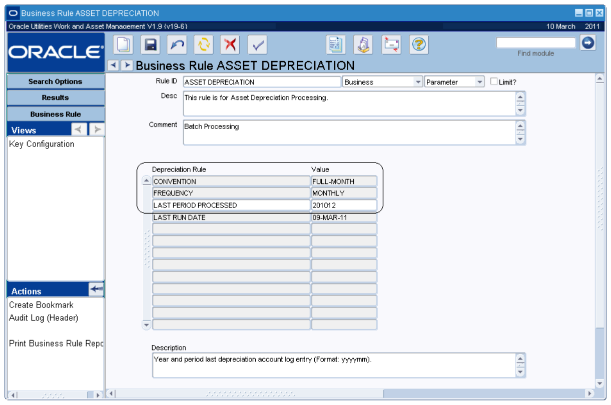
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	400.00	400.00	.00	0	6,600.00	☐
2011	1,200.00	1,600.00	.00	0	5,400.00	☐
2012	1,200.00	2,800.00	.00	0	4,200.00	☐
2013	1,200.00	4,000.00	.00	0	3,000.00	☐
2014	1,200.00	5,200.00	.00	0	1,800.00	☐
2015	800.00	6,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	☐
10	100.00	200.00	.00	0	6,800.00	☐
11	100.00	300.00	.00	0	6,700.00	☐
12	100.00	400.00	.00	0	6,600.00	☐

Modify the Asset Depreciation Business Rule and Run Batch

1. Change the ASSET DEPRECIATION business rule. Set LAST PERIOD PROCESSED, CONVENTION and FREQUENCY based on the screen below.



Business Rule ASSET DEPRECIATION

Rule ID: ASSET DEPRECIATION | Business | Parameter | ☐ Limit?

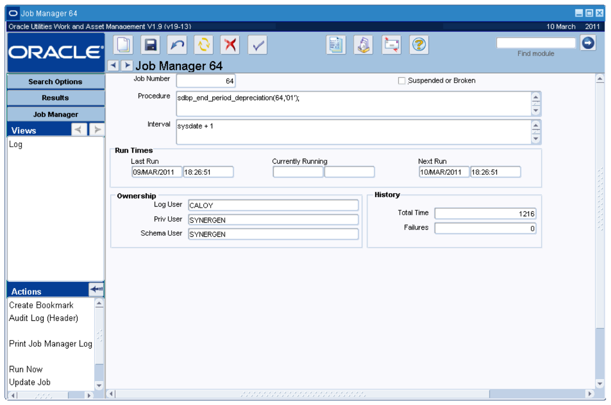
Desc: This rule is for Asset Depreciation Processing.

Comment: Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201012
LAST RUN DATE	09-MAR-11

Description: Year and period last depreciation account log entry (Format: yyyymm).

2. Run the batch job sdbp_end_period_depreciation. The batch job picks-up the LAST PERIOD PROCESSED value and processes the next period, which is Year 2011 Period.



Job Manager 64

Job Number: 64 ☐ Suspended or Broken

Procedure: sdbp_end_period_depreciation(64,'01');

Interval: sysdate + 1

Run Times

Last Run	Currently Running	Next Run
09/MAR/2011 18:26:51		10/MAR/2011 18:26:51

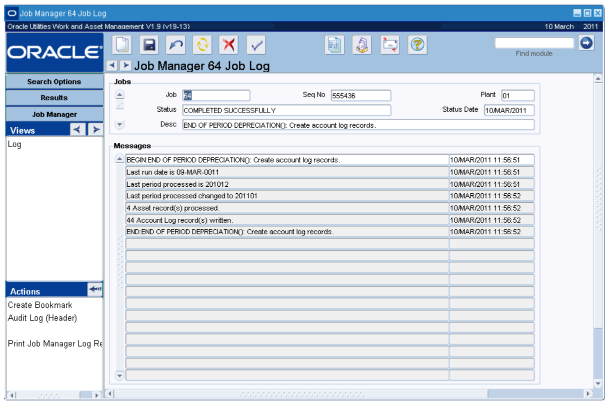
Ownership

Log User	CALOY
Priv User	SYNERGEN
Schema User	SYNERGEN

History

Total Time	1216
Failures	0

3. Make sure that the batch job ran successfully.



Job Manager 64 Job Log

Jobs

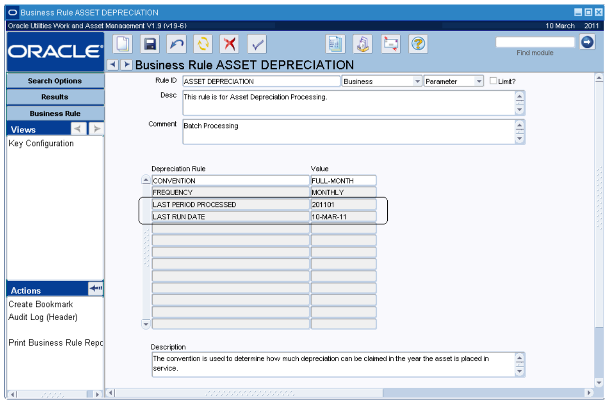
Job	64	Seq No	555436
Plant	01		
Status	COMPLETED SUCCESSFULLY	Status Date	10/MAR/2011
Desc	END OF PERIOD DEPRECIATION(): Create account log records.		

Messages

BEGIN:END OF PERIOD DEPRECIATION(): Create account log records.	10/MAR/2011 11:56:51
Last run date is 09-MAR-0011	10/MAR/2011 11:56:51
Last period processed is 201012	10/MAR/2011 11:56:51
Last period processed changed to 201101	10/MAR/2011 11:56:52
4 Asset record(s) processed.	10/MAR/2011 11:56:52
44 Account Log record(s) written.	10/MAR/2011 11:56:52
END:END OF PERIOD DEPRECIATION(): Create account log records.	10/MAR/2011 11:56:52

Verify that the Asset was Processed Correctly

1. Navigate back to the ASSET DEPRECIATION business rule. Make sure that the LAST PERIOD PROCESSED is incremented by one period and the LAST RUN DATE is updated with Sysdate.



Business Rule ASSET DEPRECIATION

Rule ID: ASSET DEPRECIATION | Business | Parameter | Limit? ☐

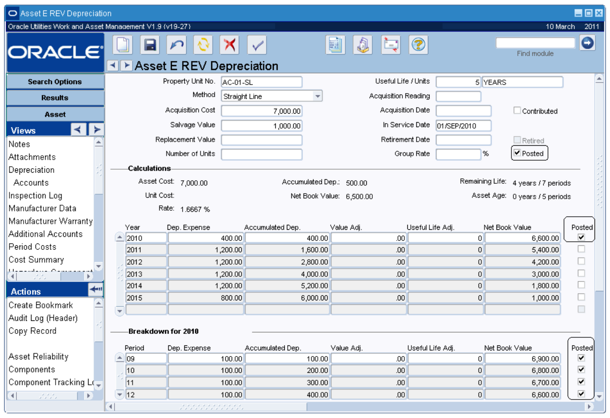
Desc: This rule is for Asset Depreciation Processing.

Comment: Batch Processing

Depreciation Rule	Value
CONVENTION	FULL-MONTH
FREQUENCY	MONTHLY
LAST PERIOD PROCESSED	201101
LAST RUN DATE	10-MAR-11

Description: The convention is used to determine how much depreciation can be claimed in the year the asset is placed in service.

2. Navigate back to the Depreciation view of the Asset module. Note the Posted check boxes. If it is checked, it means that the asset has been processed for the end of period and Account Log entries have been written against the asset.



Asset E REV Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS	
Method	Straight Line	Acquisition Reading		
Acquisition Cost	7,000.00	Acquisition Date		☐ Contributed
Salvage Value	1,000.00	In Service Date	01/SEP/2010	
Replacement Value		Retirement Date		☐ Retired
Number of Units		Group Rate	%	✓ Posted

Calculations

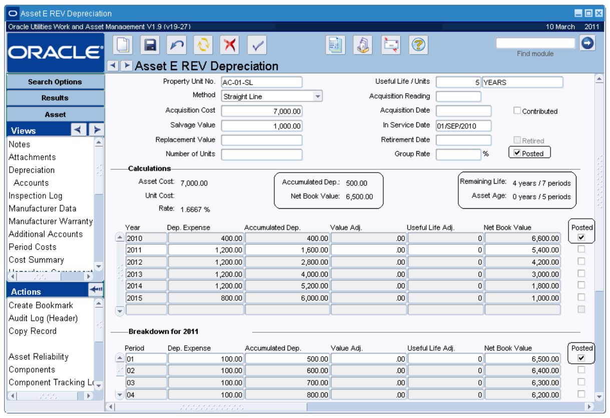
Asset Cost:	7,000.00	Accumulated Dep.:	500.00	Remaining Life:	4 years / 7 periods
Unit Cost:		Net Book Value:	6,500.00	Asset Age:	0 years / 5 periods
Rate:	1.6667 %				

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	400.00	400.00	.00	0	6,600.00	✓
2011	1,200.00	1,600.00	.00	0	5,400.00	☐
2012	1,200.00	2,800.00	.00	0	4,200.00	☐
2013	1,200.00	4,000.00	.00	0	3,000.00	☐
2014	1,200.00	5,200.00	.00	0	1,800.00	☐
2015	800.00	6,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	✓
10	100.00	200.00	.00	0	6,800.00	✓
11	100.00	300.00	.00	0	6,700.00	✓
12	100.00	400.00	.00	0	6,600.00	✓

3. Note that the Year 2011, Period 01 is also posted. The information right below the Calculations heading are updated as well.



Asset E REV Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed
☐ Retired
✓ Posted

Calculations

Asset Cost: 7,000.00
Unit Cost:
Rate: 1.6667 %

Accumulated Dep.: 500.00
Net Book Value: 6,500.00

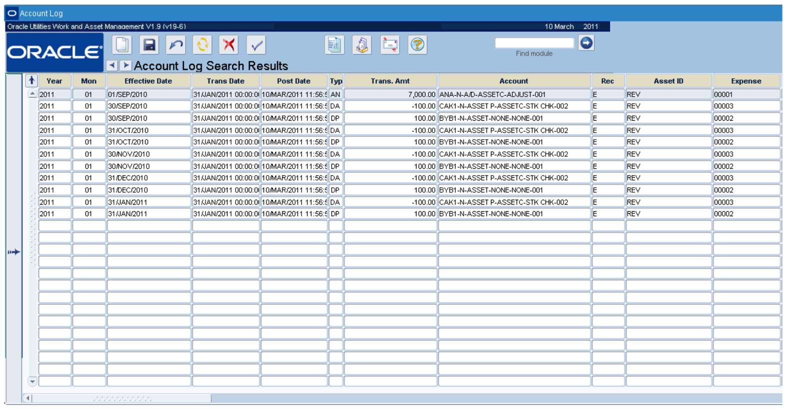
Remaining Life: 4 years / 7 periods
Asset Age: 0 years / 5 periods

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	400.00	400.00	.00	0	6,600.00	✓
2011	1,200.00	1,600.00	.00	0	5,400.00	☐
2012	1,200.00	2,800.00	.00	0	4,200.00	☐
2013	1,200.00	4,000.00	.00	0	3,000.00	☐
2014	1,200.00	5,200.00	.00	0	1,800.00	☐
2015	800.00	6,000.00	.00	0	1,000.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	100.00	500.00	.00	0	6,500.00	✓
02	100.00	600.00	.00	0	6,400.00	☐
03	100.00	700.00	.00	0	6,300.00	☐
04	100.00	800.00	.00	0	6,200.00	☐

4. Navigate to the Account Log and search for the Asset.
The Transaction Date is the End Date of the Year and Period that was processed by the batch job. In this case, the Year and Period that was last processed was 2011 and 01, respectively. The End Date of Year 2011 and Period 01 in the Accounting Period is 31-Jan-2011; hence the Transaction Date is 31/Jan/2011. The Post Date is the Sysdate. The Effective Date is the End Date of the Year and Period of the forecast record that was being posted. The Transaction Type AN (Asset New) was created the first time the asset was posted. The pair of Asset Depreciation/Asset Accumulated Depreciation Transaction Types (DP/DA) with the amount of ?$100 is the Depreciation Expense for the Forecast's Year and Period.

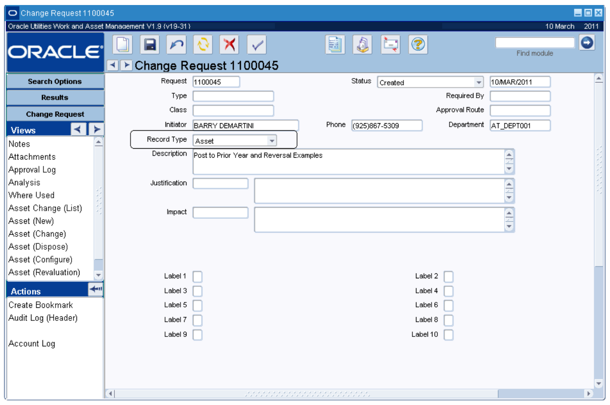


Account Log Search Results

Year	Mon	Effective Date	Trans Date	Post Date	Typ	Trans. Amt	Account	Rec	Asset ID	Expense
2011	01	01/SEP/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	AN	7,000.00	ANA-N-A/D-ASSETC-ADJUST-001	E	REV	00001
2011	01	30/SEP/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	01	30/SEP/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	01	31/OCT/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	01	31/OCT/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	01	30/NOV/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	01	30/NOV/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	01	31/DEC/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	01	31/DEC/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	01	31/JAN/2011	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	01	31/JAN/2011	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002

Create a Change Request

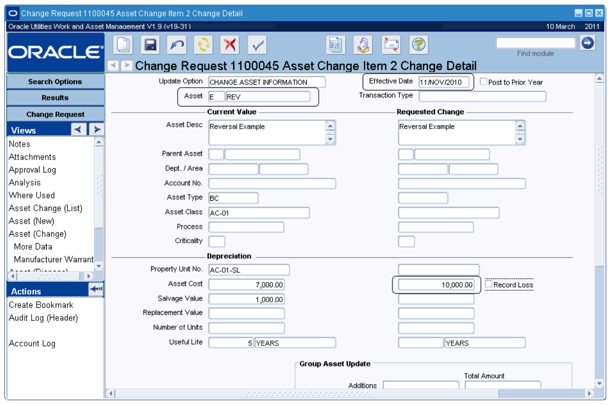
1. Create a Change Request with Asset as Record Type.



Change Request 1100045

Request	1100045	Status	Created
			10/MAR/2011
Initiator	BARRY DEMARTINI	Phone	(925)867-5309
Department	AT_DEPT001		
Record Type	Asset		
Description	Post to Prior Year and Reversal Examples		

2. Navigate to Asset (Change) view and enter the asset that was created in the first step in this procedure above. Create a record based on the screen below. Increase the asset cost from $7,000 to $10,000. Save the record.

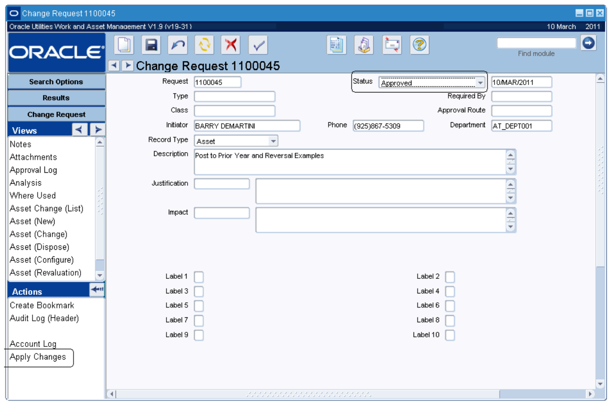


Change Request 1100045 Asset Change Item 2 Change Detail

Update Option: CHANGE ASSET INFORMATION
Effective Date: 11/NOV/2010
Asset: E REV

	Current Value	Requested Change
Asset Desc	Reversal Example	Reversal Example
Asset Type	BC	
Asset Class	AC-01	
Property Unit No.	AC-01-SL	
Asset Cost	7,000.00	10,000.00
Salvage Value	1,000.00	
Useful Life	5 YEARS	YEARS

3. Navigate back to the Change Request header and change the status from Created to Approved. The Apply Changes action appears.

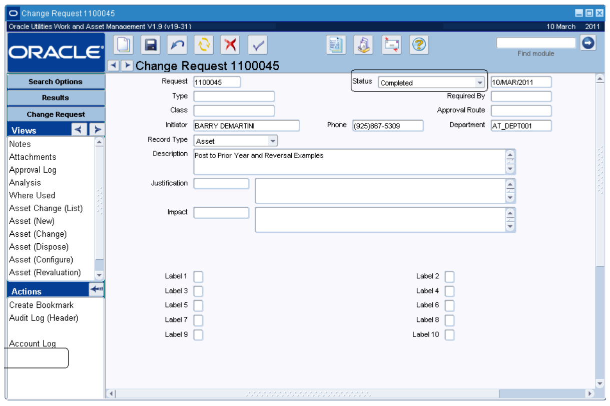


Change Request 1100045

Request	1100045	Status	Approved
			10/MAR/2011
Initiator	BARRY DEMARTINI	Phone	(925)867-5309
Department	AT_DEPT001		
Record Type	Asset		
Description	Post to Prior Year and Reversal Examples		

Actions: Create Bookmark, Audit Log (Header), Account Log, Apply Changes

4. Click on the Apply Changes action. The status changes from Approved to Completed and Apply Changes action disappears.

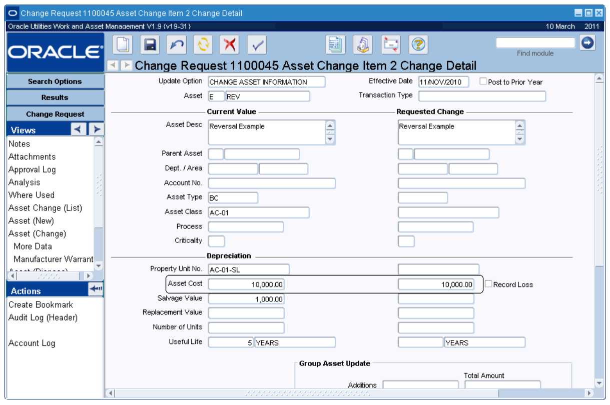


Change Request 1100045
Oracle Utilities Work and Asset Management V1.9 (v19-31) 10 March 2011

Change Request 1100045

Request	1100045	Status	Completed · 10/MAR/2011
Type		Required By	
Class		Approval Route	
Initiator	BARRY DEMARTINI	Phone	(925)867-5309
Department	AT_DEPT001		
Record Type	Asset		
Description	Post to Prior Year and Reversal Examples		
Justification			
Impact			

5. Navigate back to the Asset (Change) screen. The Asset Cost field should now be the same for both Current Value and Requested Change columns.



Change Request 1100045 Asset Change Item 2 Change Detail
Oracle Utilities Work and Asset Management V1.9 (v19-31) 10 March 2011

Change Request 1100045 Asset Change Item 2 Change Detail

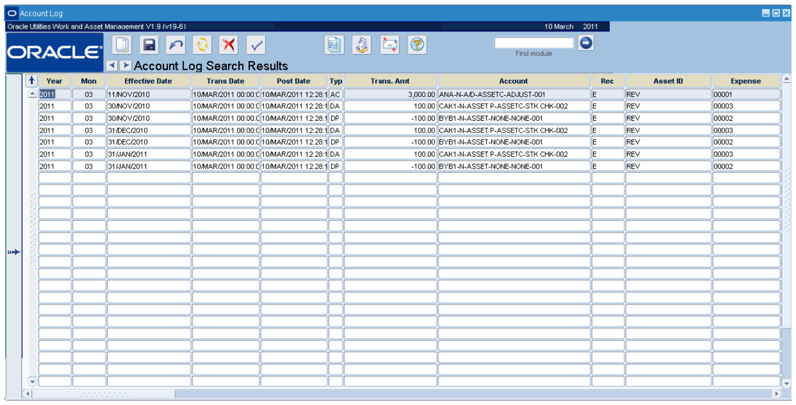
Update Option: CHANGE ASSET INFORMATION
Effective Date: 11/NOV/2010 ☐ Post to Prior Year
Asset: E REV
Transaction Type:

	Current Value	Requested Change
Asset Desc	Reversal Example	Reversal Example
Parent Asset		
Dept. / Area		
Account No.		
Asset Type	BC	
Asset Class	AC-01	
Process		
Criticality		
Depreciation		
Property Unit No.	AC-01-SL	
Asset Cost	10,000.00	10,000.00 ☐ Record Loss
Salvage Value	1,000.00	
Replacement Value		
Number of Units		
Useful Life	5 YEARS	YEARS

Group Asset Update
Total Amount
Additions

6. Navigate back to the Change Request Header and click on the Account Log action. The Transaction Type AC (Asset Cost Change in Value) reflects the increase in the value of the asset or the difference between the new and old asset costs. The Transaction Date is the status date of the Change Request when the status was changed to Completed. The Year and Mon are the Year and Period of the Transaction Date in the Accounting Period table. The Effective Date is the End Date of the Year and Period of the Asset Forecast that was being reversed. In #9 above, when the asset was posted after running the batch job, notice that the DA has a positive value while the DP has the negative value. In the Reversal process, we are backtracking the transactions and amounts that were posted when the batch job was ran, as per #9. Hence, in Reversal, the DA has the positive amount and the DA has the negative value. There are three pairs of DP/DA transactions because there were three

posted forecast records that were reversed, namely; Year 2010 Period 11, Year 2010 Period 12 and Year 2011 Period 01.

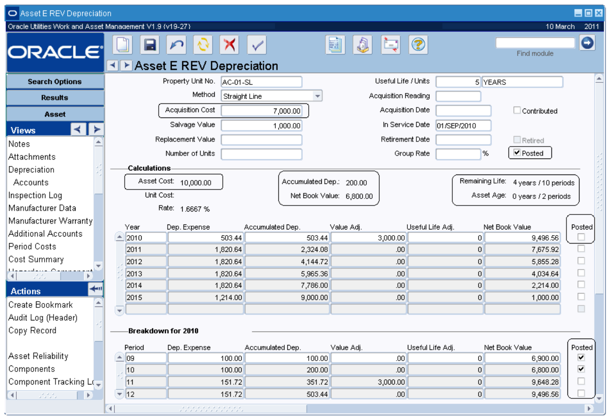

Account Log
Oracle Utilities Work and Asset Management V1.9 (v19-6)
10 March 2011
ORACLE
Find module
Account Log Search Results

Year	Mon	Effective Date	Trans Date	Post Date	Typ	Trans. Amt	Account	Rec	Asset ID	Expense
2011	03	11/NOV/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	AC	3,000.00	ANA-N-A/D-ASSETC-ADJUST-001	E	REV	00001
2011	03	30/NOV/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	03	30/NOV/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	03	31/DEC/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	03	31/DEC/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	03	31/JAN/2011	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	03	31/JAN/2011	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002

Verify Revised Cost Calculations

1. Navigate back to the Asset module and retrieve the asset that was reversed.
2. Navigate to the Depreciation view.

   The Asset Cost reflects the new asset cost. The Acquisition Cost remained at $7,000. The Acquisition Cost do not get updated if the asset has been previously posted. The Accumulated Dep. And Net Book Value display-only fields right below the Calculations heading reflect the Accum Dep. And Net Book Value of the last posted forecast record. Note that the Fixed Asset remains posted. In the Breakdown for 2010, Periods 11 and 12 became unposted because of the Reversal process, hence Year 2010 in the Calculations section got unposted as well. Year 2010, Periods 09 and 10 remained posted. Lastly, the Remaining Life increased and the Asset Age decreased.



Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed
☐ Retired
✓ Posted

Calculations

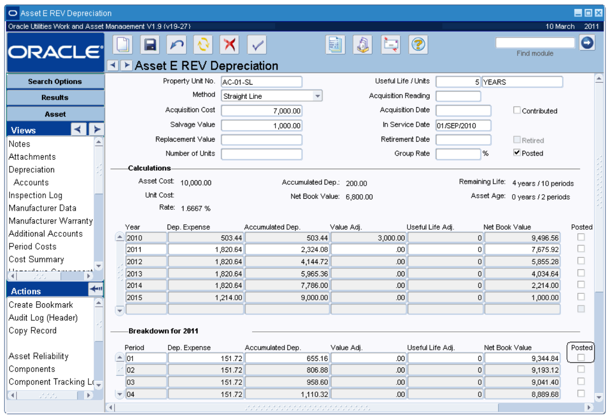
Asset Cost: 10,000.00
Unit Cost:
Rate: 1.6667 %
Accumulated Dep.: 200.00
Net Book Value: 6,800.00
Remaining Life: 4 years / 10 periods
Asset Age: 0 years / 2 periods

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	503.44	503.44	3,000.00	0	9,496.56	☐
2011	1,820.64	2,324.08	.00	0	7,675.92	☐
2012	1,820.64	4,144.72	.00	0	5,855.28	☐
2013	1,820.64	5,965.36	.00	0	4,034.64	☐
2014	1,820.64	7,786.00	.00	0	2,214.00	☐
2015	1,214.00	9,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	✓
10	100.00	200.00	.00	0	6,800.00	✓
11	151.72	351.72	3,000.00	0	9,648.28	☐
12	151.72	503.44	.00	0	9,496.56	☐

3. Highlight Year 2011. Note that Period 01 was unposted as well.



Asset E REV Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ✓ Posted

Calculations

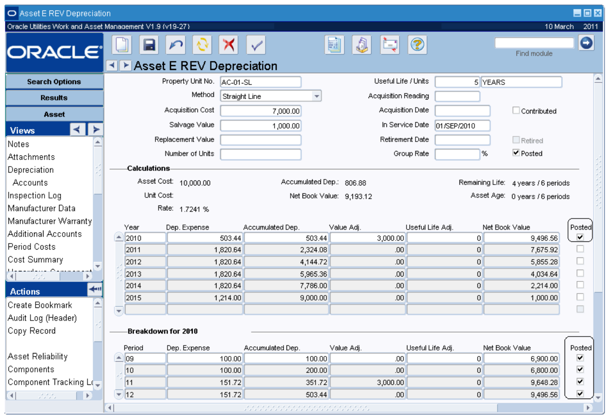
Asset Cost:	10,000.00	Accumulated Dep.:	200.00	Remaining Life:	4 years / 10 periods
Unit Cost:		Net Book Value:	6,800.00	Asset Age:	0 years / 2 periods
Rate:	1.6667 %				

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	503.44	503.44	3,000.00	0	9,496.56	☐
2011	1,820.64	2,324.08	.00	0	7,675.92	☐
2012	1,820.64	4,144.72	.00	0	5,855.28	☐
2013	1,820.64	5,965.36	.00	0	4,034.64	☐
2014	1,820.64	7,786.00	.00	0	2,214.00	☐
2015	1,214.00	9,000.00	.00	0	1,000.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	151.72	655.16	.00	0	9,344.84	☐
02	151.72	806.88	.00	0	9,193.12	☐
03	151.72	958.60	.00	0	9,041.40	☐
04	151.72	1,110.32	.00	0	8,889.68	☐

4. Rerun the batch job and make sure that it ran successfully.
   Make sure that the LAST PERIOD PROCESSED is incremented by one period (i.e. 201102)and the LAST RUN DATE is updated with Sysdate. Navigate back to the Depreciation view of the Asset module. Note that all period for the year 2010 are now posted and the Posted indicator for year 2010 is also checked.



Asset E REV Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ✓ Posted

Calculations

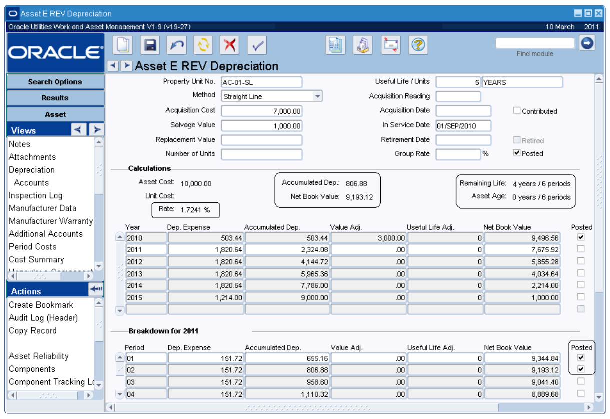
Asset Cost:	10,000.00	Accumulated Dep.:	806.88	Remaining Life:	4 years / 6 periods
Unit Cost:		Net Book Value:	9,193.12	Asset Age:	0 years / 6 periods
Rate:	1.7241 %				

Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	503.44	503.44	3,000.00	0	9,496.56	✓
2011	1,820.64	2,324.08	.00	0	7,675.92	☐
2012	1,820.64	4,144.72	.00	0	5,855.28	☐
2013	1,820.64	5,965.36	.00	0	4,034.64	☐
2014	1,820.64	7,786.00	.00	0	2,214.00	☐
2015	1,214.00	9,000.00	.00	0	1,000.00	☐

Breakdown for 2010

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
09	100.00	100.00	.00	0	6,900.00	✓
10	100.00	200.00	.00	0	6,800.00	✓
11	151.72	351.72	3,000.00	0	9,648.28	✓
12	151.72	503.44	.00	0	9,496.56	✓

5. Highlight the Year 2011.
   The Rate, Accumulated Dep., Net Book Value, Remaining Life and Asset Age were updated. The system 'caught-up' by posting the three reversed forecast records and the next period as well. Periods 01 and 02 for the Year 2011 were posted.



Asset E REV Depreciation

Property Unit No.	AC-01-SL	Useful Life / Units	5 YEARS
Method	Straight Line	Acquisition Reading	
Acquisition Cost	7,000.00	Acquisition Date	
Salvage Value	1,000.00	In Service Date	01/SEP/2010
Replacement Value		Retirement Date	
Number of Units		Group Rate	%

☐ Contributed ☐ Retired ✓ Posted

Calculations

Asset Cost:	10,000.00	Accumulated Dep.:	806.88	Remaining Life:	4 years / 6 periods
Unit Cost:		Net Book Value:	9,193.12	Asset Age:	0 years / 6 periods
Rate:	1.7241 %				

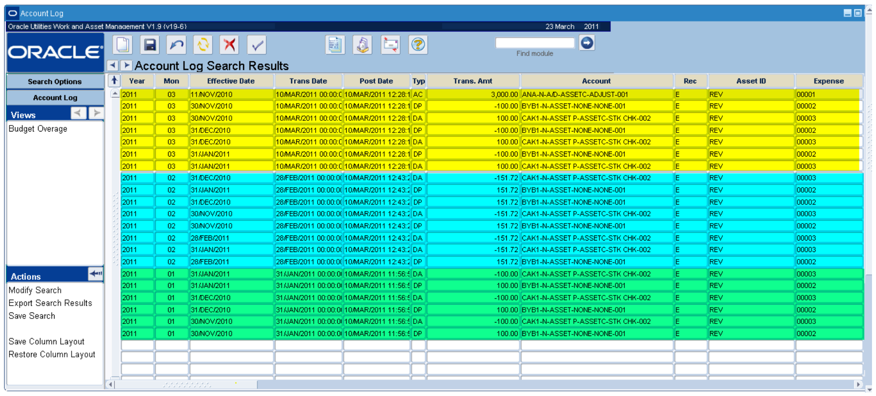
Year	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
2010	503.44	503.44	3,000.00	0	9,496.56	✓
2011	1,820.64	2,324.08	.00	0	7,675.92	☐
2012	1,820.64	4,144.72	.00	0	5,855.28	☐
2013	1,820.64	5,965.36	.00	0	4,034.64	☐
2014	1,820.64	7,786.00	.00	0	2,214.00	☐
2015	1,214.00	9,000.00	.00	0	1,000.00	☐

Breakdown for 2011

Period	Dep. Expense	Accumulated Dep.	Value Adj.	Useful Life Adj.	Net Book Value	Posted
01	151.72	655.16	.00	0	9,344.84	✓
02	151.72	806.88	.00	0	9,193.12	✓
03	151.72	958.60	.00	0	9,041.40	☐
04	151.72	1,110.32	.00	0	8,889.68	☐

6. Navigate back to the Account Log and retrieve the records related to the asset.

The records in yellow are the entries created by the Change Request. The entries in blue are the records created by the batch job the second time it was ran. The log entries in green are the records created the first time the batch job was executed. The AN and log entries for the Year 2010 Periods 09 and 10 were purposely not included.



Year	Mon	Effective Date	Trans Date	Post Date	Typ	Trans. Amt	Account	Rec	Asset ID	Expense
2011	03	11/NOV/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	AC	3,000.00	ANA-N-A/D-ASSETC-ADJUST-001	E	REV	00001
2011	03	30/NOV/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	03	30/NOV/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	03	31/DEC/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	03	31/DEC/2010	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	03	31/JAN/2011	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DP	-100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	03	31/JAN/2011	10/MAR/2011 00:00:0	10/MAR/2011 12:28:1	DA	100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	02	31/DEC/2010	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	02	31/JAN/2011	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	02	31/DEC/2010	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	02	30/NOV/2010	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	02	30/NOV/2010	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	02	28/FEB/2011	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	02	31/JAN/2011	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DA	-151.72	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	02	28/FEB/2011	28/FEB/2011 00:00:0	10/MAR/2011 12:43:2	DP	151.72	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	01	31/JAN/2011	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	01	31/JAN/2011	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	01	31/DEC/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	01	31/DEC/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002
2011	01	30/NOV/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DA	-100.00	CAK1-N-ASSET P-ASSETC-STK CHK-002	E	REV	00003
2011	01	30/NOV/2010	31/JAN/2011 00:00:0	10/MAR/2011 11:56:5	DP	100.00	BYB1-N-ASSET-NONE-NONE-001	E	REV	00002

Appendix C: Creating a Change Request with Custom Depreciation Method

This appendix establishes the guidelines on writing a batch job to create a Change Request from Work Order Construction Asset. Currently, only Straight Line and Group Depreciation are supported. With this enhancement, change requests for assets with the custom depreciation method will be processed and applied.

Custom Batch Job

- Create a batch job to create Change Request.
- Update the Asset and Fixed Asset information, including the depreciation forecast as needed, once the batch job is executed.

Create Custom Batch Job

There are four major steps that the custom batch job should perform:

1. Write batch job log entries. The entries should include number of records processed or errors encountered when it is executed.
2. Create a change request header and line item(s).
3. Apply the change request. This step depends on the business rule Work Order Processing, Apply Change Req Created from WO key. If set to Off, the change request will be set to Created status and it can be manually set to Approved and applied to set it to Completed status. If set to On, the change order will be automatically applied, i.e. set to Completed and the asset updated.
4. Write batch job summary information.

Write Batch Job Log Entries

1. Get the Locale code by calling sdbf_get_locale code and pass the plant. The locale code determines the language that will be used when logging the batch job log messages.
2. Get the Debug Mode by calling sdbg_get_debug_mode. The debug mode determines whether debug messages will be appended to the error messages. If set to On, debug messages will be concatenated to the actual error message, Off will do otherwise. Navigate to Admin> System Configuration> Installation Parameters, Batch Job Debug Mode Object Name to set this parameter.
3. Create a description for the batch job.

4. Call sdbp_ins_job_log_msg to initiate the job log messaging process. Refer to Table 44: Start/End Job Log Messaging for the list of parameters to pass.

Create Change Request Header and Line Item(s)

1. Check auto generate sequence indicator of SA_CHANGE_REQUEST in SA_SEQUENCE_NUMBERS table. If null or set to N:

    1.1 Retrieve SYN-02612 error by calling sdbg_get_message_text. Refer to Table 45: Retrieve Message Text.

    1.2 Log the error message in the job log. Refer to Table 46: Job Log Messaging Detail.

    1.3 Terminate the batch job. Refer to Table 44: Start/End Job Log Messaging.

2. Retrieve all Work Orders (WO) with CLOSED work status.

    2.1 Check if WO contains Construction Assets (SA_WORK_ORDER_FIXED_ASSET) that needs to be valuated. If at least one asset satisfies all the conditions listed below, retrieve SYN-02614 (refer to Table 45: Retrieve Message Text), log the message (refer to Table 46: Job Log Messaging Detail and skip processing the WO.

        2.1.1 Evaluate the Construction Asset.

            2.1.1.1 Change Request and Asset Value are null.

            2.1.1.2 Property Unit No, Asset Record Type and Asset ID are populated.

            2.1.1.3 Action is Create.

        2.1.2 If a Construction Asset satisfies the conditions in 2.1.1.

            2.1.2.1 Retrieve SYN-02614, refer to Table 45: Retrieve Message Text.

            2.1.2.2 Log the retrieved message, refer to Table 46: Job Log Messaging Detail.

            2.1.2.3 Skip processing the WO (i.e. go back to 2) and process the next WO.

    2.2 Retrieve all Construction Assets that satisfy the conditions below:

        2.2.1 The Property Unit No, Asset Record Type and ID are populated.

        2.2.2 Change Request is null.

        2.2.3 Action is Create or Retire.

    2.3 Retrieve the Depreciation Method:

        2.3.1 Get the Depreciation Method from SA_ASSET_FIXED_ASSET (Fixed Asset).

        2.3.2 If 2.3.1 is null, get the Depreciation Method from SA_ASSET_DEP_PARAMETER (Asset Parameter) based on the Asset Class and Property Unit No.

        2.3.3 If the Depreciation Method is SL or GD, i.e. it is not a custom method, skip processing the Construction Asset (go back to 2.2) and process the next asset.

    2.4 Get the next Change Request sequence number by calling sdbp_generate_sequence_no. Refer to Table 47: Generate Change Request Sequence Number for the parameters that need to be passed and received.

    2.5 Create the Change Request header record. Refer to Table 47: Generate Change Request Sequence Number for the columns that need to be populated. See Note 1: Change Request Creation Restriction and Note 2: Batch Job Modification for restrictions.

2.6 If the action is Retire, refer to Table 49: Retire Asset to populate the Change Request Line Item. Retrieve data from SA_ASSET and SA_ASSET_FIXED_ASSET.

2.7 If the action is Create,

2.7.1 Refer to Table 50: Create Asset to populate the Change Request Line Item.

2.7.2 If the asset has no Fixed Asset record, update the Change Request Line item based on the Asset Class and Property Unit Number's Asset Parameters. Refer to Table 51: Update Asset for the fields to update.

2.7.3 Update the depreciation related fields of the Change Request Line item. Refer to Table 52: Update Fixed Asset Data for the fields to update.

2.8 Update Change Request information of the Construction Asset. Refer to Table 53: Update Work Order Construction Asset for the fields to update.

2.9 Update SA_WORK_ORDER and set the pending asset value indicator. Refer to Table 54: Update Work Order Pending Asset Indicator for the fields to update. Refer to Note 3: Work Order Pending Asset Indicator Update Restriction for restriction.

Apply Change Request

1. If at least one Change Request is created:

1.1 Get the Status (key_value2) of the Business Rule WORK ORDER PROCESSING, APPLY CHANGE REQ CREATED FR WO key.

1.2 If Null or Off, stop processing. The Change Request(s) created should remain in Created status.

1.3 If On, call the custom procedure created in Custom Depreciation Phase 2 to set the Change Request to Completed.

Write Batch Job Summary Information

1. Call sdbp_ins_job_log_msg to write the summary information.

1.1 Refer to Table 44: Start/End Job Log Messaging for the list of parameters to pass.

1.2 Retrieve message SYN-02616 (refer to Table 45: Retrieve Message Text). Pass the number of Closed WO processed.

1.3 Retrieve message SYN-02617 (refer to Table 45: Retrieve Message Text). Pass the number of Change Requests created.

1.4 Retrieve message SYN-02618 (refer to Table 45: Retrieve Message Text). Pass the number of Change Requests completed automatically. Should be zero if Business Rule WORK ORDER PROCESSING, APPLY CHANGE REQ CREATED FR WO key is set to OFF.

1.5 Retrieve message SYN-02619 (refer to Table 45: Retrieve Message Text). Pass the number of Change Requests that failed to complete automatically. Should be zero if Business Rule WORK ORDER PROCESSING, APPLY CHANGE REQ CREATED FR WO key is set to OFF.

1.6 End batch job log messaging. Refer to Table 44: Start/End Job Log Messaging for the list of parameters to pass.

Notes

Note 1: Change Request Creation Restriction

If the WO Construction Assets contain only custom depreciation method in Create or Retire action, there should be only one CR header. If there are multiple depreciation methods, i.e. combination of custom method, SL and/or GD, and the action is Create or Retire, then there should be two CR headers. The first one is for assets with custom method. The second one is for SL and/or GD and will be created by sdbp_create_change_request. create_change_req_from_wo

Note 2: Batch Job Modification

Modify Business Rule BATCH JOB CONTROL and add the custom procedure created in this enhancement. Make sure that it is executed before sdbp_create_change_request. create_change_req_from_wo.

Note 3: Work Order Pending Asset Indicator Update Restriction

If the Construction Assets across all the Tasks of the WO has a combination of SL, GD and custom depreciation that has an action of Create and/or Retire, do NOT update the pending asset value indicator. The sdbp_create_change_request. create_change_req_from_wo will perform this step.

Tables

Table 44: Start/End Job Log Messaging

SDBP_INS_JOB_LOG_MSG START/END

COLUMN NAME	COMMENTS
PLANT_IN	
JOB_SEQ_NO_IN	
JOB_IN	Job number when job was created in the Job Manager module
MESSAGE_IN	Job Description
MSG_TYPE_IN	START – to initiate the job log messaging process END – to stop job log messaging process
LOCALE_CODE_IN	

Table 45: Retrieve Message Text

SDBF_GET_MESSAGE_TEXT

COLUMN NAME	COMMENTS
PLANT_IN	
PREFIX_NO_IN	'SYN-' || message number
LOCAL_CODE_IN	
TOKEN_REPLACE_IN	st_msg_token(parameter) Parameter corresponds to the value(s) that will be passed to replace a text in a message. Up to 10 parameters can be passed.

Table 46: Job Log Messaging Detail

SDBP_INS_JOB_LOG_MSG DETAIL

COLUMN NAME	COMMENTS
PLANT_IN	
JOB_SEQ_NO_IN	
JOB_IN	Job number when job was created in the Job Manager module
MESSAGE_IN	Message retrieved from Table 46: Job Log Messaging Detail
MSG_TYPE_IN	DETAIL
LOCALE_CODE_IN	
DEBUG_MODE_IN	
DBMS_ACTIVITY_IN	Dbms_activity, for debugging purposes.

Table 47: Generate Change Request Sequence Number
SDBP_GENERATE_SEQUENCE_NO

COLUMN NAME	COMMENTS
PLANT_IN	
SEQ_NO_NAME_IN	SA_CHANGE_REQUEST
SEQ_NO_LEN_IN	7
SEQ_NO_OUT	Sequence number generated by the system
ERROR_NO	
ERROR_MSG	

Table 48: Create Change Request Header
SA_CHANGE_REQUEST

COLUMN NAME	DATA TYPE	COMMENTS
PLANT	VARCHAR2	
CHANGE_REQUEST_NO	VARCHAR2	
CHANGE_REQUEST_STATUS	VARCHAR2	CREATED
CHANGE_REQUEST_INITIATOR	VARCHAR2	USER
REQUESTOR_PHONE_NO	VARCHAR2	Work Phone No from SA_EMPLOYEE
INITIATOR_DEPARTMENT	VARCHAR2	Department from SA_EMPLOYEE
CHANGE_RECORD_TYPE	VARCHAR2	ASSET
CHANGE_REQUEST_DESC	VARCHAR2	'Construction Assets from Work Order ' || WO Number || WO Description
CREATED_DATE	DATE	SYSDATE
CREATED_BY	VARCHAR2	USER
LAST_UPDATE_DATE	DATE	SYSDATE
LAST_UPDATE_USER	VARCHAR2	USER

Table 49: Retire Asset
SA_CHANGE_REQUEST_ASSET_DATA - RETIRE

COLUMN NAME	DATA TYPE	COMMENTS
ASSET_CHANGE_SEQ_NO	NUMBER	
CHANGE_REQUEST_NO	VARCHAR2	
PLANT	VARCHAR2	
ASSET_RECORD_TYPE	VARCHAR2	
ASSET_ID	VARCHAR2	
UPDATE_OPTION	VARCHAR2	DISPOSE OF ASSET
TRANSACTION_DATE	DATE	
ASSET_DESC	VARCHAR2	
ASSET_TYPE	VARCHAR2	
CRITICALITY	VARCHAR2	
PARENT_ASSET_RECORD_TYPE	VARCHAR2	
PARENT_ASSET_ID	VARCHAR2	
ASSET_CLASS	VARCHAR2	
PROCESS_NO	VARCHAR2	
DEPARTMENT	VARCHAR2	
AREA	VARCHAR2	
ACCOUNT_NO	VARCHAR2	
LOCATION_BASIS	VARCHAR2	
BUILDING	VARCHAR2	
ROOM	VARCHAR2	
LOCATION	VARCHAR2	
POSITION	VARCHAR2	
POINT_ID	VARCHAR2	
NUMBER_PREFIX	VARCHAR2	

STREET_NUMBER	NUMBER	
STREET_NUMBER_CHAR	VARCHAR2	
NUMBER_SUFFIX	VARCHAR2	
STREET_NAME	VARCHAR2	
STREET_DIRECTION	VARCHAR2	
CROSS_STREET	VARCHAR2	
CITY	VARCHAR2	
STATE_PROVINCE	VARCHAR2	
POSTAL_CODE	VARCHAR2	
OFFSET	VARCHAR2	
BREAKER_NO	VARCHAR2	
DIRECTION	VARCHAR2	
SUITE	VARCHAR2	
SPECIFICATION_NO	VARCHAR2	
BOM_ID	VARCHAR2	
FROM_ASSET_RECORD_TYPE	VARCHAR2	
FROM_ASSET_ID	VARCHAR2	
TO_ASSET_RECORD_TYPE	VARCHAR2	
CREATED_DATE	DATE	Default to SYSDATE
CREATED_BY	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 50: Create Asset
SA_CHANGE_REQUEST_ASSET_DATA - CREATE

COLUMN NAME	DATA TYPE	COMMENTS
ASSET_CHANGE_SEQ_NO	NUMBER	
CHANGE_REQUEST_NO	VARCHAR2	
PLANT	VARCHAR2	
ASSET_RECORD_TYPE	VARCHAR2	
ASSET_ID	VARCHAR2	
UPDATE_OPTION	VARCHAR2	CHANGE ASSET INFORMATION
TRANSACTION_DATE	DATE	
SAFETY_CRITICAL_IND	VARCHAR2	
ISO_IND	VARCHAR2	
HEALTH_IND	VARCHAR2	
ENVIRONMENTAL_IND	VARCHAR2	
RUN_TO_FAILURE_IND	VARCHAR2	
LOCATION_BASIS	VARCHAR2	
CREATED_DATE	DATE	Default to SYSDATE
CREATED_BY	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 51: Update Asset
SA_CHANGE_REQUEST_ASSET_DATA

COLUMN NAME	DATA TYPE	COMMENTS
ASSET_CLASS	VARCHAR2	
PROPERTY_UNIT_NO	VARCHAR2	
DEPRECIATION_METHOD	VARCHAR2	
USEFUL_LIFE	NUMBER	
USEFUL_LIFE_UNITS	VARCHAR2	
ASSET_COST_ACCOUNT	VARCHAR2	
ASSET_COST_EXPENSE_CODE	VARCHAR2	
DEPRECIATION_ACCOUNT_NO	VARCHAR2	
DEPRECIATION_EXPENSE_CODE	VARCHAR2	
ACCUM_DEP_ACCOUNT	VARCHAR2	

ACCUM_DEP_EXPENSE_CODE	VARCHAR2	
ASSET_GAIN_LOSS_ACCOUNT	VARCHAR2	
ASSET_GAIN_LOSS_EXPENSE	VARCHAR2	
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 52: Update Fixed Asset Data
SA_CHANGE_REQUEST_ASSET_DATA

COLUMN NAME	DATA TYPE	COMMENTS
ACQUIRED_COST	NUMBER	Asset value from WO Construction Asset
SALVAGE_VALUE	NUMBER	Salvage Price from CU Function. If null, retrieve from Asset Parameter
USEFUL_LIFE_UNITS	VARCHAR2	From Fixed Asset. If null, default to YEARS
LAST_UPDATE_USER	VARCHAR2	Default to USER
LAST_UPDATE_DATE	DATE	Default to SYSDATE

Table 53: Update Work Order Construction Asset

COLUMN NAME	DATA TYPE	COMMENTS
CHANGE_REQUEST_NO	VARCHAR2	
ASSET_CHANGE_SEQ_NO	VARCHAR2	
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Table 54: Update Work Order Pending Asset Indicator

COLUMN NAME	DATA TYPE	COMMENTS
PENDING_ASSET_VALUE_IND	VARCHAR2	Call sdbp_work_order.set_pending_asset_ind and pass the plant and the WO that is being processed
LAST_UPDATE_DATE	DATE	Default to SYSDATE
LAST_UPDATE_USER	VARCHAR2	Default to USER

Chapter 13

Keyboard Shortcuts

This section provides information on customizing Keyboard functions in Oracle Forms.

A text file containing the mapping between Oracle functions and the PC keyboard is stored on the server and controls this customization:

- **file:** <oracle_home_mid-tier>/forms
- **directory:** fmrweb.res

The following example shows mapping TAB key to NEXT_FIELD function in Forms:

```
#    JFN : JMN : URKS : FFN : URFD   (whitespace ignored)
#
#      JFN = Java function number
#      JMN = Java modifiers number
#     URKS = User-readable key sequence (double-quoted)
#      FFN = Forms function number
#     URFD = User-readable function description (double-quoted)
#
#  JAVA FUNCTION NUMBER
#         33 = PageUp
#         34 = PageDown
#         35 = End
#         36 = Home
#         37 = LeftArrow
#         38 = UpArrow
#         39 = RightArrow
#         40 = DownArrow
#    65 - 90 = Ctrl+A thru Ctrl+Z (These will always have the control
#              modifier explicitly included, as well as any other
#              modifiers that might be used.)
#  112 - 123 = F1 thru F12
#          9 = Tab (Ctrl+I, without the control modifier)
#         10 = Return (Ctrl+J, without the control modifier)
#
#  JAVA MODIFIERS NUMBER
#  Equal to the sum of the values for the modifier keys:
#    0 = None
#    1 = Shift
#    2 = Control
#    4 = Meta
#    8 = Alt
#
#  FORMS FUNCTION NUMBER
#  The Forms function numbers match the function numbers found in a
#  typical Forms key binding file.
#
```

```
#  USER-READABLE STRINGS
#  The double-quoted strings appear when users click [Show Keys], and
#  are used for this purpose only. These strings can be translated as
#  needed. Note that the strings do not affect what actually happens
#  when end users press a particular key sequence.
#
9    : 0 : "Tab"            : 1  : "Next Field"
```

For more information, please refer to Oracle Forms documentation which can be obtained from https://support.oracle.com.

Chapter 14
Integration Options

For information on how to use the Oracle Utilities Work and Asset Management integration options, refer to the following:

- Oracle Utilities Work and Asset Management Interfaces Users Guide - details on interfaces
- Oracle Utilities Work and Asset Management Release Notes - a list of related integration products.

Chapter 15
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://[server]:[port]/synergen/services

This opens a listing of all the web services with a link to the associated wsdl.

Web Service Components

Individual components of a 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-respons e	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.

Building 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

At 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:

- Set up the request using simple getter/setters,
- Obtain the service interface from a locator object,
- Make the call to the service, and
- Obtain and manipulate the response with simple getter/setters

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.

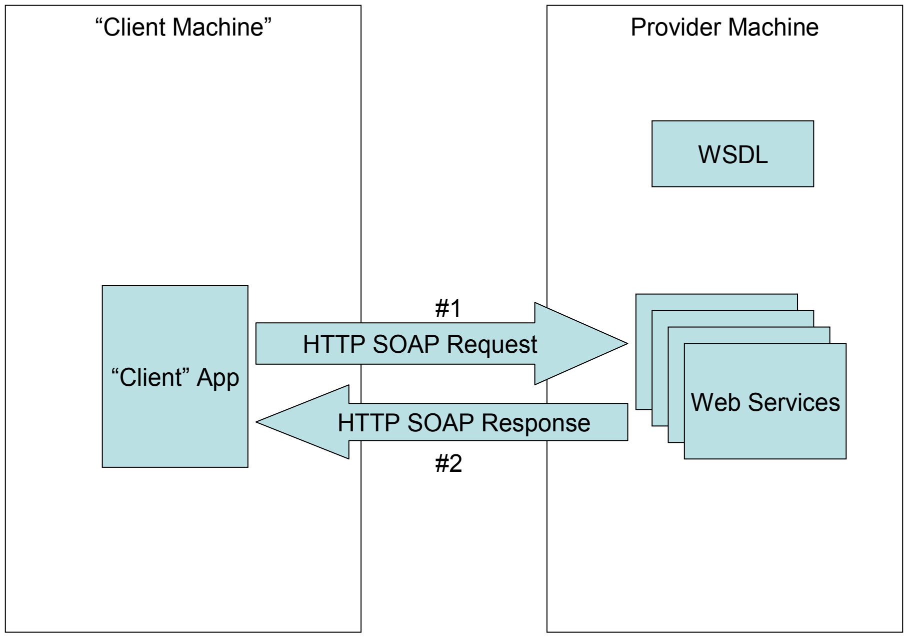
Web Services Provider Interface

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.

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 results
            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)
{
```

Chapter 16
GIS API

For information on how to use the GIS APIs to integrate the GIS features into your local GIS Viewer, see the GIS Overview documentation. This guide provides an overview of how the Oracle Utilities Work and Asset Management application works with a GIS application.

For information on the application programming interface used in these integration between Oracle Utilities Work and Asset Management and GIS system, refer to the GIS Integration Application Programming Interface (API) guide.

Chapter 17
Help Form

The Help Form module in the Administration subsystem maintains the context sensitive links between the Oracle Utilities Work and Asset Management application and the online help file. Context sensitivity is what makes online help open to discussion about the particular screen you are viewing when you select online help. The links are established by assigning a unique online help topic (.htm page) to each form and block combination in Oracle Utilities Work and Asset Management. A stored procedure determines if a user guide, cue card or tutorial topic opens, depending on which option you select from the main Help menu. Oracle Utilities Work and Asset Management technical writers use RoboHelp to maintain the hyperlinked online help topics supplied with the application. However, you can customize online help without using RoboHelp or a similar online help authoring system.

Note: Caution: Before modifying the content or navigation for the online help, you should backup the online help structure as it was delivered with the application. Also make sure that your customized structure is backed up before you install any system upgrades or patches. Only System Administrators and/or Database Administrators should have access to this functionality.

Customizing the Help Form

You can customize the "help map" to call your own customized online help documents for certain form / block combinations, while continuing to use the regular Oracle Utilities Work and Asset Management help file for all other topics.

In addition to customizing the Help Form, you must also create the customized online help documents you want to open. These documents must reside on the same server as the Oracle Utilities Work and Asset Management online help files unless they are stored on an accessible network location on the internet/intranet.

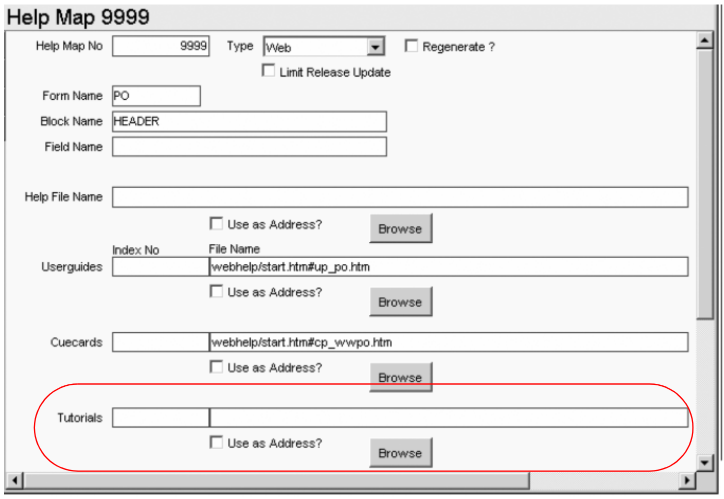
How to Customize the Help Form

1. **Open the Help Form module.**
   The Help Form module is in the Administration subsystem.

2. **Open the appropriate record by identifying the Form and Block where you want to map your content.**
   This will be something like PO / HEADER or WOTASK / ITEM.

   Note: To link from the forms screen that shows when there are no modules open, use the form and block combination: MAIN/HOME.

3. **Locate the Tutorials section and enter the name of the file you want online help to open for this form / block combination.**



Help Map 9999

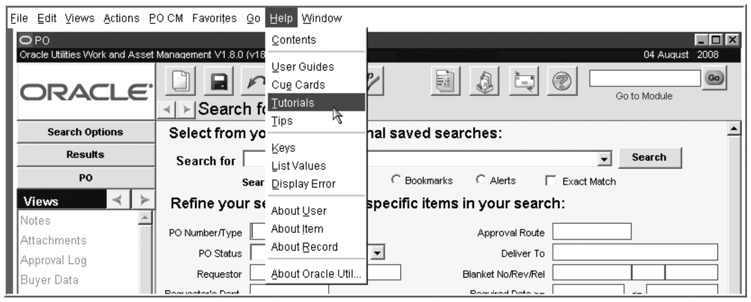
Help Map No: 9999
Type: Web
☐ Regenerate ?
☐ Limit Release Update
Form Name: PO
Block Name: HEADER
Field Name:
Help File Name:
☐ Use as Address? Browse
Userguides — Index No: — File Name: webhelp/start.htm#up_po.htm
☐ Use as Address? Browse
Cuecards — webhelp/start.htm#cp_wwpo.htm
☐ Use as Address? Browse
Tutorials —
☐ Use as Address? Browse

You can use a full url, a network location, or you can store the files in the same location as they are for the Oracle Utilities Work and Asset Management online help system and reference the webhelp folder.

4. **Determine how the online help will open.**
If you place a check in the Use as Address box, the online help is launched in a new browser with the exact URL used as the address. This assumes that you have entered a full working URL. If the box is not checked, the online help is launched in a new browser with the URL appended to the end of the standard application URL. This can be a partial URL using the existing webapp file structure used with the standard application.

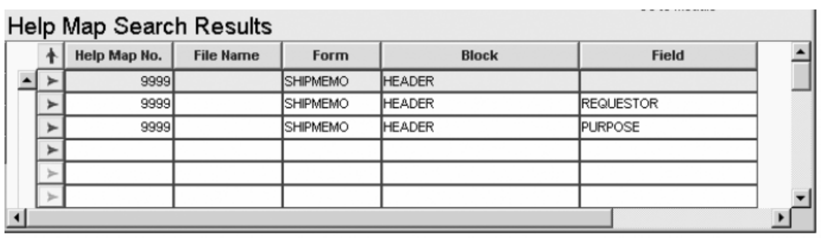
   This function can help you to direct the browser to the correct file location based on the decision you made on how to reference the custom files.

5. **Save the record.**
You can now navigate to the form and block specified, select Help > User Guides (or Cue Cards) to display your custom online help document.



Field Level Help

The Oracle Utilities Work and Asset Management application supports field level online help although online help is typically provided to the block level. To create custom online help at the field level, prepare the appropriate .htm pages and save them with the other WebHelp pages on the server. Then modify the Help Form as described above but specifying the form, block and field the user must be on to call the custom page. In the example below context sensitive Help has been established for the Requestor and Purpose fields on the Shipping Memo header.



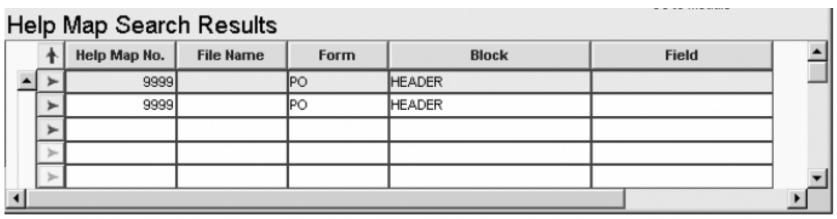
Help Map Search Results

Help Map No.	File Name	Form	Block	Field
9999		SHIPMEMO	HEADER	
9999		SHIPMEMO	HEADER	REQUESTOR
9999		SHIPMEMO	HEADER	PURPOSE

Troubleshooting

If your custom online help text does not open as expected, verify that the filename and path entered on the online help map is correct and that the file exists on the server.

Also, verify that you have not added a duplicate record for the same form/block combination. If you have added a new record, the system may be finding the old record first and ignoring your customized call. To verify this, search for the appropriate Form / Block combination from the Help Map search option window. If you find more than one record with the Help Map Number 9999 for the same Form / Block, delete the ones that are not referencing your custom online help document.



Help Map Search Results

Help Map No.	File Name	Form	Block	Field
9999		PO	HEADER	
9999		PO	HEADER	

Index