

User Defined Module

Version-1.0

9NT1425-ORACLE FCUBSV.UM.11.0.IN.1.0.0.0

[November] [2010]

Oracle Part Number E51713-01

**ORACLE**  
FINANCIAL SERVICES

**ORACLE**

---

## Document Control

Author: Documentation Team	Group: UBPG	
Created on: October 01, 2008	Revision No : Final	
Updated by: Documentation Team	Reviewed by: Development/ esting teams	Approved by: Software Quality Assurance Team
Updated on: November 12, 2010	Reviewed on: November 12, 2010	Approved on: November 12, 2010

## Table of Contents

<b>1. ABOUT THIS MANUAL.....</b>	<b>1-1</b>
1.1 INTRODUCTION.....	1-1
1.2 ORGANIZATION .....	1-1
1.3 CONVENTIONS USED IN THIS MANUAL.....	1-1
1.4 GLOSSARY OF ICONS .....	1-1
1.5 RELATED DOCUMENTS.....	1-2
<b>2. BUILDING A USER DEFINED MODULE.....</b>	<b>2-1</b>
2.1 INTRODUCTION.....	2-1
2.2 BUILDING AN EVENTS CLASS.....	2-2
2.3 CREATING A MODULE USING ORACLE FLEXCUBE CORPORATE.....	2-3
2.4 SELECTING THE ICONS FOR THE PRODUCT DEFINITION SCREEN.....	2-3
2.5 INDICATING THE EVENT DETAILS.....	2-4
2.5.1 <i>Specifying the Event Code</i> .....	2-5
2.5.2 <i>Specifying the method for event triggering</i> .....	2-6
2.5.3 <i>Specifying the validations (Execution Query)</i> .....	2-7
2.5.4 <i>Event Processing</i> .....	2-7
2.5.5 <i>Deriving the Value Date</i> .....	2-7
2.6 DEFINING THE AMOUNT TAGS.....	2-9
2.6.1 <i>Writing derivation rules for amount and currency</i> .....	2-10
2.7 SPECIFYING THE ROLE TYPE .....	2-14
2.8 AUTOMATIC TRIGGERING OF AN USER DEFINED EVENT .....	2-15
2.9 MANUAL TRIGGERING OF EVENTS.....	2-15
2.10 UPLOAD FOR MANUAL EVENT TRIGGERING .....	2-16
2.11 DEFINING ADVICES FOR USER DEFINED EVENTS LINKED TO DEPOSITS PRODUCTS.....	2-17

---

# 1. About this Manual

## 1.1 Introduction

This manual is intended as a guide to help you define User Defined Module (UDM) in FLEXCUBE. Besides this User Manual, you can find answers to specific features and procedures in the Online Help, which can be invoked, by choosing 'Help Contents' from the *Help* Menu of the software. You can further obtain information specific to a particular field by placing the cursor on the relevant field and striking <F1> on the keyboard.

## 1.2 Organization

This manual is organized as follows:










<b>Chapter 1</b>	<i>About this Manual</i> gives information on the intended audience. It also lists the various chapters covered in this User Manual.
<b>Chapter 2</b>	<i>Building a User Defined Module</i> explains how to build a module in Oracle FLEXCUBE either for your own convenience or to suit the requirements of your bank.




## 1.3 Conventions Used in this Manual

Important information is preceded with the  symbol.

## 1.4 Glossary of Icons

This User Manual may refer to all or some of the following icons.

<b>Icons</b>	<b>Function</b>
	New
	Copy
	Save
	Delete
	Unlock
	Print
	Close
	Re-open
	Reverse

Icons	Function
	Template
	Roll-over
	Hold
	Authorize
	Liquidate
	Exit
	Sign-off
	Help
	Add
	Delete

*Refer the Procedures User Manual for further details about the icons.*

## 1.5 **Related Documents**

For further information on procedures discussed in the manual, refer to the Oracle FLEXCUBE manuals on:

- Core Entities
- Core Services
- Common Procedures
- Products
- User Defined Fields

---

## 2. Building a User Defined Module

### 2.1 Introduction

Often while handling large quantity of data in your bank you might want to capture and process information in a particular fashion. Consequently you might want to define your own module either for your own convenience or to suit the requirements of your bank.

The User Defined module of Oracle FLEXCUBE Corporate gives you the opportunity to define your own module whereby you can capture and process information based on your specifications to meet your needs.

You can do this by way of defining the parameters listed below:

- Fields, to capture information specific to individual transaction. You can do this by way of defining custom fields through the User Defined Fields screen of the Core Services module
- Events to be triggered during contract processing
- Amount Tags which are nothing but tags to be attached to amounts which are used to apply charges and taxes
- Accounting Roles for the purpose of passing accounting entries

Oracle FLEXCUBE also offers you the flexibility of selecting the functionality that should form a part of the module being defined. You can choose to have a combination of any of the following features at the Product Definition level:

- Accounts
- Branch
- Charges
- Customer
- Events
- Interest
- MIS
- Preference
- Tax

In addition you also need to specify whether the parameters pertaining to Interest Charges, Commission; and Tax are applicable or not.

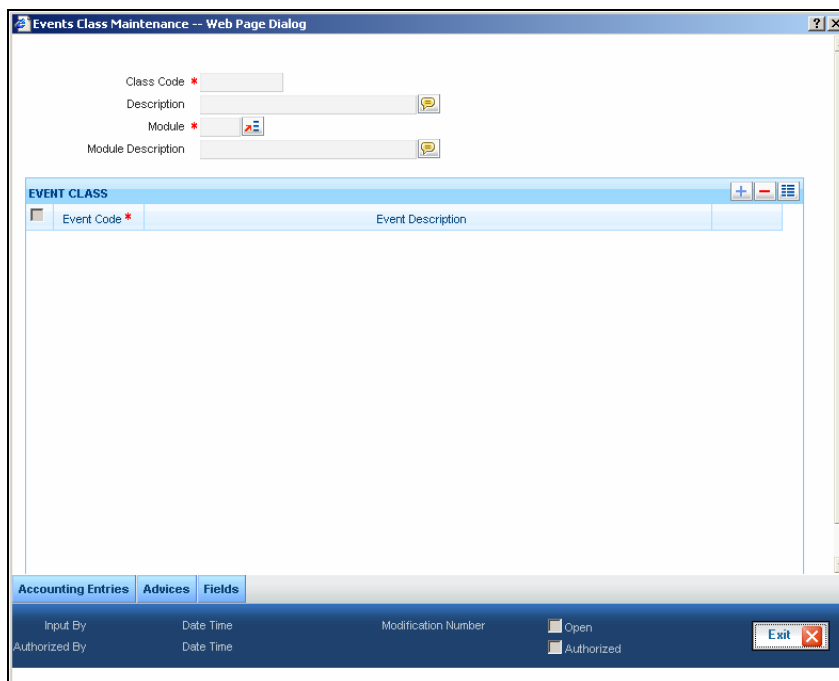
## 2.2 Building an Events class

A class is a specific type of component that you can build with certain attributes. You can build a charge class, for instance, with the attributes of a specific type of charge, such as Charges for provision of services. Similarly, you can build an event class with the attributes of a specific type of events, such as a Booking a Transaction, Collecting Charges, Cancellation and so on.

You can identify an Events Class with a unique Code and Description. When you define an Events Class, you choose, first of all, the set of events that would belong to the class.

Events are, typically, unique to a module.

You can build the events that you would like to include in an Events Class in the 'Events Class Maintenance' screen. You can invoke this screen by typing 'CSDACTCL' in the field at the top right corner of the Application tool bar and clicking on the adjoining arrow button.



For every event constituting the class that you are building, you have to specify the accounting entries that should be passed (if any), and the advices that should be generated. You can do this through the Product Accounting Entries and Advices maintenance screen, which is explained in detail later in the manual.

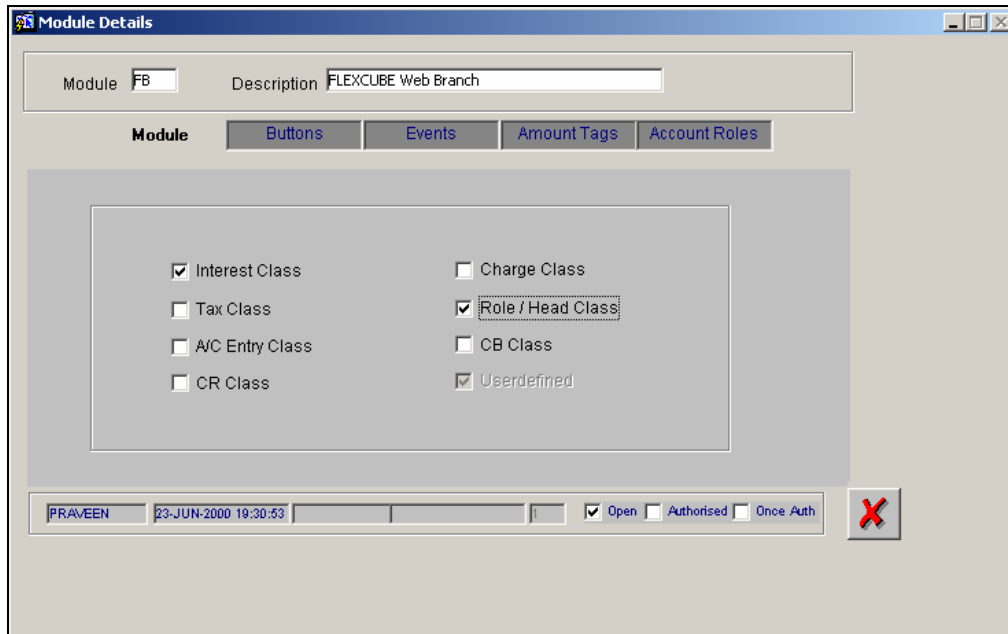


All events need to be triggered manually. The procedure of triggering events manually is explained subsequently in the chapter on Manual Triggering of events.

Since events will have to be triggered manually you will not be allowed to maintain an event Sequence Number as of now.

## 2.3 Creating a Module Using Oracle FLEXCUBE Corporate

The 'Module Details' screen allows you to define a module based on your requirements.

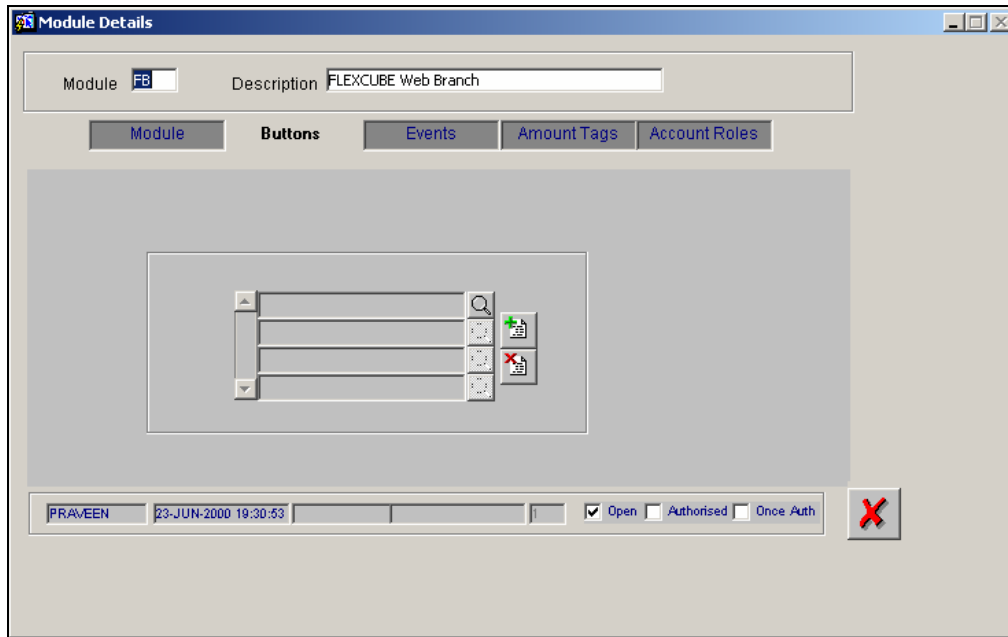


## 2.4 Selecting the icons for the product definition screen

In the Product Definition screen of any module, a horizontal array of buttons is displayed. You need to click on each of the buttons to define the specific attributes of the product. E.g. the 'Interest' button is used to define Interest details, 'tax' button is used to define tax details etc. While creating a new module, you need to indicate the icons that have to be included in the Product Definition screen.

Click 'Buttons' to specify the icons that have to be included in the 'Product Definition' screen of the new module that is being defined.





Depending on your selections in this screen, the Product Definition screen of the new module will have the attributes (represented by the respective icons as in any other Product Definition screen).

In this screen, you can specify the icons for:

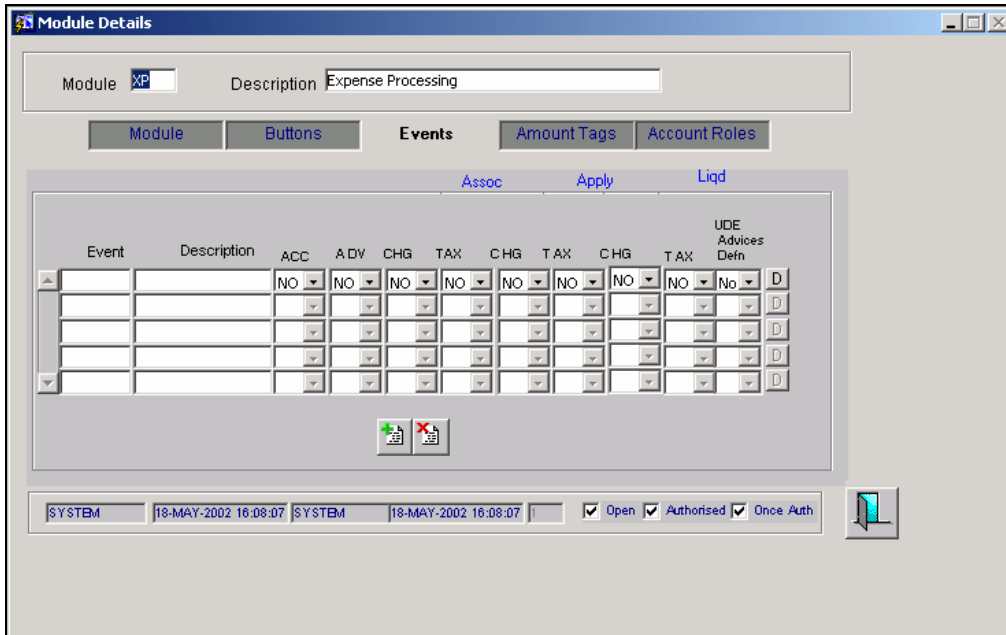
- Defining the interest and charges that you would like to levy on transactions involving the product.
- Maintaining tax details that will be applicable on the transactions involving the product.
- Indicating the type of accounts and the GL/SLs to which the accounting entries have to be posted.
- The preferences specific to a product.
- Maintaining the events that will be generated at different points in the life cycle of contracts involving the product.
- Maintaining a list of allowed or disallowed branches, currencies and customers that can use a product.
- Maintaining the Management Information System (MIS) details.

## 2.5 Indicating the Event Details

A contract that you process in Oracle FLEXCUBE goes through different stages during its life cycle. These stages are defined as Events. Every new module that you maintain has to be associated with a set of events, which will be triggered at appropriate stages during the lifecycle of the contract.

Apart from the factory shipped events, you can create your own events as per the requirements of the bank.

Click 'Events' button in the 'Module Details' screen. You can define events for a new module and also for existing modules in this screen.



User defined events will be linked to a product and is triggered in the life cycle of a contract, which is processed under that product.

### 2.5.1 Specifying the Event Code

The event will be defined by a code. The event code should be unique for a module. Indicate the code through which the event will be identified and also give a brief description of the new event.

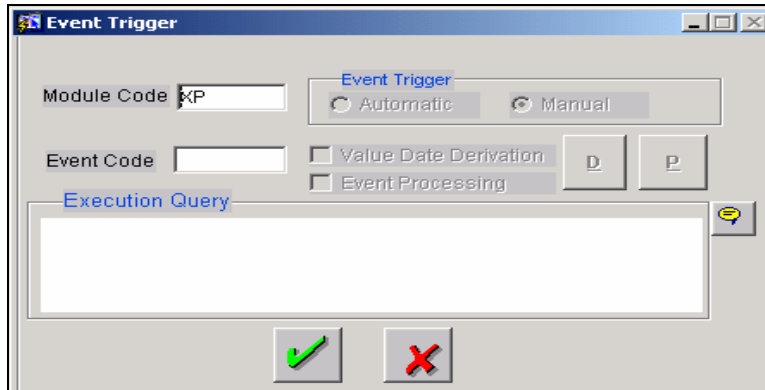
You can also specify the following parameters for the event being defined:

- Whether accounting entries and advices are allowed for this event.
- Whether interest, charge and tax must be computed, but not accrued or levied during this event. This is represented in the respective options under the section 'Assoc'.
- Whether the accounting *entries have to be passed* for interest, tax and charges. This can be indicated in the respective options under the section 'Apply'.
- Whether the interest, charge and tax components must be liquidated when the new event being defined is triggered.
- Whether contract UDE Advices are allowed for this event.

The two charge classes are linked to the LD product definition while accounting entries for the charges is defined in the HDBC and LIQD events respectively.

Click 'D' button to specify the event triggering details.

## 2.5.2 Specifying the method for event triggering



The event, which you are defining, can be triggered either:

- Manually, or
- Automatically

### **UDE for commission on highest outstanding balance of a contract**

For instance, you can set up a user defined event, HDBC, for calculation of commission on the highest outstanding balance of a contract, with the following parameters:

- Event Trigger = Automatic
- Apply Charge = Yes
- Execution Query = Select all contracts which satisfy the following criteria:
  - Product type is L (Transfer)
  - Contract Status is Active
  - The system date is the last working day of the month so that the System processes HDBC events for the contracts only at the end of the month.
- An amount tag UDE\_HOB\_AMT, defined for the Highest Outstanding Balance
- The amount tag derivation rule for UDE\_HOB\_AMT returns the highest outstanding balance for the contract for the month
- Another pre-shipped amount tag MATU\_HOB\_AMT is provided which returns the highest outstanding balance of contract as of maturity, since the previous month end. The charge can be applied based on this amount tag on the LIQD event.
- The event is available at product level

The applicable charges are tracked through two charge classes, Charge on Highest Outstanding Balance and Charge on Maturity Balance. The waiver applicable to specific customers is defined at charge class maintenance level. The event of liquidation for charge for the Charge on Highest Outstanding Balance is HDBC and for the Charge on Maturity Balance, LIQD. These events must be linked to an LD product in the product definition.

The event HDBC computes the charge for Charge on Highest Outstanding Balance, and is executed as part of the UDE batch which executes during the Post-EOTI batch programs; it is run after the execution of the LD batch. The event trigger logic for HDBC ensures that the event only triggers on month ends.

### 2.5.3 Specifying the validations (Execution Query)

You can write an Execution query by which you can instruct the system to retrieve certain contracts to associate with the event that is being defined. The validation rule contains the condition based on which the contracts will be retrieved and associated with the event that is being defined.

### 2.5.4 Event Processing

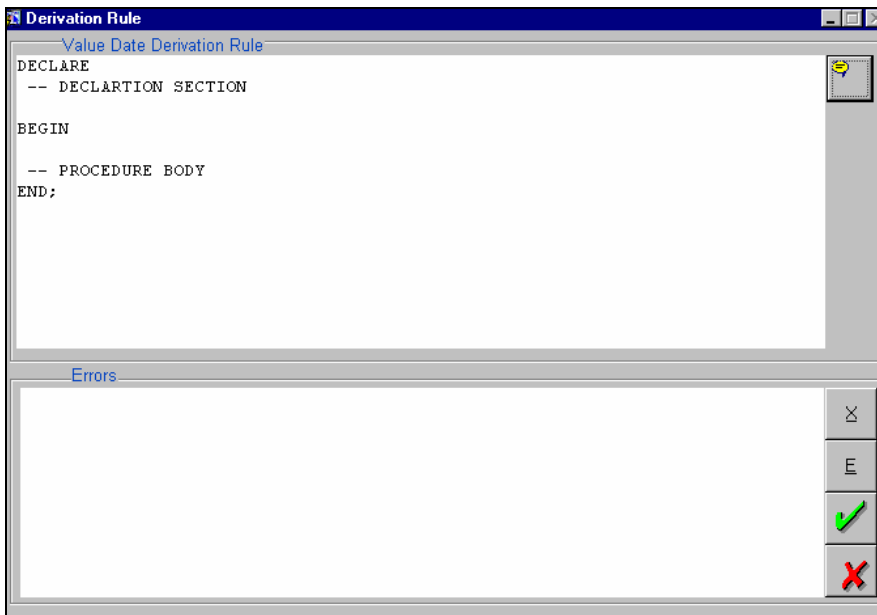
The system derives the values for certain UDF parameters based on the values you give here. During the processing of the event, the values that are mentioned in this place will supercede the ones maintained for the contract.

### 2.5.5 Deriving the Value Date

System derives the value date of the event based on the derivation rule. The accounting entries associated with the event being defined will be posted on the value date.

If the event triggering is set to 'Automatic', you need to write a code to derive the value date of the event. The accounting entries associated with the event being defined will be posted on the value date derived by the system.

When you select automatic mode of event triggering, 'Value Date Derivation' option is automatically checked, as it is mandatory for events triggered automatically. Click 'D' button to write a code to derive the value date for the event.



The value that you derive in these procedures should be assigned to specific tags. You can use the following condition keys:

L_VAL_DT	For assigning value date
P_CONTRACT_REF_NO	Contract reference number
P_VERSION_NO	Version number
P_EVENT_CODE	Event code
P_EVENT_SEQ_NO	Event sequence number



After entering the code, click on the X button to compile the code.

**Example (A)**

Scenario

All Deposit transactions having prepayments will incur a penalty on the outstanding amount.

The rate at which the penalty should be charged is maintained at the UDF defined for the contract. The UDF is called PREPAYRATE.

To meet this requirement we need to define a new event called PREP for the Deposits module. The event will be allowed for accounting entries only.

In the event triggering screen you can either select automatic or manual processing. For manual processing, there is no need for Value Date derivation or Execution query. In case of automatic processing, under value date derivation capture the following procedure to pass the application date as the value date for posting entries.

Begin

L\_VAL\_DT := global.application\_date ;

End;

In the event processing procedure window, (which is invoked by selecting the button) there is no need to pass any values or parameters runtime. So we will write the following:

Begin

Null;

End;

In case of automatic triggering of event an execution/validation query is required. This will identify the contracts for which the event should be executed.

Select contract\_ref\_no from ldtbs\_amount\_due

where component = 'PRINCIPAL'

and due\_date > GLOBAL.APPLICATION\_DATE

and contract\_ref\_no in (select trn\_ref\_no from actbs\_daily\_log where delete\_stat <>'D' and auth\_stat ='A' )  
 having sum(amount\_settled) > 0 group by contract\_ref\_no



The derivation rule for the value date can be written for events, which are triggered manually also. However, if the derivation rule is not written, the value date can be entered at the time of manually triggering an event.

## 2.6 Defining the Amount Tags

You can specify the amount tags that have to be included in a new module/existing module and the method by which the system has to derive the amount and currency for a particular amount tag. Click 'Amount Tags' button in the 'Module Details' screen to specify these details.

You can link two amount tags to a single accounting role at the time of creating a product.

At the time of processing transactions, the currencies of the two amount tags for the debit and credit legs can be different.

### **Example (B)**

Let us assume that you have created a product LDML. The accounting entries for the event LIQD are as follows:

Accounting Role	Amount Tag	Debit/Credit
CUSTOMER	ASSGN-DISCOUNT	Credit
LDML-INTINC	ASSGN-DISCOUNT	Debit
CUSTOMER	ASSGN-PREMIUM	Debit
LDML-INTINC	ASSGN-PREMIUM	Credit

You will notice that the amount tags for the debit and credit legs are different for the event LIQD.

Let us assume that at the time of liquidating a transaction under the product LDML:

- The currency of amount for the debit leg is GBP
- The currency of amount for the credit leg is INR.

However, the local currency of your bank is USD.

Therefore, the currencies of the amount tags linked though linked to the same accounting role are different for the debit and credit legs.

The different currencies of the amount tags for the debit and credit legs when converted to local currency may not match. In such cases, wherein the amounts in different currencies when converted to local currency result in different amounts, you can instruct the system to calculate the amount by taking the average of the two amounts.

With reference to the above example, when the amounts in GBP and INR are converted to USD, the amounts in local currency maybe different because of different exchange rates.

Suppose:

- The amount in GBP when converted to USD (local currency) results in 110 USD, and
- The amount in INR when converted to USD results in 120 USD.

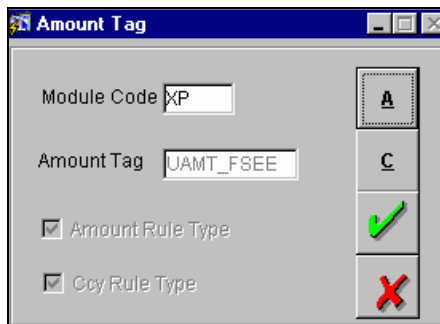
Therefore, at the time of specifying the amount tag ASSGN-DISCOUNT, you can specify the method as 'Average' and amount tag as ASSGN-PREMIUM.

Consequently, system will calculate the average amount of the amount tags ASSGN-DISCOUNT and ASSGN-PREMIUM as 110 USD for the event LIQD.

This calculation can be done for both automatic and manually triggered events.

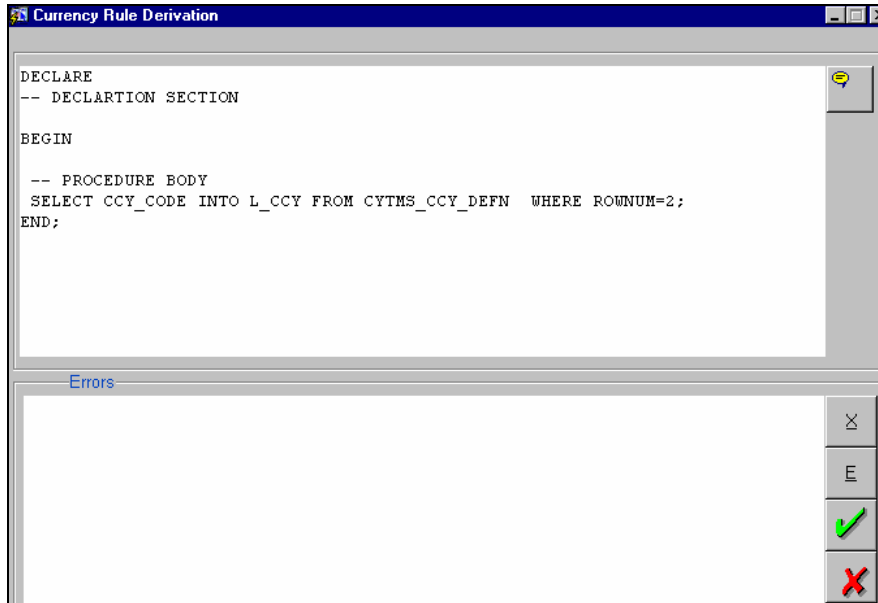
## 2.6.1 Writing derivation rules for amount and currency

Click 'D' button if you want the system to populate the values of amount and currency based on certain conditions.



Currency can either be derived from the existing contract currency or can be maintained as a UDF field in the contract or it can be hard coded in the 'Currency Rule Definition' screen.


In the 'Amount Tag' screen, check against 'Ccy Rule Type' if you want to write a derivation code for currency. Then, click 'C' button to write the derivation code.



```
DECLARE
-- DECLARTION SECTION

BEGIN
-- PROCEDURE BODY
SELECT CCY_CODE INTO L_CCY FROM CYTMS_CCY_DEFN WHERE ROWNUM=2;
END;
```

The screenshot shows a window titled "Currency Rule Derivation" with a text area containing the above PL/SQL code. Below the text area is an "Errors" section which is currently empty. On the right side of the window, there are several control buttons: a close button (X), an edit button (E), a save button (checkmark), and a delete button (X).

 The variable should necessarily be assigned to L\_CCY. The Selecting/Condition keys available are:

- P\_CONTRACT\_REF\_NO
- P\_VERSION\_NO,
- P\_EVENT\_SEQ\_NO
- P\_EVENT\_CODE.

With reference to **Example (A)**:

The **Amount tag** you can use UAMT\_PREP\_CHG

The currency derivation can be taken from cstb\_contract as follows:

Begin

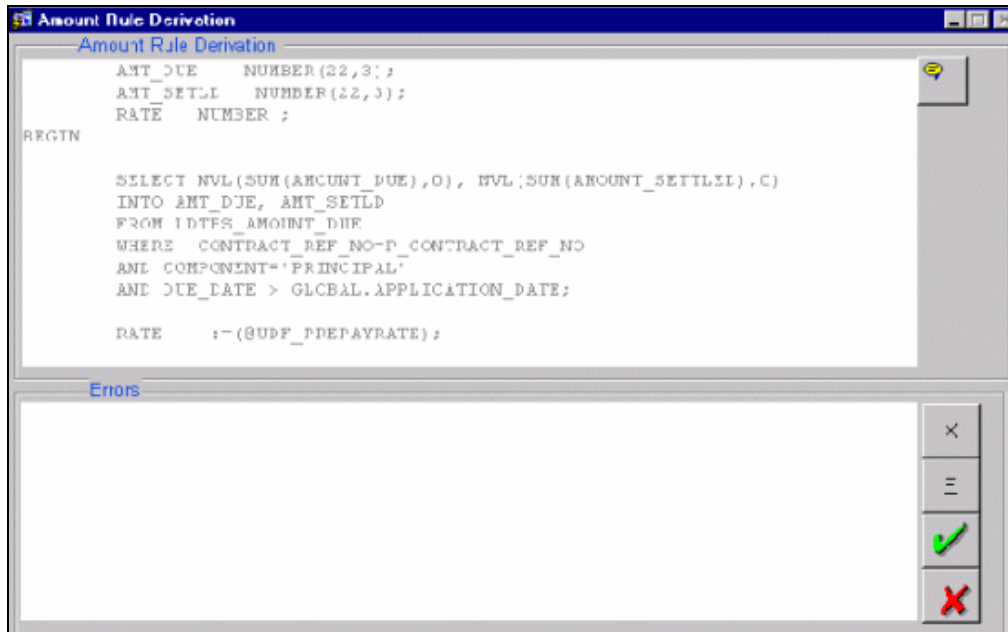
Select contract\_ccy into l\_ccy from cstbs\_contract

where contract\_ref\_no = p\_contract\_ref\_no;

end;

To write a derivation code for amount, check against 'Amount Rule Type' and then click 'A' button.





According to your selections and the derivation code, the derivation rule will return a value (either a currency or an amount).

You can use a user defined field in the currency and amount tag derivation rule. While attaching these amount tags in the 'Product Event Accounting Entries Maintenance' screen at the product level, the user defined fields, which are used in the amount tag and derivation rule will be automatically attached to the product. Enter the values for the user defined fields. Consequently, system will derive the value of the amount/currency from the product.



The variable should necessarily be assigned to L\_AMOUNT. The Selecting/Condition keys available are

- P\_CONTRACT\_REF\_NO
- P\_VERSION\_NO
- P\_EVENT\_SEQ\_NO
- P\_EVENT\_CODE.

With reference to Example A

Here we will derive the outstanding amount of the LD contract that has undergone a prepayment today before the prepayment was done.

**Note:** There are 2 possibilities by which we can arrive at the outstanding amount.

In case of **Manual Triggering** of the event, we use the following and trigger the event before the payment.

```
DECLARE
RATE          NUMBER;
PRIN_OUT      NUMBER;
BEGIN
SELECT PRINCIPAL_OUTSTANDING_BAL
INTO PRIN_OUT
FROM LDTBS_CONTRACT_BALANCE
WHERE CONTRACT_REF_NO=P_CONTRACT_REF_NO;
RATE          :=(@UDF_UD#PREPAYRATE);
L_AMOUNT      := RATE * PRIN_OUT;
END;
```

In case of automatic triggering of the event , we can use the following:

```
DECLARE
AMT_DUE  NUMBER(22,3);
AMT_SETLD  NUMBER(22,3);
RATE  NUMBER ;
BEGIN
SELECT NVL(SUM(AMOUNT_DUE),0), NVL(SUM(AMOUNT_SETTLED),0)
INTO AMT_DUE, AMT_SETLD
FROM LDTBS_AMOUNT_DUE
WHERE CONTRACT_REF_NO=P_CONTRACT_REF_NO
AND COMPONENT='PRINCIPAL'
AND DUE_DATE > GLOBAL.APPLICATION_DATE;
RATE:=(@UDF_PREPAYRATE);
L_AMOUNT:=(AMT_DUE)*RATE/100;
END;
```

Use of UDF

You can use the UDF as a variable in the Derivation in two ways. The options are:

Let us assume that the UDF is called PREPAYRATE

Option I

RATE: =( @UDF\_UD#PREPAYRATE);

This used to be the case in the beginning

Option II

RATE: =( @UDF\_PREPAYRATE);

## 2.7 Specifying the Role Type

Click 'Account Roles' button to specify the details of the accounting roles.

The screenshot shows a software window titled "Module Details". At the top, there are fields for "Module" (DX) and "Description" (Data Extraction). Below these are several tabs: "Module", "Buttons", "Events", "Amount Tags", and "Account Roles". The "Account Roles" tab is active, showing a table with three columns: "Role Code", "Description", and "Role Type". The first row has "ASSET" in the "Role Type" column. To the right of the table are icons for adding and deleting rows. At the bottom of the window, there are fields for "SYSTEM" and "24-NOV-2004 00:00:00" and checkboxes for "Open", "Authorised", and "Once Auth".

In this screen, you can define user defined accounting roles. These accounting roles will be available at the product while mapping the accounting roles to the account heads in the 'Accounting Role to Head Mapping Definition' screen.

## 2.8 Automatic Triggering of an user defined event

You should run the batch program for automatic firing of a user defined event.

At the time of running the batch process, invoke the 'UD Batch Event' screen from the Application Browser.



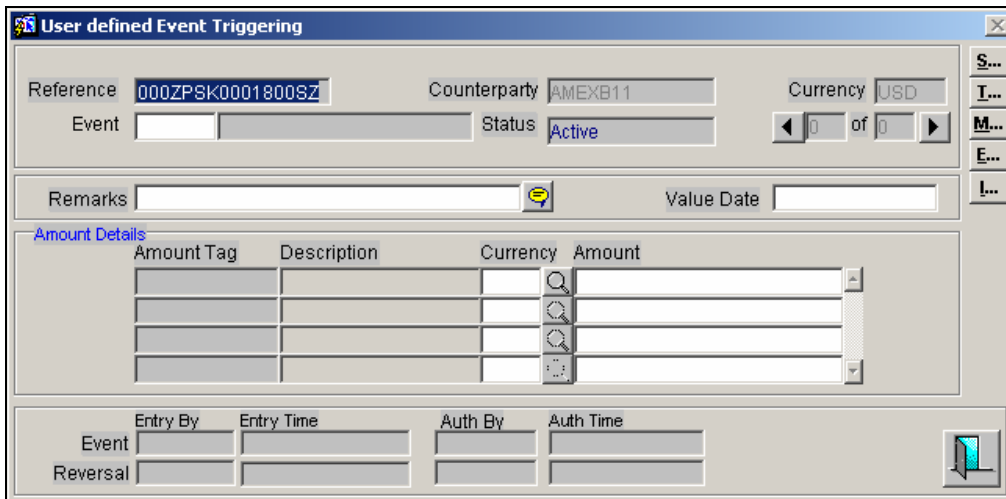
Click 'Ok' button to run the Batch Event Triggering program. The batch program will check if there are any events that have to be fired for active contracts.

These are the steps involved in the automatic triggering of a user defined event:

1. The system will execute the validation code and retrieve the appropriate contracts associated with the event.
2. Derivation rule will be executed to get the value date of the new event. (Accounting entries are posted to the respective GL's on the value date).
3. If there are any derivation rules written for amount and currency, the system will execute the derivation rules to get the amount and currency for a particular amount tag for each entry that is passed.

## 2.9 Manual triggering of Events

Any user defined event, which is set for 'Manual' event triggering can be triggered through the 'User defined Event Triggering' screen. Invoke this screen from the Application Browser.

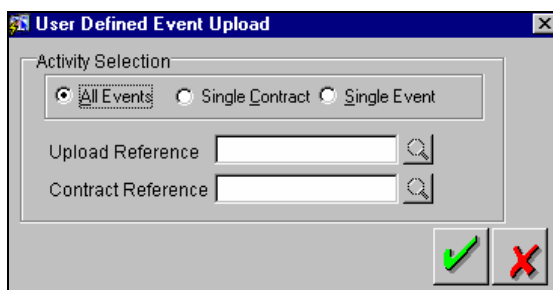
A screenshot of the 'User defined Event Triggering' window. The window has a title bar with a close button. Below the title bar, there are several input fields: 'Reference' (000ZPSK0001800SZ), 'Counterparty' (AMEXB11), 'Currency' (USD), 'Event' (empty), and 'Status' (Active). There are also navigation arrows and '0 of 0' text. Below these fields is a 'Remarks' field and a 'Value Date' field. A section titled 'Amount Details' contains a table with columns: Amount Tag, Description, Currency, and Amount. The table has three rows. At the bottom of the window, there are fields for 'Entry By', 'Entry Time', 'Auth By', and 'Auth Time', along with 'Event' and 'Reversal' checkboxes. A small icon is visible in the bottom right corner.

Navigate to the contract for which you want to trigger an event and click new icon. The user defined events linked to the product under which the contract has been processed will be displayed. Select the event, which has to be triggered. System displays the value date, amount and currency if any derivation logic is written. However, you can change the values (of value date, currency and amount) to suit your requirements.

The associated settlement details, advices, charge and tax details are picked up and the event will be triggered when all the functions are successfully executed.

## 2.10 Upload for Manual Event triggering

Any user defined event, which is set for 'Manual' event triggering, can also be triggered through the 'User Defined Event Upload' screen. Invoke this screen from the Application Browser.



In this method of event triggering, events can be triggered from an upload table, which contains information like the amount, currency and value date of the events, which are set to 'Manual' type of triggering.

The options available for selecting the events for triggering are:

- Single Event – select the appropriate event for triggering from the option list for Upload Reference.
- Single Contract - If this option is selected, all events related to a contract will be triggered. Therefore select a Contract Reference Number. The system will trigger all related events.
- All events – the system will trigger all events of all active transactions.

System will trigger all events of all active transactions.

Select the appropriate option according to your requirement and click 'Ok' button. System will execute the following steps and trigger the event through the upload table:

1. The system will check the method of event triggering for the events that have been selected. Only if it is set to 'Manual', the event will be processed further. Otherwise, it will move on to the next record.
2. System will check whether the upload table has the values of value date, amount and currency. If the table doesn't have the values, the derivation rules will be executed to fetch the values.

3. Next, settlement details and advices are picked up. When all the functions are successfully executed, the event will be triggered from the upload table.

## **2.11 Defining advices for user defined events linked to Deposits products**

The User Defined Event Maintenance screen in Oracle FLEXCUBE allows you to define if the event requires a contract advice to be generated or not. You can generate contract advices for a commitment or a deposit contract when it passes through these stages. You are allowed to generate messages for combination of Branch, Product and each user defined event.

After you specify a list of user defined events applicable for a product, the messages are generated and then handed off at the next stage of the contract upon authorization.

You can generate as many messages for each settlement account if the contract has split settlements or messages can be sent to the counter party of the contract.

### **Linking the Product to the UDE**

You can select the product for which the user defined event details are being maintained. The user defined event details will be validated only for transactions involving the product selected in this field. The description for the product code will be displayed on selection of the product code.

In Oracle FLEXCUBE, when ever you generate messages with type codes 'LD\_CONT\_' + UDE for any of the UDE, the following message tags is displayed for a LD\_CONT\_ advice:

- A unique contract reference number
- The contract amount
- The currency of the contract
- A brief description of the customer including counter-party name, address etc.
- The value date of the contract
- The maturity date of the contract
- The interest rate
- The currency of the contract
- Details of the split settlement

### 2.11.1.1 Maintaining the Receiver Mapping

You can maintain the receiver of the user defined advice as the counter party or the owner of the Settlement account using the 'UDE – Advices - Receiver Mapping' screen invoked from the Application Browser.

Entry By	Date Time	Auth By	Date Time	Authorised	Open
RAMKUMAR	02-DEC-2009 18:19:00			<input type="checkbox"/>	<input checked="" type="checkbox"/>

If receiver mapping is not maintained then the receiver will be defaulted to counterparty of the contract.

#### **Specifying the Branch**

Capture the branch code for which the user defined advice is generated.

#### **Specifying the Message Type**

Select the message type from the option list available.

#### **Indicating the Receiver**

Specify whether the receiver of the user defined advice is a Counterparty or the owner of the Settlement account. Based on this, the advices generated will be sent either to the counter party or to the owner of the settlement account.

---

## 3. User Defined Events for CL

### 3.1 Maintaining User Defined Events

You can add a new event for CL Module using the 'User defined Events' Maintenance Screen. You can invoke this screen by typing 'UDDEVMNT' in the field at the top right corner of the Application tool bar and clicking on the adjoining arrow button.

Specify the following details:

### Module

Specify the module name as CL. Also, you can select the Module from the adjoining option list.

### Event Code

Specify the Event code.

### Event Description

Specify a short description of the event code.

### Accounting Entries Definition

Indicate whether the accounting entries should be defined from the adjoining drop-down list. This list displays the following values:

- Yes
- No

### Advice Definition

Indicate whether the advice needs to be defined from the adjoining drop-down list. This list displays the following values:

- Yes



- No

### **UDE Advices Definition**

Indicate whether the UDE advice needs to be defined from the adjoining drop-down list. This list displays the following values:

- Yes
- No

### **Application**

Specify the following details:

#### **Allow Charge**

Indicate whether the charges should be allowed from the adjoining drop-down list. This list displays the following values:

- Yes
- No

#### **Allow Transaction Tax**

Indicate whether the transaction tax should be allowed from the adjoining drop-down list. This list displays the following values:

- Yes
- No

### **Association**

Specify the following details:

#### **Allow Charge**

Indicate whether the charges should be allowed from the adjoining drop-down list. This list displays the following values:

- Yes
- No

#### **Allow Transaction Tax**

Indicate whether the transaction tax should be allowed from the adjoining drop-down list. This list displays the following values:

- Yes
- No

## **Liquidation**

Specify the following details:

### **Allow Charge**

Indicate whether the charges should be allowed for the liquidation from the adjoining drop-down list. This list displays the following values:

- Yes
- No

### **Allow Transaction Tax**

Indicate whether the transaction tax should be allowed for the liquidation from the adjoining drop-down list. This list displays the following values:

- Yes
- No

Click 'Default' button to invoke the 'Event Trigger' screen. You can configure the triggering parameters for the event in this screen.

The screenshot shows a web page dialog titled "Event Trigger -- Web Page Dialog". The dialog contains the following elements:

- Module Code**: \* CL
- Event Code**: \* UBAC
- Event Trigger**: Radio buttons for  Automatic and  Manual.
- Value Date Derivation
- Event Processing
- Execution Query**: A large empty text area with a small icon to its right.
- Bottom Bar**: Two tabs labeled "Value Date Derivation" and "Event Processing", and two buttons labeled "Ok" and "Cancel".

Specify the following details:

### **Module Code**

Module code is displayed.

### **Event Code**

Event Code is defaulted here.

### **Event Trigger**

Indicate the type of the event trigger that you are maintaining. You can select one of the following options:

- Automatic - To trigger the event automatically during EOD
- Manual - To trigger the event manually from the User defined Event triggering screen

### **Value Date Derivation**

Check this box to indicate that value date derivation is defined for this event.

### **Event Processing**

Check this box to indicate that event processing is defined for this event.

### **Execution Query**

Add the query to select the list of accounts for the execution of the event during EOD. This is mandatory for automatic triggering of event.

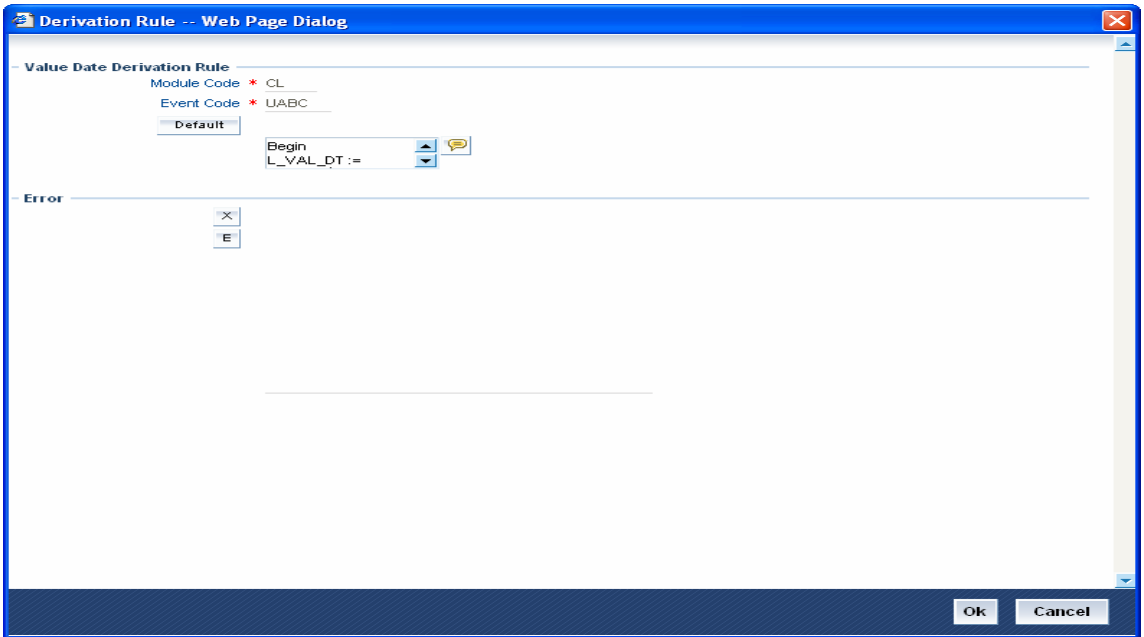
#### **Example**

You can select accounts that have principal overdue schedules:

```
SELECT ACCOUNT_NUMBER FROM cltb_account_schedules
WHERE COMPONENT_NAME='PRINCIPAL'
AND SCHEDULE_DUE_DATE<global.application_date
HAVING SUM (AMOUNT_DUE-AMOUNT_SETTLED)>0 GROUP BY ACCOUNT_NUMBER
```

### **3.1.1 Specifying Value Date Derivation Logic**

Click 'Value Date Derivation' button to invoke the 'Derivation Rule' screen. You can specify the value date derivation logic in this screen.



You have to assign value to L\_VAL\_DT.

**Example**

The following rule derives the application date as the value date:

Begin

```
L_VAL_DT:=global.application_date;
```

End

The Selecting/Condition keys available for modules other than CL are as follows:

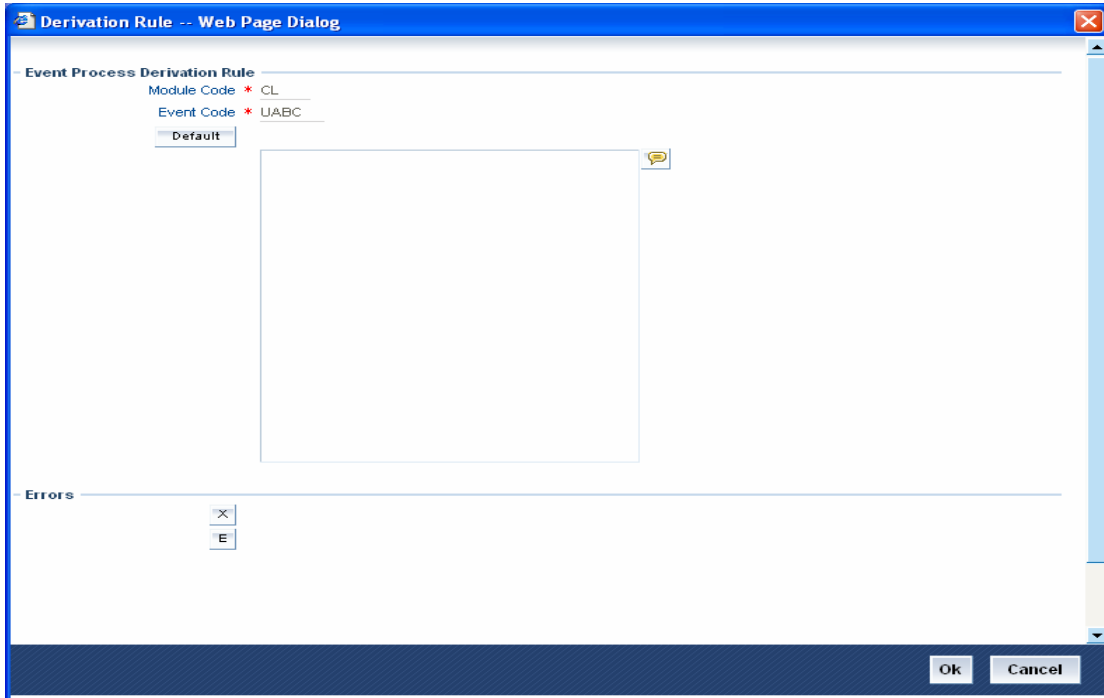
- P\_CONTRACT\_REF\_NO
- P\_VERSION\_NO
- P\_EVENT\_SEQ\_NO
- P\_EVENT\_CODE

The Selecting/Condition keys available for CL module are as follows:

- P\_ACC\_NO
- P\_BRANCH
- P\_EVENT\_SEQ\_NO
- P\_EVENT\_CODE

### 3.1.2 Assigning Values for Event Processing

Click 'Event Processing' button to invoke the 'Derivation Rule' screen. You can assign values which will be fired during processing of the event in this screen.



## 3.2 Maintaining Accounting Role

'Accounting Role and Head Maintenance' screen is used to define the new amount tags and accounting roles for CL module. You can invoke this screen by typing 'UDDRLTAG' in the field at the top right corner of the Application tool bar and clicking the adjoining arrow button.

Amount Tag	Description	Charge	Accounting	Transaction Tax	Local Currency	Local Currency Eq. Tag	Derivation
<input type="checkbox"/> UAMT_COM_AMT	COMM AMOUNT UDEF	No	Yes	No	Average		Derivation

Following details are displayed:

### **Module Code**

Specify the module name.

### **Description**

Specify the description of the screen.

### 3.2.1 Amount Tag tab

You can add amount tags in the amount tag tab. Specify the following details :

### **Amount Tag**

Specify the amount tag.

**Description**

Specify the description of the amount tag.

**Charge**

Indicate if charge is required from the drop-down list. This list displays the following values:

- Yes
- No

**Accounting**

Indicate if accounting is required from the drop-down list. This list displays the following values:

- Yes
- No

**Transaction Tax**

Indicate if transaction tax is required from the drop-down list. This list displays the following values:

- Yes
- No

**Local Currency**

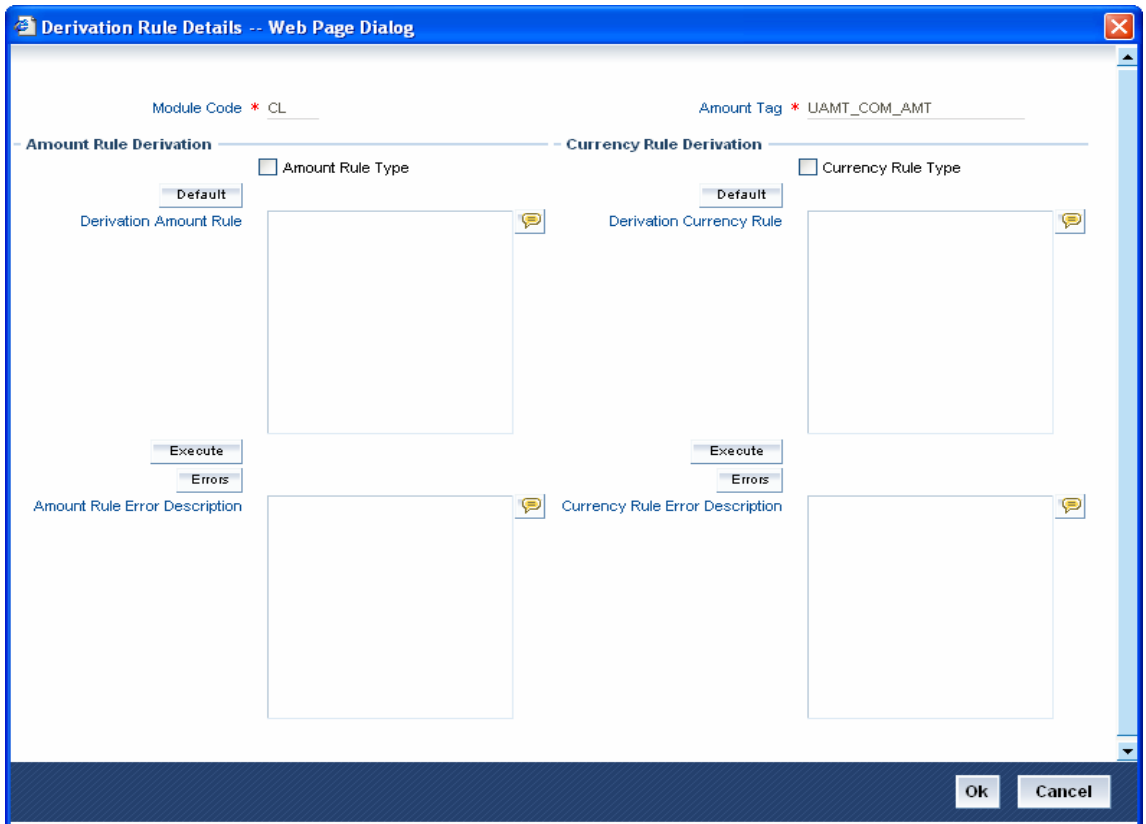
Select the type of the local currency from the drop-down list.

**Local Currency Eq. tag**

Specify the local currency equivalent tag.

**3.2.2 Deriving amount and currency for amount tag**

You can derive the amount and currency for the amount tag in the 'Derivation Rule Details' screen. To invoke this screen click 'Derivation' button in the 'Accounting Role and Head Maintenance' screen.



Currency can either be derived from the existing account currency or can be maintained as an UDF field in the account or it can be hard coded in the Currency Rule Definition screen

### Currency Rule Type

Check this box to indicate that the currency rule should be applied.

Add the logic in Derivation currency rule to derive the currency for the amount tag.

Assign the variable to L\_CCY. The Selecting/Condition keys available are:

- P\_ACC\_NO
- P\_BRANCH,
- P\_EVENT\_SEQ\_NO
- P\_EVENT\_CODE.

### **Example**

In the following derivation logic, currency is derived from the account.

```
BEGIN
```

```
Select currency into l_ccy from cltb_account_master
```



where account\_number = p\_acc\_no and branch\_code=p\_branch;

END;

### **Amount Rule Type**

Check this box to indicate that the amount rule should be applied.

Add the logic in Derivation Amount rule to derive the amount for the amount tag.

Assign the variable to L\_AMOUNT. The Selecting/Condition keys available are:

- P\_ACC\_NO
- P\_BRANCH,
- P\_EVENT\_SEQ\_NO
- P\_EVENT\_CODE

#### **3.2.2.1 Use of UDF**

You can use the account level UDF as a variable in the derivation in two ways. The options are explained in the example given below:

##### **Example**

Let us assume that the UDF is called OSCHGRATE

##### **Option I**

RATE:=(@UDF\_UD#OSCHGRATE);

This used to be the case in the beginning

##### **Option II**

RATE:=(@UDF\_OSCHGRATE);

#### **3.2.2.2 Use of SDE**

You can use the SDE as a variable in the derivation logic for CL as follows:

Lvar:=(@SDE\_VAL#<SDEID>);

Where SDEID – should be a valid SDE ID.

*For the list of SDE, refer the section 'List of SDE Used in Derivation Logic' later in this user manual.*

To get the principal outstanding amount, you can use PRINCIPAL\_OUTSTAND SDE as given below:

```
I_osamt:=(@SDE_VAL#PRINCIPAL_OUTSTAND);
```

The following is derivation logic of amount with SDE and UDF variables.

```
DECLARE
```

```
I_osprin_amt number;
```

```
rate number;
```

```
BEGIN
```

```
L_osprin_amt:=(@SDE_VAL#PRINCIPAL_OUTSTAND);
```

```
rate:=(@UDF_UD#OSCHGRATE);
```

```
I_amount:=rate * I_osprin_amt;
```

```
END;
```

### **3.2.3 Accounting Roles Tab**

Click 'Accounting Roles' Tab to invoke the following screen:

Accounting Role and Head Maintenance -- Web Page Dialog

Module Code \* CL  
 Description Accounting role and amount tags CL

Amount Tag Accounting Roles

Role Code	Description	Role Type
<input type="checkbox"/> UCOMINC	User defined acc role for CL	Income
<input type="checkbox"/> UCOMCUST	User defined acc role for CL	Customer

Input By SHANG      Date Time      Modification Number       Open  
 Authorized By      Date Time       Authorized      **Cancel**

In the Accounting role and head maintenance screen you can add new accounting roles. The accounting roles added are available in the accounting role head mapping at product level. Specify the following details:

**Role Code**

Specify the role code.

**Description**

Specify a description of the role code.

**Role Type**

Select the role type from the drop-down list.

**3.3 Triggering User Defined Events Manually**

You can trigger the user defined event of a CL account manually using the 'CL User Defined Events Triggering' Screen. To invoke this screen type 'UDDCLEVT' in the field at the top right corner of the Application tool bar and click the adjoining arrow button.

Specify the following details:

### Account Number

Specify a valid account number for which the user defined event should be triggered. This adjoining option list displays all valid account numbers maintained in the system. You can choose the appropriate one.

### Event Code

Specify a valid event code. This adjoining option list displays all valid event code maintained in the system. You can choose the appropriate one.

On selecting the account number and event code, if the execution query exists for the user defined event, it gets validated against the selected account number. If the validation fails, appropriate error message is displayed. In such cases, you are not allowed to proceed in triggering the event for the selected account.

### Branch Code

Branch Code gets displayed here.

### **Event Sequence Number**

Specify the sequence number of the event.

### **Value Date**

Value date is defaulted by the system if value date derivation rule exists. However, you can change the value date with any valid date.

### **Payment Remarks**

Here you can specify remarks pertaining to the payment, if any.

Click 'Default' button, to populate the following details:

- Status
- Counterparty
- Currency

### **Amount Tag**

Amount tags get defaulted by the system based on the accounting entries defined for the event at product level.

### **Description**

Description of the amount tag gets displayed here.

### **Ccy**

Currency and amount details are defaulted for each amount tag, if the derivation rule exists for them. However, you can change this value.

### **Amount Paid**

Amount details are defaulted for each amount tag, if the derivation rule exists for them. However, you are allowed to change this value.

### **Settlement Branch**

Specify a settlement branch for each amount tag. This adjoining option list displays all valid settlement branch maintained in the system. You can choose the appropriate one.


### **Settlement Account**

Specify a settlement account for each amount tag. This adjoining option list displays all valid settlement account maintained in the system. You can choose the appropriate one.

After specifying all the above details, click 'Save' to trigger the event.

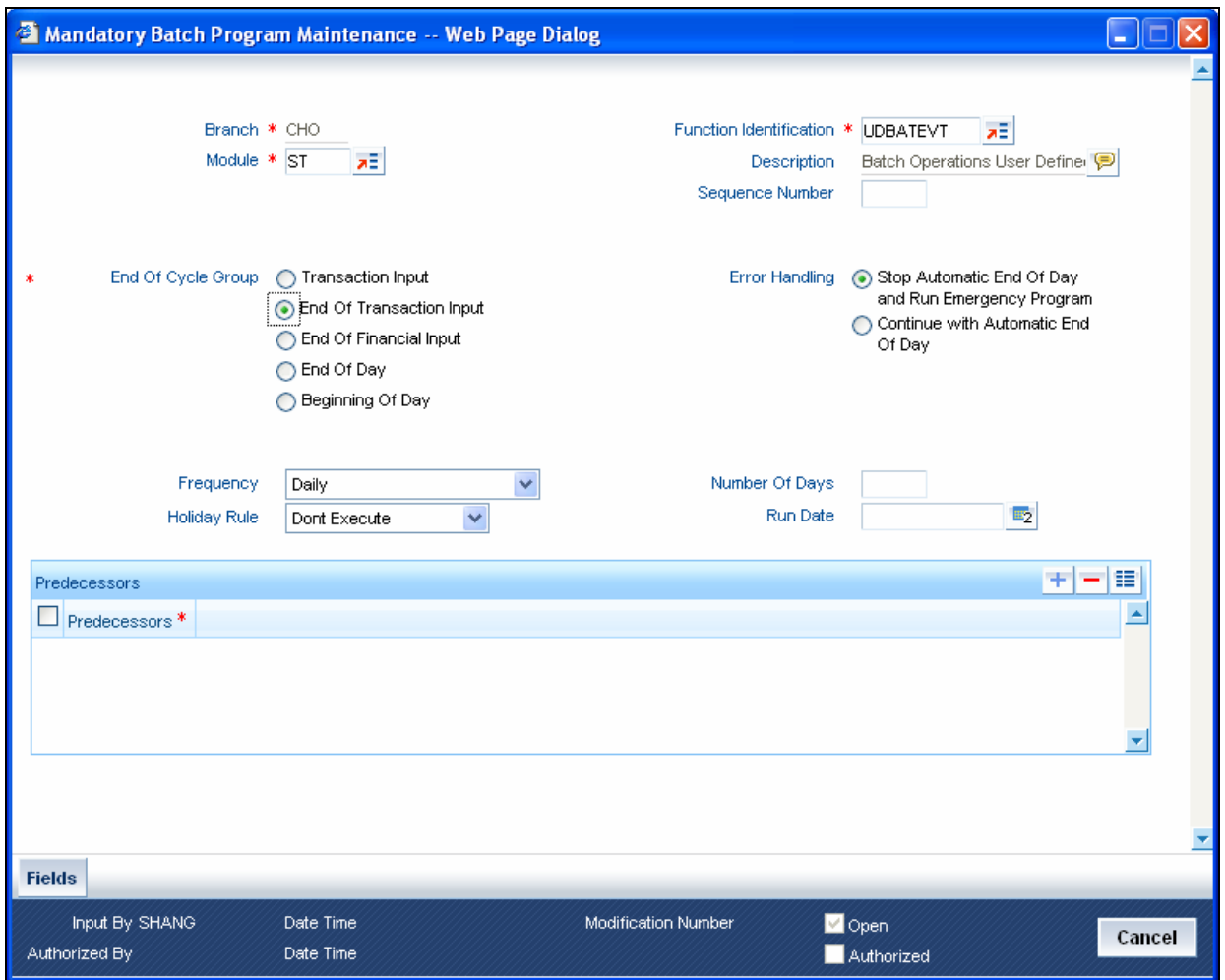
You can carry out the following operations in this screen:

- New - To trigger the user defined event of a CL account
- Query - To query the user defined event triggered for a CL account
- Authorize - To authorize the user defined event triggered for a CL account
- Delete - To delete the user defined event triggered for a CL account before authorization
- Reverse – To reverse the event triggered after authorization

 You can not do any other operation for the account until the event triggered is authorized.

### 3.4 Triggering of User Defined Events Automatically

You can indicate the functions that should be automatically triggered using the 'Mandatory Batch Program Maintenance' screen. To invoke this screen type 'EIDMANPR' in the field at the top right corner of the Application tool bar and click the adjoining arrow button.



**Mandatory Batch Program Maintenance -- Web Page Dialog**

Branch \* CHO  
 Module \* ST

Function Identification \* UDBATEVT  
 Description Batch Operations User Define  
 Sequence Number

\* End Of Cycle Group  
 Transaction Input  
 End Of Transaction Input  
 End Of Financial Input  
 End Of Day  
 Beginning Of Day

Error Handling  
 Stop Automatic End Of Day and Run Emergency Program  
 Continue with Automatic End Of Day

Frequency Daily  
 Holiday Rule Dont Execute

Number Of Days  
 Run Date

Predecessors

Predecessors *

Fields

Input By SHANG      Date Time      Modification Number       Open  
 Authorized By      Date Time       Authorized      **Cancel**

In this screen, you need to maintain the details as follows:

**Function Identification**

Select UDBATEVT batch as the function Identification. This batch triggers the user defined events for CL accounts.

**End Of Cycle Group**

Select End of Transaction Input as the End of Cycle Group.

**Predecessor**

Select UDBATEVT batch as predecessor for the CL batch

*For further details on 'Mandatory Batch Program Maintenance' screen, refer the section 'Defining Functions to be Run Automatically' in the 'Automated End of Cycle Operations' chapter in the 'Automated End of Day' User Manual.*

### 3.4.1 Processing of UDBATEVT Batch

Processing of the UDBATEVT batch takes place as follows:

- System fetches the list of user-defined events of CL for which the event triggering is set as Automatic. The valid accounts associated with the event by the executing the query (Execution query) defined for the event are also fetched.
- Settlement account and branch are defaulted from the settlement details maintained for the CL account. If not maintained the same is defaulted from the settlement details maintained for the PRINCIPAL component.
- Derivation rule is fired to arrive the value date of the event being triggered. Accounting entries are posted to the respective GLs using the derived value date. System derives the amount and currency for each amount tag of the accounting entries defined for the event.
- The user defined event for the account is triggered.
- Charges (if any) defined for the event is handled.
- Accounting entries are posted for the event.
- Advices are processed for the event, if any.
- System supports only CR\_ADV ,DR\_ADV payment message advice for user defined event.

#### 3.4.1.1 List of SDE Used in Derivation Logic

The following are the list of SDEs and its description, which can be used in the derivation logic of Amount, Currency and value date:

SDE_ID	DESCRIPTION
CUSTOMER_CATEGORY	Category of the customer of the loan
CUSTOMER_ID	Customer ID
LOAN_STATUS	STATUS
ACCOUNT_CCY	Loan Account's Currency
DATE_TODAY	Date of Today
DATE_NEXT	Date Of Nextworkday
PAYMENT_VALUE_DATE	Payment Value Date
LAST_PAYMENT_DATE	Last Payment Date for the Customer
DAYS_FOR_NEXT_PMNT	Days for next Payment
AMOUNT_DISBURSED	Current Disbursement Amount



<b>SDE_ID</b>	<b>DESCRIPTION</b>
COMMITMENT_UTILIZED	UTILIZED PART OF COMMITMENT
EMI_AMOUNT	EMI AMOUNT
NET_PRINCIPAL	NON-FUNDED AMOUNT FOR WHICH THE LOAN IS APPLIED
TOTAL_DUE_AMOUNT	Sum of all the Outstanding Balances for the Customer
LAST_PAYMENT_AMOUNT	Last Payment Amount for the Customer
CURRENT_PAYMENT_AMOUNT	Current Amount which is Due
PAID_INTEREST_AMOUNT	Total Interest Amount Paid
INTEREST_ACCRUED_AMOUNT	Sum of all the Accrued Amounts for Interest
AMOUNT_NOT_DISBURSED	Amount Not Disbursed
COMMITMENT_UNUTILIZED	UNUTILIZED PART OF COMMITMENT
TOTAL_PRINCIPAL	AMOUNT FOR WHICH THE LOAN IS APPLIED
TENOR	DURATION FOR WHICH THE LOAN IS APPLIED
LOAN_PRINCIPAL	PRINCIPAL
PRINCIPAL	AMOUNT Disbursed
OVERDUE_MONTHS	OverDue Months
TOTAL_OVERDUE_AMOUNT	Total Overdue amount for all components
INS_TENOR	INSTALLMENT TENOR
NO_OF_RPMNT_SCHS	Number of Non-Moratorium repayment schedules
MORA_OUTSTANDING	Moratorium Outstanding Amount
COMPOUND_VALUE	Amount Compounded
TOTAL_MORA_AMOUNT	Total Moratorium Amount
_OVR_DAYS	<Component> overdue days
_EXPECTED	<Component> expected amount
_OUTSTAND	<Component> outstanding amount
_CCY	<Component> currency

<b>SDE_ID</b>	<b>DESCRIPTION</b>
_OVERDUE	<Component> overdue amount
_FN_NO_REPAY_SCH	<Component> number of schedules for a component if periodic
_NET_PRINCIPAL	<Component> amount financed

### **3.4.1.2 List of Tables Used in Derivation Logic**

The following are the list of CL tables which can be used in the derivation logic:

<b>TABLE NAME</b>	<b>DESCRIPTION</b>
CLTB_ACCOUNT_MASTER	Master table for the loan account
CLTB_ACCOUNT_COMPONENTS	Loan account components
CLTB_ACCOUNT_SCHEDULES	Schedule details of the loan account
CLTB_ACCOUNT_EVENTS_DIARY	Event details of the loan account
CLTB_ACCOUNT_PARTIES	Parties of the loan account



User Defined Module  
[November] [2010]  
Version-1.0

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[www.oracle.com/ financial\\_services/](http://www.oracle.com/financial_services/)

Copyright © [2010] Oracle Financial Services Software Limited. All rights reserved.

No part of this work may be reproduced, stored in a retrieval system, adopted or transmitted in any form or by any means, electronic, mechanical, photographic, graphic, optic recording or otherwise, translated in any language or computer language, without the prior written permission of Oracle Financial Services Software Limited.

Due care has been taken to make this document and accompanying software package as accurate as possible. However, Oracle Financial Services Software Limited makes no representation or warranties with respect to the contents hereof and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this document and the accompanying Software System. Furthermore, Oracle Financial Services Software Limited reserves the right to alter, modify or otherwise change in any manner the content hereof, without obligation of Oracle Financial Services Software Limited to notify any person of such revision or changes.

All company and product names are trademarks of the respective companies with which they are associated.