Oracle® Banking Treasury Management

User Defined Fields User Guide

Release 14.7.4.0.0
G10927-01
June 2024

ORACLE

Oracle Banking Treasury Management User Defined Fields User Guide, Release 14.7.4.0.0

G10927-01

Copyright © 2020, 2024, Oracle and/or its affiliates.

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience v
Acronyms and Abbreviations v
Conventions vi
List of Topics vi
Related Resources vi
Symbols and Icons vii

1 Creating Custom Fields in Oracle Banking Treasury Management

1.1 User Defined Fields Maintenance Screen 1-1
1.2 Scope of the Field 1-3
1.3 Numeric Field 1-4
1.4 Text Field 1-5
1.4.1 Specify Text Fields 1-5
1.4.2 Validation Type — LOV (List of Values) 1-6
1.4.3 Derivation 1-6
1.4.4 Derivation Rule 1-7
1.5 Date Field 1-8
1.5.1 Validations that you can Specify for Date Fields 1-8
Indicate the Validation Type 1-9
1.5.2 Mark the Field as Unique 1-9
1.5.3 Specify the Default Value 1-9
1.5.4 Allow the Modification of Values after Population 1-10
1.5.5 Factory Shipped Fields 1-10
1.5.6 Define Additional Validation Rules 1-10
1.5.7 Validation Rule 1-11
Specify the UD Function Key Mapping Details 1-12
1.5.8 Cube Entity 1-15
Method for Populating the Cube Entity Values 1-15
1.5.9 Enable the Update Allowed Option for a UDF 1-17
1.6 Make a Field Applicable to a Product 1-17
1.7 Creating UDF 1-18
1.7.1 Link UDF to Function ID 1-19

1.8 Population of UDF Values at the Contract Level 1-20
1.9 Upload User Defined Fields 1-21
1.9.1 Contract Upload 1-21
1.9.2 Function Upload 1-22

Index

Preface

This manual is designed to help you to quickly get acquainted with the User Defined Fields module of Oracle Banking Treasury Management.

This preface has the following topics:

- Audience
- Acronyms and Abbreviations
- Conventions
- List of Topics
- Related Resources
- Symbols and Icons

Audience

This guide is intended for Back Office Data Entry Clerk, Back Office Managers/ Officers, Product Managers, End of Day Operators, and Financial Controller users.

Acronyms and Abbreviations

The acronyms and abbreviations are listed in this below table:

Table 1 Acronyms and Abbreviations

Abbreviations or Acronyms	Definition
AEOD	Automated End of Day
AIF	Alternative Investment Fund
CLS	Continuous Linked Settlement
CIF	Customer Information Files
DV	Derivatives
Dr	Debit
EOFI	End of Financial Input
EOD	End of Day
FX	Foreign Exchange
GL	General Ledger
IRS	Internal Revenue Service
ICCB	Interest Commission Charge and Fee
LCY	Local Currency

Table 1 (Cont.) Acronyms and Abbreviations

Abbreviations or Acronyms	Definition
LIBOR	London Interbank Offered Rate
MM	Money Market
OBTR	Oracle Banking Treasury Management
OT	Over the Counter Options
RFR	Risk Free Rates

Conventions

The following text conventions are used in this document:

Table 2 Conventions and Meaning

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
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.

List of Topics

This guide is organized as follows:

Topics	Description
Creating Custom Fields in Oracle Banking Treasury Management	Explains how to create Custom Fields in Oracle Banking Treasury Management explains how UDFs can be defined and used. It details the procedure involved in defining UDFs and associating them with Products, so that the UDF is associated with con- tracts associated with all contracts involving the product. It also explains how to define UDFs and associating them with specific functions in Oracle Banking Treasury Management.

Related Resources

For more information, see these Oracle Banking Treasury Management resources:

- *Core Entities and Services*
- *Common Procedures*
- *The Products User Manual*

Symbols and Icons

Table 3 Symbols

Icons	Function
	Exit
	Add row
	Delete row
	Option List

1

Creating Custom Fields in Oracle Banking Treasury Management

While working with Oracle Banking Treasury Management Corporate Corporate, there are additional fields that you would like to use either for your convenience or to suit the requirements of your bank. Adding to its flexibility, Oracle Banking Treasury Management Corporate now provides you the option to add fields based on your specifications to meet your needs.

- User Defined Fields Maintenance Screen
  This topic describes the systematic instructions to maintain user defined fields.
- Scope of the Field
  This topic describes the scope of the user defined field.
- Numeric Field
  This topic describes how to define the numeric fields.
- Text Field
  This topic describes how to specify the text fields, validation type- List of Values (LOV), derivation, and derivation rules.
- Date Field
  This topic describes the date fields, specify the default value, how to modify the values after population, factory shipped fields, additional validation rules definition, validation rule, cube entity, and how to enable the allowed option for a UDF.
- Make a Field Applicable to a Product
  This topic describes a systematic instructions to make a field applicable to a product.
- Creating UDF
  This topic describes the systematic instruction to create user defined fields.
- Population of UDF Values at the Contract Level
  This topic describes the systematic instructions to populate the UDF values at the contract level.
- Upload User Defined Fields
  This topic description contract upload and function upload of contract.

1.1 User Defined Fields Maintenance Screen

This topic describes the systematic instructions to maintain user defined fields.

Based on your requirement and the nature of the field, you can specify default values and validations for the field. Oracle Lending validates all entries made to the field with the validations you define for a field.

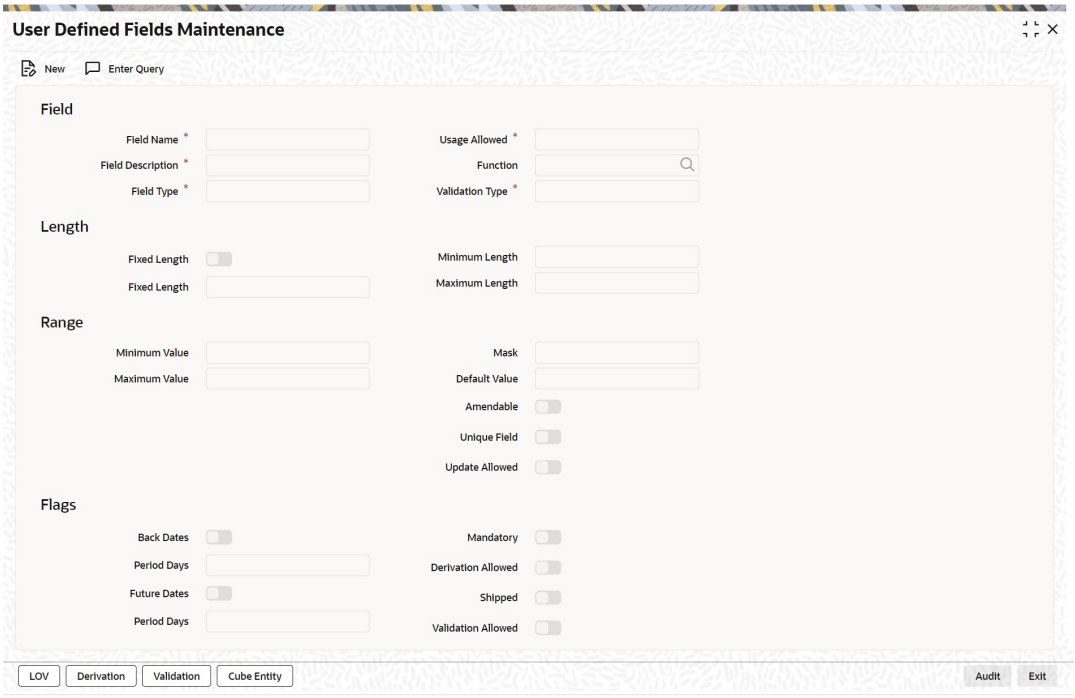
To capture details of User Defined Fields Maintenance screen

1. On the Homepage, Type **UDDUDFMT** and click next arrow.

   The **User Defined Fields Maintenance** screen is displayed.

> **Note:**
>
> The fields which are marked in asterisk red are mandatory fields.

Figure 1-1 User Defined Fields Maintenance



2. A field that you have created becomes operational in Oracle Lending only after it is authorized. A user bearing a different Login ID can authorize a field definition record that you have created.
3. You can enter below basic details in **User Defined Fields Maintenance** screen. For information on fields, refer to the field description table.

Table 1-1 User Defined Field Maintenance Details

Field	Description
Field Name and Description	You can identify a field that you create with a unique identifier, and a brief description. Each field that you define in Oracle Lending should be assigned a unique code. You can briefly describe the field in Description field. The description is for your information only. Itis not be printed on any customer correspondence.

Table 1-1 (Cont.) User Defined Field Maintenance Details

Field	Description
Field Type	The type of field that you can create in Oracle Lending can be of the following formats: • **Number** - Choose this option to create a Numeric field. • **Text** - Choose this option to create a Text field. • **Date** - Choose this option to create a Date field. • **Cube Entity** - Should you need to reuse any of the existing fields of Oracle Lending, to enter additional details you can indicate the Field Type as Cube Entity. The Cube Entity can be any of the existing fields in Oracle Lending like customer, currency, account, and so on.
Marking a Field as Mandatory	You can make entry to a field mandatory. To do so, select the Mandatory option. You are forced to make an entry to the field. Leave it deselected to indicate that the field is not mandatory. **Note:** When a UDF is created, you have to first map it to the corresponding function ID and then provide the validation rule for the UDF.

1.2 Scope of the Field

This topic describes the scope of the user defined field.

While defining a new user defined field, you need to specify whether the new field has to be used at the product or maintenance level. The scope or usage of the field that is being defined can be specified as **Usage Allowed**.

- If the new field that is being defined has to be used at the product level, select usage as ‘Product’.
- If the new field is to be used in any of the maintenance screens (screens related to Core Maintenance like Currency Definition screen, Customer Accounts maintenance screen etc), select usage as ‘Function Id’.

Note:

In Oracle Banking Treasury Management, every screen has a unique function Id. However to differentiate between product and maintenance levels, the scope of a user defined has been classified as Product and Function Id.

If the new field that is being defined is for a maintenance screen (Usage Allowed is ‘Function Id’), you can specify the function Id of the screen in which the new field has to be used. This has to be indicated in the ‘Function Id’ field. Click on the option list positioned next to this field. The function Ids of all the maintenance screens will be displayed. Select the appropriate function Id.

For example, if you want to use the new field in the **Chart Of Accounts - Detailed** screen, select the Function Id **GLDCHACT**, thus allowing the usage of the field in the Chart of accounts screen only.

> **Note:**
>
> If you want the Field to be made available for all the Functions, you will have to leave it blank.

1.3 Numeric Field

This topic describes how to define the numeric fields.

1. To define a numeric field, choose the number option at the Field Type field. You can set up validation rules for a numeric field. The validation types applicable to a numeric field are:
   - Range
   - Length
   - LOV
   - None

   You can indicate your preference at the Validation type field of this screen. Choose None to indicate that no validation should be performed on entries made to this field.

   Table 1-2 Numeric Field - Field Description

Field	Description
Validation Type — Range	You can specify the range validation type only for Numeric fields. In this case the entry to the field should be within a permissible range. On choosing this validation type, you should indicate either the maximum or minimum values or both values allowed for the field. Any valid entry to the field should be within the range that you specify.
Validation Type — Length	For a numeric field you can indicate that the entry should be of a certain length. You have the option to indicate, • a fixed length • the maximum and minimum length for the field Depending on the option you select, indicate the fixed field length or indicate the maximum and minimum length for valid entries to the field.
Validation Type — LOV (List of Values)	Choose LOV to indicate that the entry to this field can be chosen from a predefined list. On choosing this option you can define the items that should be displayed on this list. Click **LOV** button and define the list of values and their description. The items that you define for the list are displayed whenever the field is used in Oracle FLEXCUBE.

2. Click the **Ok** button to close the screen.

1.4 Text Field

This topic describes how to specify the text fields, validation type- List of Values (LOV), derivation, and derivation rules.

This topic contains the following sub-topics:

- Specify Text Fields
  This topic describes how to specify the text fields.
- Validation Type — LOV (List of Values)
  This topic describes the how to define the validation types like List of Values.
- Derivation
  This topic describes how to populates the values of a field.
- Derivation Rule
  This topic describes how to derive the rules.

1.4.1 Specify Text Fields

This topic describes how to specify the text fields.

To define a text field, choose the text option at the Field Type field. A text field can contain alphabets of the English language or a combination of alphabets and numeric values.

1. You can specify validation rules for a text field. The validation types applicable to a text field include:
   - Length
   - Mask
   - LOV
   - None

   You can indicate your preference at the Validation type field of this screen. Choose None to indicate that no validation should be performed on entries made to this field.

Table 1-3 Text Fields - Field Description

Field	Description
Validation Type — Length	For a text field you can indicate that a valid entry to the field should be of a certain length. You have the option to indicate, • A fixed length • The maximum and minimum length for the field Depending on the option that you select, indicate the fixed field length or indicate the maximum and minimum length of entries made to the field.
Validation Type — Mask	To indicate a field as a masked field, choose Mask as the field type. This option allows you to define a broad field structure to which all entries to the field should conform. The mask structure can consist solely of ‘a’ or ‘n’ or a combination of these. An ‘a’ would indicate an alphabet of the English language and ‘n’ a numeric value. All entries made to the field will be validated against the format that you specify for the mask.

2. Click the **Ok** button to close the screen.

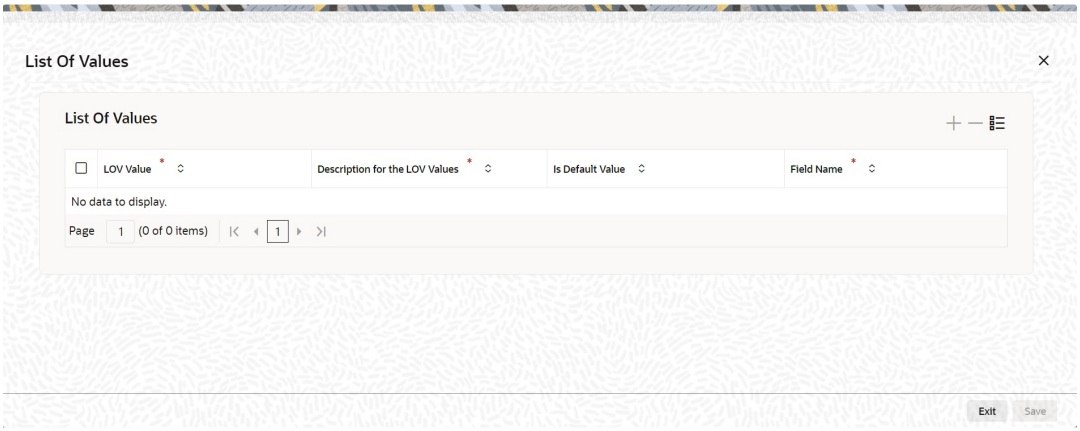
1.4.2 Validation Type — LOV (List of Values)

This topic describes the how to define the validation types like List of Values.

1. Choose LOV to indicate that the entry to this field can be chosen from a predefined list. On choosing this option you can define the items to be displayed on this list.
2. Click **LOV** button and define the list of values and their description.

   The items that you define for the list are displayed whenever the field is used in Oracle Banking Treasury Management.

Figure 1-2 List of Values



3. Click the **Ok** button to close the screen.

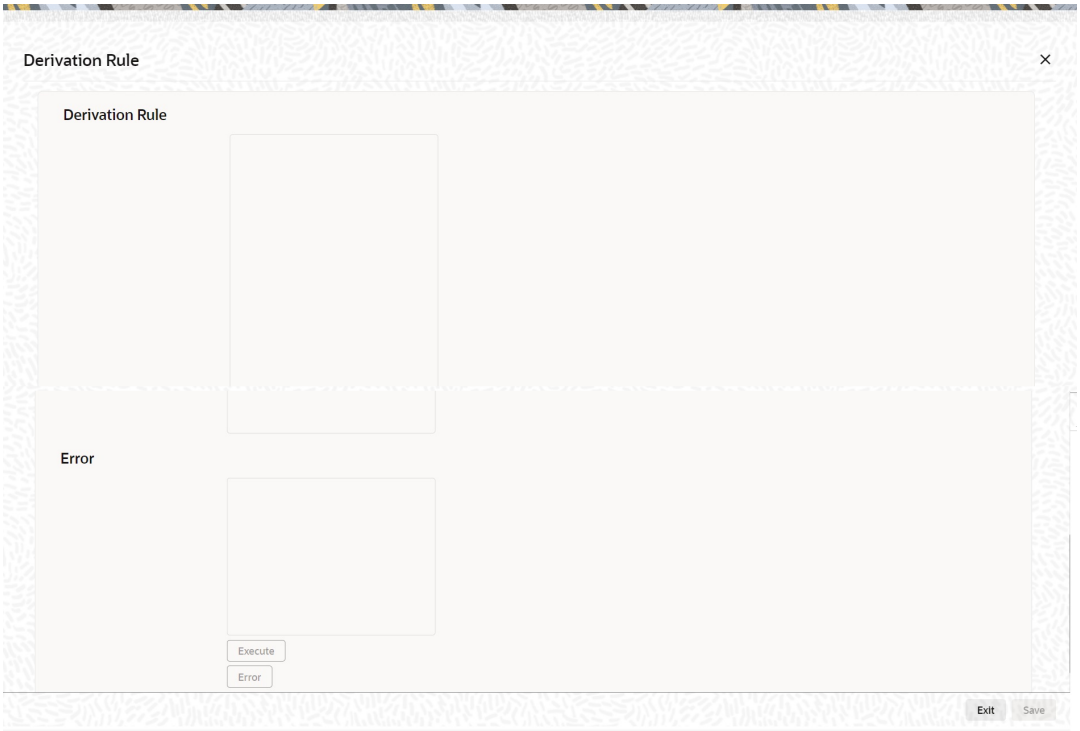
1.4.3 Derivation

This topic describes how to populates the values of a field.

1. This indicates the procedure for populating the values of a field. When you are processing a transaction that would use the user defined field.

   The value of the UDF can be populated in the ‘User Defined Field/Field Name to Value Definition’ screen that is processed from the Contract Input screen or the Function Id screen.
   The value for the field can be populated based on certain conditions, which can be defined as statements of code by the user.
2. Check against **Derivation Allowed** to specify that the value of a field has to be populated based on certain conditions.

Figure 1-3 Derivation Rule



3. Click the **Ok** button to close the screen.

1.4.4 Derivation Rule

This topic describes how to derive the rules.

1. Check against the option **Derivation Allowed** if you want to the values of a UDF to be populated based on certain conditions. According to the requirements of the bank, the implementer of Oracle Banking Treasury Management will write a PL/SQL code to populate the values of the user defined field.

   The values of the UDF will be displayed in the 'User Defined Field/Field Name to Value Definition' screens of Contract Input or Function Id screen.

2. Click **Derivation** button to write the PL/SQL code.
3. After writing the PL/SQL code, click **X** button to execute the code. The derivation code will be validated by the system.

   If any checks fail, you must alter the statement so that the validation can be made successfully.

4. Click **E** button to view the errors.

   For example, the bank wants to have a new field to display the Euro equivalent of the contract amount in the **User Defined Field** screen of Contract Input screen. To do this, you need to:

   - Define a field of type 'Number',
   - Specify 'Usage Allowed' as 'Product',
   - Check against 'Derivation Allowed', and

- Write a code to display the Euro equivalent of the contract amount.

Subsequently, you will link the UDF to a product and process transactions under it. In the 'User Defined Field' screen of Contract Online screen, system will execute the derivation rule to convert the transaction amount to Euro equivalent and displays the Euro equivalent of the transaction amount.

5. After writing the PL/SQL code, click **X** button to execute the code. The derivation code will be validated by the system.

   If any checks fail, you must alter the statement so that the validation can be made successfully.

6. Click **E** button to view the errors.
7. Click the **Ok** button to close the screen.

1.5 Date Field

This topic describes the date fields, specify the default value, how to modify the values after population, factory shipped fields, additional validation rules definition, validation rule, cube entity, and how to enable the allowed option for a UDF.

This topic contains the following sub-topics:

- Validations that you can Specify for Date Fields
  This topic describes how to specify the date fields.
- Mark the Field as Unique
  This topic describes how to mark the field as unique.
- Specify the Default Value
  This topic describes how to specify the default values.
- Allow the Modification of Values after Population
  This topic describes how to modify the values after population.
- Factory Shipped Fields
  This topic describes how to define the default fields.
- Define Additional Validation Rules
  This topic describes how to define the validation rules.
- Validation Rule
  This topic describes how to define the validation rule.
- Cube Entity
  This topic describes how to define cube entity.
- Enable the Update Allowed Option for a UDF
  This topic describes how to enable update allowed option for a user define fields.

1.5.1 Validations that you can Specify for Date Fields

This topic describes how to specify the date fields.

- To define a date field, choose Date as the Field Type. For a date field, you can indicate validations like whether back and future dates can be entered into the field and the back or future period applicable to the field.

Indicate the Validation Type

- You can specify the validation type for a date field. The validation types applicable to a date field include:
  - LOV
  - None

  If you indicate LOV then an entry to the field can be made only from the predefined list that you maintain for the field. Choose None to indicate that no validation should be made for the field.
  Back/Future Date Allowed: For a date field you can indicate whether back or future dates can be entered. Check against the relevant options to indicate your preference.

  If you choose the back or future date options, you should also indicate a future or back period permissible for the field. The back or future period should be represented as a number.

  For example, if you indicate '3' as the back date period, the field will accept dates upto three days before the current system date as a valid entry.

  > **Note:**
  >
  > If you do not choose any of these options, the field will only accept the current system date as a valid entry.

1.5.2 Mark the Field as Unique

This topic describes how to mark the field as unique.

Check against 'Unique' if the field that is being defined has to be unique. Consequently,

- If the usage allowed for the field is 'Product', the new field will be unique across modules.
- If the usage allowed is 'Function Id', it will be unique for the particular function id i.e., if you have specified a function ID, the new field can be used only in the specified Function ID screen).

1.5.3 Specify the Default Value

This topic describes how to specify the default values.

- You can specify a default value for a user-defined field.

  The option list positioned next to this field will be enabled only if the field is a cube entity. Else the option list will be disabled.
  For example, assume that your bank wants to capture an additional currency field for processing certain transactions and the default value for this additional field has to be USD. Therefore, your selections will be as follows:
  - Field Type –Cube Entity
  - Cube Entity Type- Currency

  The option list in the 'Default Value' field displays all the currencies maintained at your bank. Select USD. Consequently, in the UDF screen of the Contract Input or Function Id screen, USD will be displayed as the default currency.

> **Note:**
>
> If you have specified a default value for a field and also the derivation rule, the value obtained from the derivation rule will take precedence over the default value at the Contract/ Function Id level.

1.5.4 Allow the Modification of Values after Population

This topic describes how to modify the values after population.

1. You can modify the value of a field after it is populated in the 'User Defined Field'/'Field Name to Value Definition' screen of Contract Input or Function Id screen.
2. To allow amendments to the values after they are populated, check against the field 'Amendable' at the time of defining a new field.

> **Note:**
>
> You will not be allowed to change the values of those fields for which you have disallowed the amendment option (if the option 'Amendable is unchecked at the time of defining an UDF) and derivation has been allowed (Derivation Allowed option has been checked).

1.5.5 Factory Shipped Fields

This topic describes how to define the default fields.

Apart from the fields that you have created, there are a set of fields that are sent by default as part of Oracle FLEXCUBE. These fields are referred to as factory shipped fields. You cannot define or change validations specified for factory shipped fields.

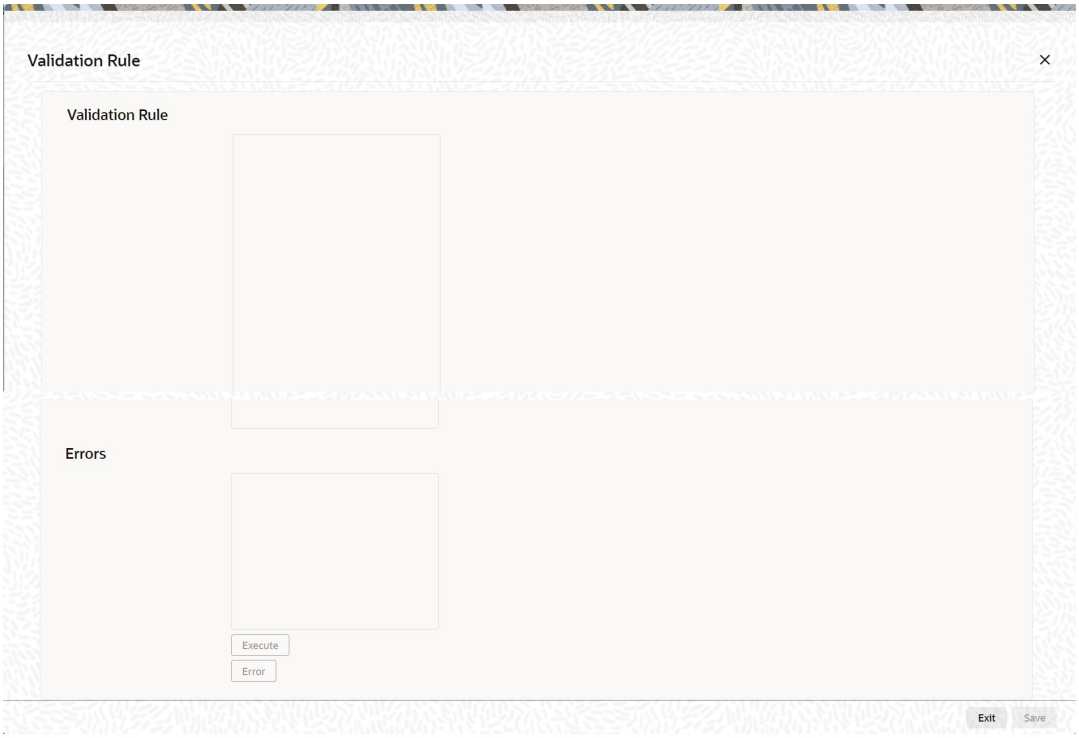
1.5.6 Define Additional Validation Rules

This topic describes how to define the validation rules.

1. Apart from specifying the validation type for a field, you can specify additional validation rules to meet the requirements of the bank.
2. To indicate that additional validations are required for a particular user defined field, check against 'Validation Allowed'.

   This will allow you to write a code to validate the value in the new field.
3. Click **Validation** button to write the validation code.

Figure 1-4 Validation Rule



4. Click **Ok** to close the screen.

1.5.7 Validation Rule

This topic describes how to define the validation rule.

This is the PL/SQL validation code based on which the system will check the value of the new field and validates at the time of transaction processing. Entry to this field is mandatory if you have checked against 'Validation Allowed'.

For example, at the time of processing a contract for a customer, the bank wants to add a new field to enter the first nominee that the customer has specified in his account. The names of the nominees of a customer are maintained in the **Customer Accounts Maintenance** screen. The bank wants the system to check that the name of the nominee that is entered in the new field and the name of the first nominee that is maintained in the Customer Accounts Maintenance screen are the same.

To do this, you need to:

- Create a new user defined field of type 'Text',
- Specify 'Usage Allowed' as 'Product',
- Check against 'Validation Allowed', and
- Write a code to validate the value entered in the new field. In the validation rule, you write a code to check that the value of the first nominee specified in the Customer Accounts Maintenance screen for the customer for whom you are processing a contract and the value entered at the time of processing a contract are same.

Syntax to be used in Derivation and Validation Rules (@FIELD_VAL): (@FIELD_VAL) holds the Value UDF. This is s mandatory for Derivation Logic and it can be used in Validation logic to get the value of the field.

For example,

```
(@FIELD_VAL): = 'USD';


Select ccy_code into (@FIELD_VAL) from


CYTMS_CCY_DEFN where country = 'USA';
```

(@RECORD_KEY): (@RECORD_KEY) behaves differently for UDFs' linked to Product and different for UDFs' linked to Function ID.

(@RECORD_KEY) When UDF is linked to FUNCTION_ID: To use (@Record_key) in UDFs' where Usage allowed is 'Function_id' you have to maintain Function Key Mapping, which will be used to determine the record key before you define the UDF.

Specify the UD Function Key Mapping Details

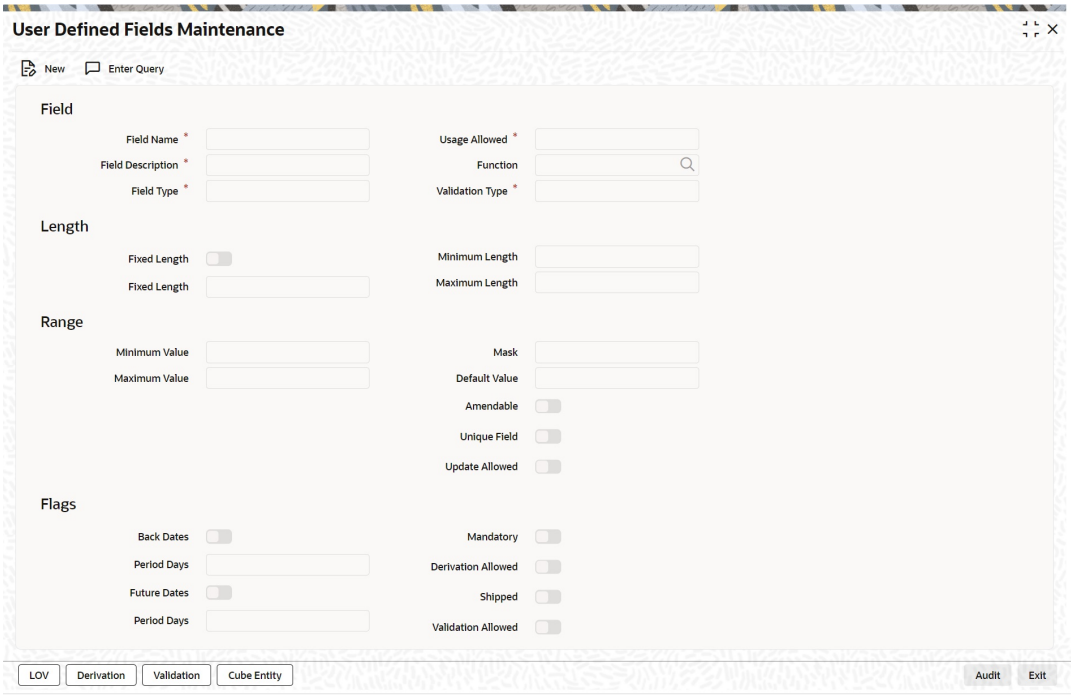
Through the Function Key Mapping screen you can define the Record Key items that are used for the Function ID. This maintenance becomes mandatory if you have to use an UDF in all or any of the Functions.

Specify **User ID** and **Password** and login to **Home** screen.

1. On **Home** screen, type **UDDFNMPT** in the text box, and click next.

   User Defined Fields Function Key Mapping Maintenance screen displays.

Figure 1-5 User Defined Fields Function Key Mapping Maintenance



2. On **User Defined Fields Function Key Mapping Maintenance** screen, specify the fields.

> **Note:**
>
> The fields which are marked in red asterisk are mandatory.

For more information on fields, refer to the field description table.

Table 1-4 User Defined Fields Function Key Mapping Maintenance - Field Description

Field	Description
Function	you must identify the function ID to which you need to link user-defined fields.
Description	The system displays the description of the function.

For example, the Function STDCIF is based on the STTM_CUSTOMER table. The primary key maintained for this table is CUSTOMER_NO. From the STDCIF.FMB, check the block name for the item and then enter the BLK Name (block name in the Form for the Item), ITM_NAME (Item name in the Form) and the Order Number in this screen (Order number cannot be Duplicated and Should be sequential).

3. After completing the Function Key maintenance you can continue defining the UDF for the Function ID.

You will be prompted to link the Function ID (e.g. STDCIF) to the Field at the Definition of the UDF.

(@RECORD_KEY) will hold the value of a combination of all item name values separated by '~' that are mapped in the UD function key mapping screen in the same order of order_no. **The Record key can be used only in the select statement**.

For example, select country into (@FIELD_VAL) from

```
STTM_CUSTOMER WHERE(@RECORD_KEY) and default_media = 'MAIL';
```

(@RECORD_KEY) When UDF is linked to PRODUCT: (@RECORD_KEY) in product level takes the value of the Contract Reference Number (transaction reference number). You need not maintain the function key mapping. You can use the (@RECORD_KEY) in any statements in the Derivation rule and Validation Rule.

For example, select book_date into (@FIELD_VAL) from

```
CSTBS_CONTRACT WHERE CONTRACT_REF_NO = (@RECORD_KEY);


L_prod_code := substr((@RECORD_KEY),4,4);


(@FIELD_VAL) := substr((@RECORD_KEY),4,4);
```

(@RETURN_VAL): This syntax can be used only in validation logic. You can Assign TRUE or FALSE Values to this parameter. Example is given below.

```
(@RETURN_VAL) := TRUE;


if (@FIELD_VAL) = 'USD' then


(@RETURN_VAL) := TRUE;


else


(@RETURN_VAL) := FALSE;


err_code := 'UD-UDF-01';


err_param := (@FIELD_VAL);


end if;
```

The 'err_code' and 'err_param' parameters can also be assigned, and will be displayed if the validation fails in population of values.

(@UDF_field_name): We can use another UDF that is already defined in the derivation logic of a UDF.

For example, we have another UDF DDD

```
(@FIELD_VAL): = (@UDF_DDD);
```

4. After writing the PL/SQL code, click ‘X’ button to execute the code.

   The derivation/validation code is validated by the system. If any checks fail, you must alter the statement so that the validation/derivation can be made successfully.

5. Click **E** button to view the errors.

> **Note:**
>
> You are allowed to Use any syntax in the pl/sql code other than DML statements (INSERT, DELETE, UPDATE) and DBMS package. You will not be allowed to USE any package, function, and procedures other than GLOBAL, DEBUG and CSPKE_MISC.

6. Click the **Exit** button to close the screen.

1.5.8 Cube Entity

This topic describes how to define cube entity.

If the field that you are creating is of the type **Cube Entity**, indicate the entity that is applicable to the field. You can select a Cube Entity from the option list positioned next to this field. This contains the following Cube Entities:

- Currency
- GL
- Branch
- Customer
- Account
- Contract Ref Number
- User Ref Number
- Liability ID

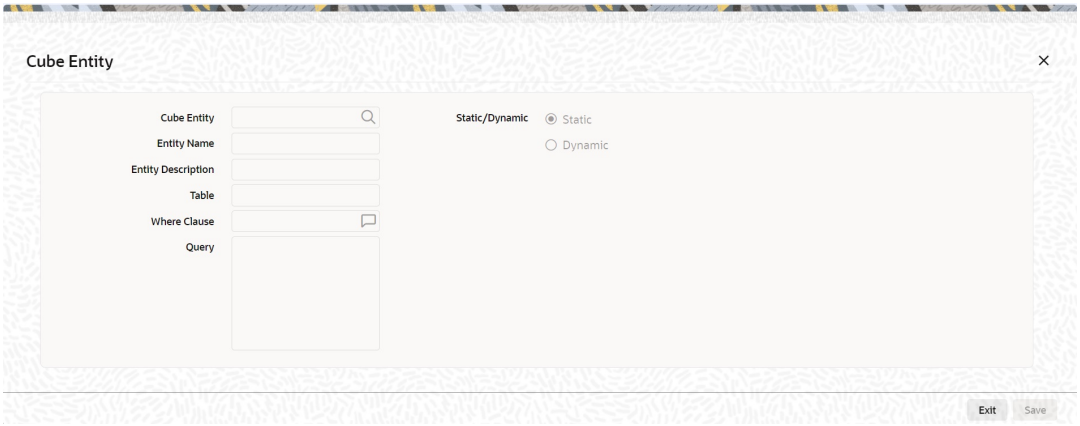
For instance, you need to capture details of an additional customer for a transaction. You can choose the Cube Entity field type and select the Customer field. The list of customers maintained for your bank will be made available at the field as in any other Customer field in Oracle Banking Treasury Management.

Method for Populating the Cube Entity Values

1. If the field type is ‘Cube Entity’, the values of a field can be selected from Oracle Banking Treasury Management tables. If you select the field type as ‘Cube Entity.’

   The **Cube Entity** screen displays.

Figure 1-6 Cube Entity



2. On **Cube Entity** screen, specify the fields.

you need to specify the method by which the values have to be populated. The options available are:

For more information on fields, refer to the field description table.

Table 1-5 Cube Entity - Field Description

Field	Description
Static/Dynamic	**Static** – If you choose this option, system will retrieve the values from a Oracle FLEXCUBE table. After specifying the cube entity, specify the table name, column name and where clause based on which the data will be retrieved from the table. If you have defined a cube entity once, you can reuse it. The adjoining option list displays all the cube entities, which you had used earlier displays. Select the appropriate cube entity. On selection of the reused cube entity, system will automatically display the table name, column name and where clause (if specified for the selected cube entity). You can however modify the condition in the where clause but not the table name and column name. **Dynamic** - If you choose this option, the values on the UDF will be populated based on the query written by the user **Note:** At the time of installation, the implementer will write the PL/SQL code for the derivation rule, validation rule and also the query for fetching a cube entity according to the requirements of the bank.
Table	Depending on your selection in the 'Cube Entity' field, system displays the table name in which the selected cube entity is located. The value of the column (which is specified in the previous field) from this table will be picked up to populate the values of the new field.
Description	After you enter the cube entity, you may enter a brief description of the cube entity. This description will be used for information retrieval.

Table 1-5 (Cont.) Cube Entity - Field Description

Field	Description
Where Clause	Enter the condition based on which the values from the specified column and table, the values of the new field should be picked up. Enter the condition in this field if you have indicated 'Static' method for populating the values of the cube entity.
Query	This is the code, based on which the values of the new field will be picked up. The query has to be written if the option 'Dynamic' is selected.

3. Click the **Ok** button to close the screen.

1.5.9 Enable the Update Allowed Option for a UDF

This topic describes how to enable update allowed option for a user define fields.

The Update Allowed option is used during Event Processing. You can assign values to a UDF when a UDE is processed. During event processing, the system assigns certain UDF parameters based on the Execution Query you have maintained in the Event Processing screen. You will be allowed to assign UDFs only for those UDFs for which the Update Allowed option has been enabled.

1.6 Make a Field Applicable to a Product

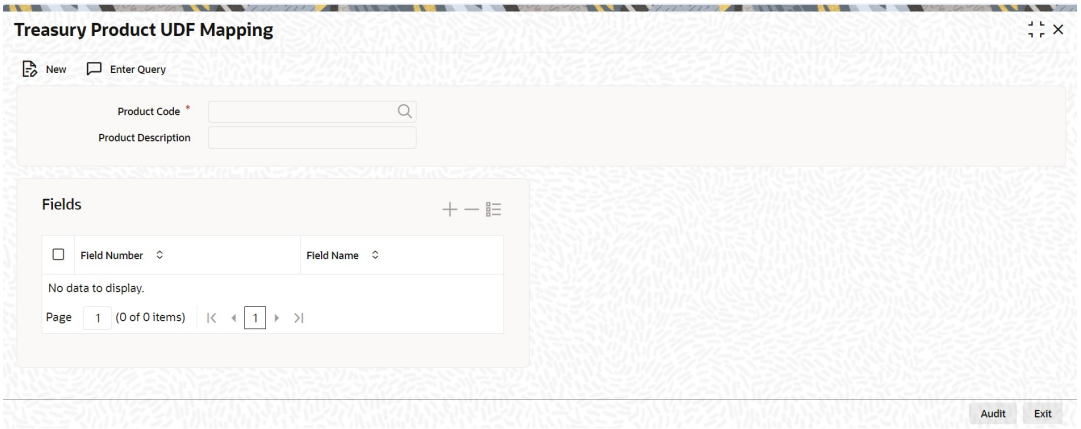
This topic describes a systematic instructions to make a field applicable to a product.

The fields that you define in the User Defined Fields screen can be made applicable to the products (and thereby to the contracts) that you create for the front-end modules of Oracle Banking Treasury Management depending on your selection in the 'Usage Allowed' field.

For example, suppose you have defined a user-defined field DATE1 with 'Usage Allowed' as Product, the UDF will be displayed for association only in the Product Definition screen of a module.

1. At the time of creating a product click the **Fields** button from the 'Product Definition' screen. From this list of all the fields that you created, you can choose specific fields and make them applicable to the product.

Figure 1-7 Product User Defined Fields



In this screen, you can link the user-defined fields to the product that you are defining. Oracle Banking Treasury Management allows you to link up to two hundred fields to a product.

2. To link a user-defined field, click **add** icon.

   A list of all the user-defined fields for which the Usage Allowed is 'Product' will be displayed.

3. Select the fields that have to be included in the Contract Input screen for the contracts processed under the product that is being defined.

   When a contract is processed under that product, the fields associated with the product will be available in the Contract Input screen.

4. Click the **Ok** button to close the screen.

1.7 Creating UDF

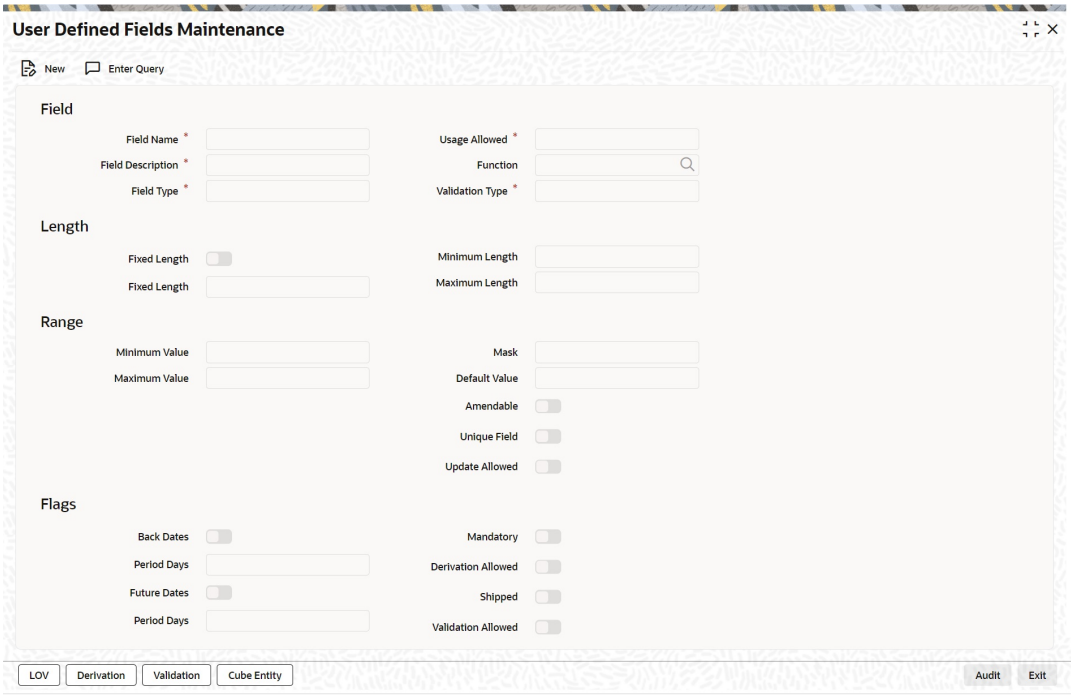
This topic describes the systematic instruction to create user defined fields.

You can create UDFs using the **User Defined Fields Maintenance** screen for a particular Function Id.

1. On the Homepage, Type **UDDUDFMT** and click next arrow.

   The **User Defined Fields Maintenance** screen is displayed.

Figure 1-8 User Defined Fields Maintenance



2. If you select **Usage Allowed** as **Product**, then you need not provide Function Id. You can fetch these fields in the **Product Definition** screen.

3. If you select **Usage Allowed** as **Function**, then you need to provide the Function Id in the **Function** field.
4. After saving and authorizing the record in the **User Defined Fields Maintenance** screen, go to **User Defined Fields Function Field Mapping Maintenance** screen.

- Link UDF to Function ID
  This topic describes the systematic instruction to link UDF to function ID.

1.7.1 Link UDF to Function ID

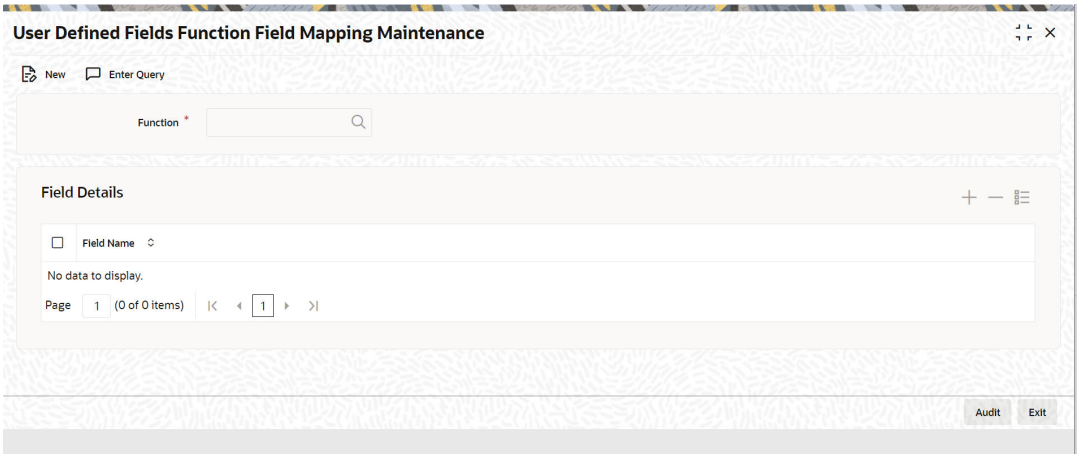
This topic describes the systematic instruction to link UDF to function ID.

You can link the user-defined fields (fields for which 'Usage Allowed' is Function Id) to a function Id in the **User Defined Fields Function Key Mapping Summary** screen.

1. On **Home** screen, type **UDDFFLMT** in the text box, and click next.

   User Defined Fields Function Key Mapping Maintenance screen displays.

Figure 1-9 User Defined Fields Function Key Mapping Maintenance



2. On **User Defined Fields Function Key Mapping Maintenance** screen, specify the fields.

   For more information on fields, refer to the field description table.

Table 1-6 User Defined Fields Function Key Mapping Summary - Field Description

Field	Description
Function	Select the Function Id from the adjoining option list. For instance, if you want to include the user defined fields in the **Chart Of Accounts- Detailed** screen, select the function Id **GLDCHACT**. After selecting the Function Id, click add icon to select the fields that have to be included in the selected Function Id screen. A list of all function Id 's for which the Usage Allowed is specified as 'Function Id' will be displayed. Select the appropriate fields that have to be included from the list. Oracle Banking Treasury Management allows you to link up to two hundred fields to a Function ID.
Authorization Status	Select the authorization status from the drop-down list.
Record Status	Select the record status from the drop-down list.

After you save and authorize the record, the user defined fields will be included in the specified Function Id screen.

3. Click **Search** button.

   The screen displays the following details:

   - **Authorization Status**
   - **Record Status**
   - **Function**

4. Click the **Exit** button to close the screen.

1.8 Population of UDF Values at the Contract Level

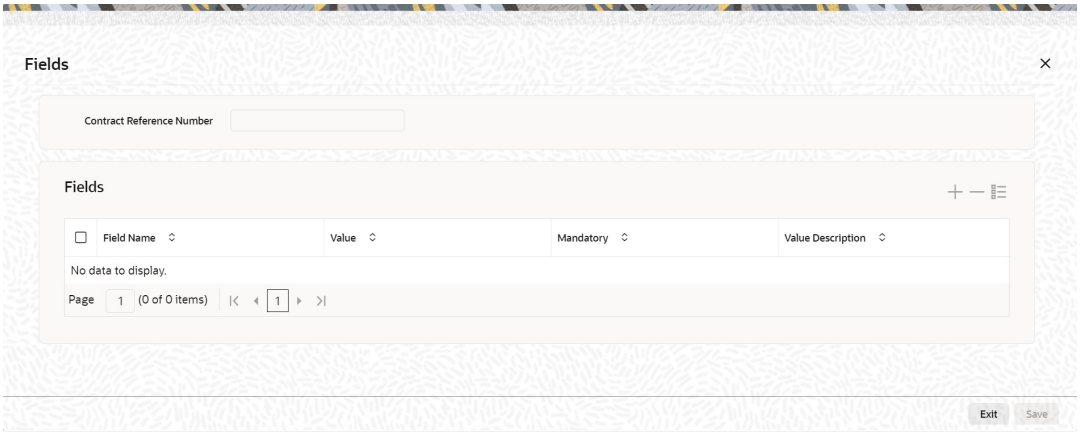
This topic describes the systematic instructions to populate the UDF values at the contract level.

The defaulted or derived values of the UDF linked at the product level will be populated in the Contract Input screen at the time of processing contracts under a particular product. Similarly, the values of the UDF's linked to a function Id will be populated in the function Id screen. The UDF's linked to a particular product will be available when a contract is processed under that product.

1. On the Contract Input screen, click **Fields** button.

   User Defined Fields screen displays.

Figure 1-10 User Defined Fields



In this screen, all the user defined fields associated with the product under which you are processing the contract will be displayed.

According to your specifications at the time of defining the user defined fields, system may derive the values of the UDF from:

- **Derivation Logic**: When you click 'Fields' button, system derives and displays the values of the fields for which derivation rules are written. The value description is also displayed alongside.
- **Static List of Values**: If the validation type of the UDF that is associated with the contract is 'List of Values' (LoV), system will display the static values, along with the value description. If you have indicated a default value for the LoV, the default value and its description will be displayed. When a user selects the value from the LoV, the system displays the description on the screen.

- **Method of derivation for cube entity fields**: If a UDF is of type Cube Entity, system populates the values in a LoV. The values will be derived from the specified Oracle Banking Treasury Management table (if the cube entity type is ‘Static’) or from the query written by the user (if the cube entity type is ‘Dynamic’)

You will not be allowed to change the values of the fields for which you have not allowed amendment (if the option ‘Amendable is unchecked at the time of defining an UDF) and derivation is allowed (Option ‘Derivation Allowed’ is checked).

2. Enter the values for the fields, for which the values are not derived by the system.

   You can change the value of a UDF after the system has derived the value from the derivation logic. But it might so happen that the derivation logic of another UDF might use the value of the UDF that you have changed. Consequently, if you change the value of the UDF whose value is used in another UDF, the value of that UDF will also change.

   For example, let us assume that you have defined a user defined field UDF1 to display the Euro equivalent of the contract currency.

   The derivation logic of another user defined field (UDF2) utilizes the values of UDF1 to arrive at the value of UDF2. Assume that in the derivation logic written for UDF2, you have specified that the value of UDF2= 2 UDF1.

   At the contract level, system converts the contract currency into Euro equivalent and displays the value of UDF1, as 220.00. According to this value of UDF2 will be 444.00. However, if you change the value of UDF1 to 250.00, it will effect on the value of UDF2 also (it becomes 500.00).

   If the change in the UDF value has effected any other UDF, system will display an override informing you about the UDF whose value will be changed.

3. Select **Ok** button if you want the system to re-calculate the value of the UDF based on the modified UDF value.
4. Click the **Ok** button to close the screen.

1.9 Upload User Defined Fields

This topic description contract upload and function upload of contract.

This topic contains the following sub-topics:

- Contract Upload
  This topic describes how to upload contract.
- Function Upload
  This topic describes how to upload function.

1.9.1 Contract Upload

This topic describes how to upload contract.

Your bank may use Oracle FLEXCUBE to process certain transactions (say LD and FT transactions) and use another application to process other transactions (say MM and FX transactions).

At some point of time, your bank may want to use the values from the external system in Oracle FLEXCUBE. For this purpose, Oracle FLEXCUBE stores the values of the various fields from the external system (an application other than Oracle FLEXCUBE) in a database table. If your bank wants to use the value of the fields from the external system in Oracle

FLEXCUBE, the values have to be brought into Oracle FLEXCUBE database. This is done through a process called ‘Upload’.

Let us assume that your bank is processing LD transactions in Oracle FLEXCUBE and MM transactions in an external system. For processing certain LD transactions, your bank wants to use the values from certain MM transactions. Therefore, you will have to put the value of the contract reference number along with the field name and value of the MM transactions, which you want to use in Oracle FLEXCUBE into a database Upload table with the status marked as ‘U’ (indicating that it has to be taken up for processing). This Upload table contains the values posted from the external system. Subsequently, you will have to bring the values of these fields into Oracle FLEXCUBE. This is done by running the contract upload program.

When you run this program, Oracle FLEXCUBE selects the fields for which the status is ‘P’. It will upload the values from the external system into Oracle FLEXCUBE database.

1.9.2 Function Upload

This topic describes how to upload function.

The values of the fields from external system used in the maintenance screens should be uploaded into Oracle FLEXCUBE database.

Index

U

UDDFFLMT- User Defined Fields Function Key Mapping Maintenance, *1-19*

UDDUDFMT- User Defined Fields Maintenance, *1-1*, *1-18*