

Generic Interface

Version-10.2

9NT832-ORACLE FCUBSV.UM 10.2.0.0.0.0.0

[October] [2008]

Oracle Part Number E51712-01

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/Testing teams	Approved by: Software Quality Assurance Team
Updated on : October 01, 2008	Reviewed on : October 01, 2008	Approved on : October 01, 2008

Table of Contents

1. GENERIC INTERFACE	1-1
1.1 MAINTAINING INTERFACE DETAILS	1-1
1.1.1 <i>Maintaining the External System details</i>	<i>1-1</i>
1.1.2 <i>Maintaining the Interface Definition details</i>	<i>1-2</i>

1. Generic Interface

1.1 Maintaining Interface details

Oracle FLEXCUBE interfaces with other systems installed at your bank (Murex and PSGL). In order to streamline the exchange of data between Oracle FLEXCUBE and these systems (Murex and PSGL), several functions have been added to Oracle FLEXCUBE. You can maintain these functions through the Generic Interface function.


The interface between Oracle FLEXCUBE and the external system can either be a batch process or can be done online depending on the requirement. Besides when the interface type is outgoing and the handoff is to be done online then Oracle FLEXCUBE will trigger the relevant interfaces automatically.

In the next few sections, we will learn how to utilize the Generic Interface of Oracle FLEXCUBE to its optimum advantage.

1.1.1 Maintaining the External System details

The basic requirement for this interface to function effectively is the maintenance of the External System (s) with which Oracle FLEXCUBE is supposed to interface. You can do this through the External Interface maintenance screen. Invoke this screen by clicking on External System under Generic Interface in the Application Browser.

Input By	Date / Time	Auth By	Date / Time	Mod No
TAMTTT	12-FEB-2001 10:51:05	RAV000	12-FEB-2001 10:53:25	1

To maintain details of a new system choose New from the Actions Menu or click  from the toolbar. The External Interface maintenance screen will be displayed without any details.

An external system that you have defined will become operational in Oracle FLEXCUBE only after it is authorized. A user bearing a different Login ID can authorize a external system definition record that you have created.

Defining the External System

For every external system that you maintain you need to ascribe a suitable name to identify it. The name that you give the system should contain a maximum of ten characters. It can be a combination of alphabet and numeric characters. You can follow your own convention for devising the code.

Since you can more than one external system it is necessary that you give every external system a unique name.

It is also important that you specify a brief description of the external system. The description that you enter is for information purposes only. Here are some examples of names of external systems along with their descriptions:

- ACCSERVER - Account Server.
- MUREX – Murex

1.1.2 Maintaining the Interface Definition details

After you maintain the names of external systems with which Oracle FLEXCUBE is supposed to interface you can go on to defining the actual interface related details.

You can maintain these details through the Interface Definition screen. To invoke this screen, choose Generic Interface from the Application Browser. Thereafter, choose Interface Definition. The Interface Definition screen is displayed without any details.

Through this screen, you can:

- Indicate whether it is an Incoming or Outgoing type of interface.
- Indicate whether the handoff is to be done Online or as a Batch process.
- Specify the components of the message.
- Define fields at the Message, Component and Record level.

Branch Code: 000 External System: ACSERVER

Type: Incoming Outgoing

Mode: Batch Online

Interface Code: INT21

Description: INTERFACE CODE

Directory: jupiter/cube80/common

Filename: AanDDMMRRRR

Last Run Date: Next Run Date: 12-FEB-2000 Interval: 1

FTP: Required Not Required

Acknowledgement: Required Not Req

CRC: Required Not Required

On Error: Reject Hold

On Override: Hold Ignore Reject

Buttons: Pre Message Trigger, Post Message Trigger, Interface Trigger, Message Components, Field Definition

Input By	Date Time	Auth By	Date Time	Mod No
RAJ000	12-FEB-2001 14:37:50			1

Open: Authorised:

1.1.2.1 Indicating the External System

As a first step in defining the interface details you have to specify the name of the system with which Oracle FLEXCUBE is supposed to interface. The names of all the external systems maintained in Oracle FLEXCUBE will be available in the picklist. You can select the appropriate external system.

1.1.2.2 Specify the interface code and description

You can associate every interface that is carried out in the system with a unique Interface Code. This code identifies the interface between the external system and Oracle FLEXCUBE.

Once you specify the Interface Code you need to briefly describe what the interface is all about. The description that you enter is for information purposes only and will not be printed on any correspondence with your customer.

1.1.2.3 Indicating the Interface Type

After you associate the interface with a unique code and a suitable description you have to indicate whether the interface is an outgoing interface or an incoming interface.

An outgoing interface is one, which results in the transfer of data from Oracle FLEXCUBE to an external system. When data is received into Oracle FLEXCUBE from an external source it is called an incoming interface.


1.1.2.4 Indicating the Handoff Mode

Every time you define a new interface the system automatically generates an output file. You need to specify the manner in which the output file is to be handed off. Whether the handoff should be done Online or whether it should be a part of the Batch processes.

When you specify that the handoff is to be made as part of the batch processes you are also required to select the batch process during which handoff is to be made. You can select any one of the following options:

- End of transaction Input.
- End of Financial Input.
- End of Day.
- Beginning of Day.

Depending on your preference the handoff is made when any one of the above processes is run Oracle FLEXCUBE.

 As stated earlier the output file contains interface-related data. When the interface is an outgoing interface this data will be sent to the external system either online or during one of the batch process depending on your preference. In the case of an incoming interface the system either creates a new record or updates existing data. This is again dependent on what you specify in the Action field in the Message Components screen.

1.1.2.5 Identifying the Interface Trigger

When the interface type is outgoing and the handoff is to be done online then you are required to maintain certain parameters based on which Oracle FLEXCUBE will trigger the relevant interfaces mandatorily.

Click on the **Interface Trigger** button in the Interface Definition screen. The Interface Trigger screen will be displayed. In this screen you can select the trigger type based on which the corresponding interface has to be triggered in Oracle FLEXCUBE. You can select any one from the trigger types listed below:

- Accounting.
- Messaging.
- Static Maintenance.
- Contracts.

Module	Event	Authorization Status
ST	NEW	Authorized

STOP Note that the Interface Trigger button is enabled only when the interface type is outgoing and the handoff mode is online.

Selecting the Driving Table

You are required to select an appropriate driving table only when Static Maintenance happens to be the trigger type specified. You can choose the relevant table name from the list of all the existing static data maintenance tables available in the system. As a consequence every time some change is made to the particular static maintenance table a corresponding outgoing interface will be triggered for the Static Maintenance, Driving Table, Event and Authorization Status combination.

Selecting the Module

When the trigger type selected happens to be Contracts then you have to indicate the module in Oracle FLEXCUBE to which the particular contract belongs. A list of all modules in Oracle FLEXCUBE is available you have to choose the correct module name.

A corresponding outgoing interface will be triggered in Oracle FLEXCUBE for the Contract, Module, Event and Authorization Status combination.

Specifying the Event and the Authorization Status

When you specify Contracts, Messaging or Accounting Entries as the trigger type you are required to indicate the event and the status of the event for which the interface needs to be triggered.

For example you might want to trigger an outgoing interface every time a Hedge type Derivative Contract is accrued. Similarly you can specify that every time a Message is generated for a particular event bearing a specific status a corresponding interface has to be generated.

Similarly you also need to specify the status of the event for the particular Trigger Type and Event combination. The status of the event can be either one of the following;

- Authorized.
- Unauthorized.

The interface will be triggered when the specified combination of Trigger Type, Driving Table (when the Trigger is Static maintenance) Module (when the Trigger is Contracts) Event and Authorization Status are in sync with each other.

1.1.2.6 Specifying the date on which the batch process is to be run

When the handoff for a particular interface is to be made as part of the batch processes you are also required to indicate the date on which the next handoff should be run for that interface.

The date that you specify should be later than the current application date.



Note that when you have specified that the interface is a batch interface the date on which the handoff was last run is automatically displayed by the system.

In addition to specifying the date on which the interface will be run next you have to indicate the interval period in terms of days. The interval period is the date on which the last hand off was run to the date on which it will be run next.

1.1.2.7 Specifying the area in which the data should be stored

When the interface results in the inward transfer of data you have to specify a valid area in the server where the received data is to be stored. You can do this by way specifying the name of the directory under which data has to be stored. When the interface is of Type outgoing, the system writes the flat files in this directory.

In addition to identifying a directory in the server you have to indicate the filename mask from which data is to be read. For outgoing interfaces data is written according to the filename mask parameters that you set.

An example of the filename mask would be: DD, MM, MON, HH.

1.1.2.8 Specifying the File Transfer Protocol details

You have to indicate whether it is necessary to specify the File Transfer Protocol (FTP) details. When FTP details have to be maintained you have to indicate the Internet Protocol Address (IP Address) of the server and a specific area within the server where the incoming data is to be stored.

Consequently for all incoming interfaces, once the transfer of incoming data is complete the system will automatically retransfer data (from the temporary directory) to the new FTP area you have specified. When the interface is of Type outgoing, the system writes the flat files from the temporary directory into the FTP directory.

Subsequent to specifying the server name and the directory in which data has to be stored you have to specify the Login ID or the User ID and the password, which grants you access rights to save or extract data or perform similar operations in the FTP server.

1.1.2.9 Specifying the Message Component details

An interface message can be made up of many components, and each component in turn consists of single/multiple records. While defining the interface related details you have to identify the various components which need to form a part of the message.

To identify the list of components which should form a part of the interface message click on **Message Components** in the Interface Definition screen. The Message Components screen is displayed.

The screenshot shows the 'Message Components' configuration window. At the top, there is a table with one row containing the component ID 'M1'. Below the table are several configuration sections:

- Type:** Acknowledged (dropdown)
- Sequence Number:** 1 (text input)
- Fixed Length Fields:**
- Delimited:** **Delimiter:** (text input)
- Tag:**



There are also sections for **Incoming** and **Outgoing** with various clause buttons like **Where Clause**, **Order By Clause**, **Group By Clause**, and **Having Clause**. A **Component Break** section includes a **Break required** checkbox and buttons for **Header Fields** and **Footer Fields**.

At the bottom, there are **Component Level** and **Record Level** sections, each with **Start Identifier** and **End Identifier** fields (both set to 1) and **Pre Trigger** and **Post Trigger** buttons.

Indicating the Component name

Every component within a message needs to be identified with a unique name. Therefore you need to assign a suitable name to every component that is to be included in the message. You can define as many components as you want.

After you specify the name of the component you can briefly describe the component in the adjacent field. The description that you capture is for your reference only and will not be printed on any data that is transferred.

To define a new component click on , enter the name of the component. To delete an existing component, highlight it and click .

Specifying the message type

Since every message consists of a Header, a Footer and a Body you need to indicate the way in which the various components are to be structured within the message. The structure of the message is determined by how you segregate the various components. Therefore you need to identify each component that you define either as the Header, the Footer or as the Body component.

You can identify components as Acknowledgement components only when the message is an incoming message.

Indicating the Sequence number

By specifying the sequence number you can indicate the manner in which the various components are to be sequenced in the output message. For instance when the component named COMP1 is to be part of the message header then you enter 1 as the Sequence number. Similarly when COMP2 and COMP 3 are to be a part of the body of the message then you need to enter 2 and 3 respectively as the Sequence number.

The arrangement of components in the output file is entirely dependent on the way in which individual components have been sequenced.

Indicating the uniqueness of individual fields

Each component consists of multiple records and each record contains fields with valid data in them. You can indicate whether the fields present in each record within the component have to be of a Fixed Length, should have a Delimiter whether they should be Tagged or whether they should be a combination of all three.

If you specify that all fields should only be of a fixed length then all fields for records within the particular component will only be of the length that you specify.

You can also indicate that the text of all fields for records within the particular component should end with a delimiter character. The text of each field of records within the particular component will display the delimiter character in the output message. For instance if you want the text of every field to begin on a new line you can specify @LF (Linefeed) or if the text has to be separated with a tabbing character you can specify @TB.

Similarly you can indicate the tagging character that should precede the start of data within a field.



Note that you can specify the Field length and the Tagging character while maintaining details of individual fields in the Field Definition screen. The details of this screen have been explained under the sub-head Defining Fields.

Indicating the Action for incoming interfaces

When the transfer is of incoming type you have to indicate whether the system has to create new records in Oracle FLEXCUBE for storing the incoming data or whether existing records have to be updated. You can indicate the same by selecting one of the two options:

- Insert or
- Update.

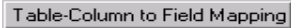
If you specify that the system has to Insert new records upon receipt of data as a result of the incoming interface you have to indicate what should be done when a record for the same data already exists in the system. Whether the system should insert a new record regardless of existing data, whether it should reject the newly received data or whether the new data is to be ignored. The system will take the necessary action based on your preference.

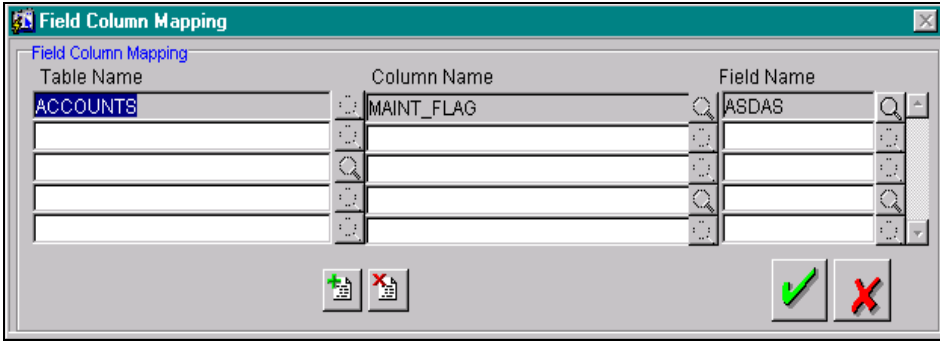
Similarly if you specify that the system has to Update existing records every time new data is received, you have to indicate what should be done when previous records for the same data do not exist. You can specify any one of the following options:

- Update – a new record is created in Oracle FLEXCUBE.
- Reject - the newly received data will be rejected.
- Ignore - the newly received data will be ignored.

Indicating the Table Column to Field mapping



When the interface is an incoming interface you have to map the data that is received to a table in Oracle FLEXCUBE. Depending on your preference of action in the Action field the system will either update existing data or insert a new record for the new data.

To map a column in a table with a specific field you have to click the  button in the Message Components screen. The Table-Column to Field mapping screen will be displayed.

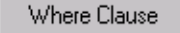


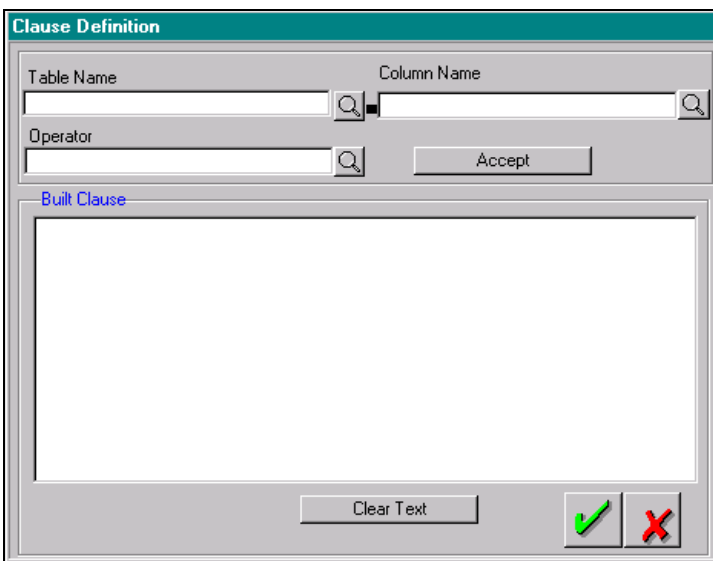
In this screen firstly you have to specify the name of the Table in Oracle FLEXCUBE. All the columns within that particular table will be listed in the picklist available for Column Name. Identify the column within that table into which data should be inserted or updated.

Next you have to specify the field within the record from which data will be extracted and updated or inserted into the specific column of the table in Oracle FLEXCUBE.

To map another table and column combination with a specific field click  and follow the procedure listed above. To delete a mapping that you have made, highlight the mapping and click .

Specifying the where clause

Occasionally while receiving incoming an interface you may be required to update existing information on a conditional basis. Conversely in the case of outgoing interfaces you might want to extract data based on certain parameters. Oracle FLEXCUBE offers you the choice of doing so. Click on the  button. The Clause Definition screen will be displayed.



In this screen indicate the table and column name combination based on which data is to be inserted/updated (in the case of incoming messages) or extracted (for outgoing messages).

After you specify the table and column name combination you can choose the operative clause for data extraction for outgoing interfaces and insertion or updation of records for incoming interfaces. Listed below are some examples of available operators:

- >.
- <.
- =.
- > =.
- < =.
- AND.
- OR.

Defining the Clauses for outgoing interfaces

Apart from defining the Where Clause, for outgoing messages you are allowed to define various other clauses based on which data should be extracted. The other clauses based on which you can extract data are as follows:

- The Order By clause.
- The Group By Clause.
- The Group By clause.

For example let us assume that you want to order the way in which records within a particular component have to be structured. Invoke the Order By Clause Definition screen by clicking on the relevant button in the Message Definition screen. Indicate the table and column combination in Oracle FLEXCUBE from which data is to be extracted and select the operative clause based on which the extracted data is to be ordered or structured.



Similarly invoke the Group By Clause Definition screen when extracted data has to be summarized. If summarized data needs to be restricted by specific criteria, you can specify the same through the Having Clause definition screen.

Indicating whether message components should be sub-divided

Each component consists of single/multiple records and each record has a number of fields within it. Further each component can be structured to form the Header, the Footer or the Body of the message.

While identifying the various components, which should form a part of the interface message you can indicate whether the fields available within a component can further be grouped together into logical groups of records.

For instance, let us assume that you have specified that the component bearing the name COMP1 should be the message header. Now there are some fields in COMP1, which need to be grouped together according to certain criteria. Indicate that the component break is required and go on to maintain the group Header fields (fields that occur in the top section of the group) and the group footer fields.

To separate the header and footer fields within a component click on the  and  buttons respectively. The Fields Definition screen is launched. You can start identifying individual fields as header and footer fields.

Indicating the Group level Start and End identifiers

Since you can logically group together a set of records you should identify where the particular grouping of records begin by capturing data (in the form of characters) that is present in the beginning of a group.

Just as you identified specific data, which occurs at the beginning of the group of records so also you need to identify data, which marks the closure of a particular group of records.

Indicating the Component and Record level Start and End identifiers


Just as you identify group level start and end identifiers, it is important that you specify the start and end identifiers at the component and the record level. This identification is necessary for the system to differentiate the component or record (as the case may be) as being separate from the other components or records existing in the file.


You can indicate the start and end identifiers by capturing data in the form of characters such as @LF to indicate a line feed or @TB to indicate tabs.

1.1.2.10 Defining Fields

Just as you define components that should become a part of the interface message so also you can define fields that should either become a part of the interface Message, the Component or the Record. This is dependent on how you launch the Field Definition screen.



For instance if you launch this screen from the Interface Definition screen you will be allowed to define fields which will appear in the interface message. Similarly to define fields for individual components or records you have to invoke the Field Definition screen from the Message Components screen.

Click on  either from the Interface Definition or the Message Components screen depending on which level you are defining the fields for. The Field Definition screen will be displayed.

 If you want to segregate the header and footer fields within a component you have to click on the respective buttons under the Break Required section of the Message Components screen. The above screen will be displayed.

Indicating the Field name and description

Specify a suitable name to the field that you are defining. After you specify the name you can briefly describe this field in the adjacent field. The information that you capture about the field is for your reference only and will not be printed on any data that is transferred.

At any level you can define as many fields as required. To define a new field click on , enter the name of the field. To delete an existing field, highlight it and click .

Indicating the field level

As said earlier the fields that you define can be used at different levels in the interface. The field that you define at the message level will be used as a temporary variable for the entire message and can be referenced to other fields within the interface. Similarly a field defined for the component will become the temporary variable for that component and can be referenced with other fields within that component. Record level fields appear in the message and can be referenced both other fields within that record.

Specifying the type to which the field belongs

In addition to indicating the level at which the field will be used you have to specify the Type to which the field you are defining belongs. it could be either one of the following:

- Message - the field will be physically present in the handoff file.

- Intermediate – the field will not be present in the handoff file but will be used to derive certain other fields.



When specifying the level if you indicate that the field you are defining is a Message field or a Component field then the value in the Type field will default to Intermediate, you will not be allowed to change it.

Only while defining Record fields you can specify whether the particular field is to be a Intermediate field or a Message field.

Indicating the Type of data that the field should contain

A field can either contain data in the form of numerals, a combination of numbers and alphabets or data in the form of dates. You have to specify the type of data that the field you are defining should contain.

When the data in a specific field is meant to be in date format you have to specify the date mask as well. For instance you can specify that the date should be in the following format:

DD-MMM-YYYY

The data in the generated file will contain the specified data in the relevant format.

The Group Type

The value that is displayed in the Group Type field is entirely dependent on where you launch the Field Definition screen from. For instance let us assume that while maintaining the interface details you decide that the fields within a component need to be grouped together, then this field will default to Header or Footer depending on which screen you invoke. In all other cases the value of this field will be displayed as Normal.

- Normal - the system will recognize the field as a normal field.
- Header - the system will recognize the field as a part of the header group of fields.
- Footer - the system will recognize the field as part of the footer group of fields.

Sequencing the fields in a particular order

The sequence number determines the sequence in which individual fields will appear within the records.



Note that for Intermediate fields you will not be able to define a sequence number.

Specifying the other details of the field

While defining a field you can choose to associate it with a default value. This value will be recognised by the system as the default value of the field.


Also, every field that you maintain should be of a particular length. You can specify the field length in terms of characters. You can also specify the starting position of the field in terms of number of characters in the file.

After you specify the length of the field, its default value and the position of the field in the file you have to decide the way in which the field should be aligned. You can either choose to have it Left aligned, Right aligned or Justified.

Indicating whether Translation is required

In the case of incoming as well as outgoing interfaces you have to indicate whether data received or extracted has to be read as it is or whether it has to be translated into a different value. You can choose one of the following options to indicate this:

- Required
- Not Required.

If you specify that data translation is required then click on the  button and specify the other mandatory parameters such as the Input and Output Values.

The input value is the value of the particular field in Oracle FLEXCUBE when it is translated. The output value is the value of the particular field in the external system with which Oracle FLEXCUBE is interfacing.

Entering the Padding Characters

You can maintain some padding characters that will be prefixed or post fixed (depending on the justification) on to the existing value of the field being defined.

For example let us assume that you are maintaining the field BRANCH. The requirement is that every time the length of this field is less than let's say three characters 0 should be appended to BRANCH so that the end result is 001. You can do so by specifying the padding character as 0.

Indicating the Tag

While defining the message component if you have indicated that the fields present in records within a component have to be tagged then you can indicate the tagging character that should precede the start of data within a field.

The text of every field within that component will display the tagging character that you have indicated.

Specifying the table from which data should be extracted

When the interface is of Type outgoing you can indicate that for the field being defined data should be extracted from an existing table in Oracle FLEXCUBE.

If data is to be extracted from an existing table you have to indicate the Table and Column name combination from which data should be extracted. From this list of valid table names available in the database select the name of the table from which data is to be extracted for the specific field.

After you select the name of the table from which data is to be extracted for the outgoing interface the picklist available for Column Name will get populated with the names of all columns within that table. You can select the name of the particular column from which data is to be extracted.

If you do not want to extract data from an existing table the Default Value that you have set for the field will be picked up and recognised as the actual value of the field.

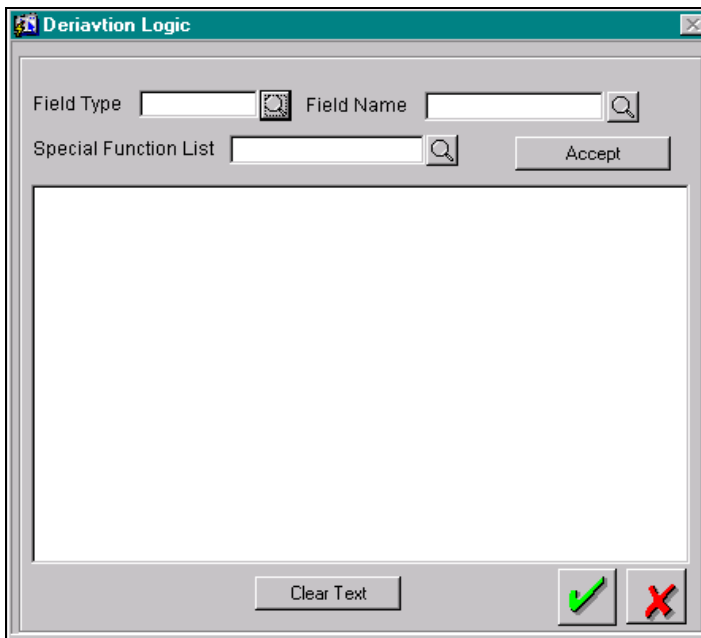
Specifying how incoming data is to be validated

While receiving data as a result of incoming interfaces you can specify that all data received for the field you are defining should be validated against data already existing in Oracle FLEXCUBE.

If you indicate that incoming data for this field has to be validated against a particular field in Oracle FLEXCUBE then you have to specify the Table and Column Name combination against which data has to be validated.

Deriving data from existing fields

While defining a particular field you might want to extract or derive data from existing fields. In such as case click on the Derivative Logic button and invoke the Derivative Logic screen.



In this screen you can do the following:

- Access/assign values from/to field belonging to higher levels.
- Access (cannot set) values to fields at the same level.
- Reference with global variables in Oracle FLEXCUBE such as @GLB.

Firstly you have to select a valid Field Type from the list of field types maintained in the system. They are as follows:

- @REC_ to indicate record level fields.
- @MSG_ to indicate message level fields.
- @CMP_ to indicate component level fields.

After you specify the Field Type the system populates the names of all the fields available under the particular Field Type in the Field Name list. You can specify the name of the field from which data is to be extracted.



Generic Interface
[October] [2008]
Version 10.2

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 © [2008] 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.