

Translation Framework

Oracle FLEXCUBE Direct Banking

Translation Framework

Release 12.0.3.0.0

Part No. E52543-01

April 2014

ORACLE®

Translation Framework

April 2014

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax:+91 22 6718 3001

www.oracle.com/financialservices/

Copyright © 2008, 2014, Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 failsafe, 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.

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.

This software or hardware and documentation may provide access to or information on 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. 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.

Contents

1. Preface	4
1.2. Documentation Accessibility	4
1.3. Access to OFSS Support	4
1.4. Structure	4
1.5 Related Information Sources	5
2 Abbreviations	6
3 Adding new language in FCDB	7
4 Define language validation on DATA_DICTIONARY Level	8
5 Define language validation on TXN_DATA Level	9
6 Some Guidelines	10
7 Reference Documents	11

1. Preface

1.1. Intended Audience

This document is primarily targeted at

- Oracle FLEXCUBE Direct Banking Development Teams
- Oracle FLEXCUBE Direct Banking Implementation Teams
- Oracle FLEXCUBE Direct Banking Implementation Partners

1.2. Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

1.3. Access to OFSS Support

<https://support.us.oracle.com>

1.4. Structure

Oracle FLEXCUBE Direct Banking, the multi channel solution, for direct customer touch points like Internet and Mobile devices have introduced translation in different languages so that it is available for implementation in particular language with minimal changes

The purpose of this document is to outline the requirements to support integration of multiple languages in the Oracle FLEXCUBE Direct Banking product with data input and validation on the business requirement.

This document provides overview on the current methodology for the accepting multiple data input and validation for Oracle FLEXCUBE Direct Banking.

The entire Translation project can be divided into following phases

1. Adding new language in FCDB
2. Define language validation on DATA_DICTIONARY Level
3. Define language validation on TXN_DATA Level

1.5. Related Information Sources

For more information on Oracle FLEXCUBE Direct Banking Release 12.0.3.0.0, refer to the following documents:

Oracle FLEXCUBE Direct Banking Licensing Guide

2 Abbreviations

The following abbreviations may be used within the document.

FCDB	Oracle FLEXCUBE Direct Banking
LDB	Local Database

3 Adding new language in FCDB

For adding a new language in FCDB there is a master table for language i.e. MSTLANG. So first an entry is made into this table. The table has following columns:

IDLANG: Language identifier

DESCRIPTION: Language description

RESPENCODING: Response encoding

XSLENCODING: XSL encoding

ISO_LANG: ISO language identifier

ISO_COUNTRY: ISO country code for that language

SUPPORTED_RANGE: UNICODE range or Regular expression of character that are supported in that language

BLOCKED_RANGE: UNICODE range or Regular expression of character that are blocked in that language

DEFAULT_DATE_FORMAT: Default date format for that language

DECIMAL_SEPRATOR: decimal separator for that language

TIME_STAMP_FORMAT: Default time stamp format for that language

For validating data against particular language main columns are SUPPORTED_RANGE and BLOCKED_RANGE. At server startup time application creates a regular expression on basis of this and validates all the fields against this regular expression (This validation is only applicable when data type is defined as String[S]).

For date format and time stamp format earlier this was defined at TXN_DATA i.e. at each field level. Now default date format and default time stamp would be created on language basis. All data input must be in that particular format.

Since for UI we are using JQuery and for JQuery date format is in different form. So language specific date format is mention in corresponding language specific JS file. This file is located at **WAR/JS/i18n/<filename>**. The name of file is like **ui.datepicker-<iso_lang_code>.js** . So if the application doesn't find the language specific file at the above location it would pick the default settings.

4 Define language validation on DATA_DICTIONARY Level

One can configure the data level field validation on the basis of data dictionary. Data dictionary is the master level of field validation. Multiple attributes could be defined here like minimum length, maximum length, white list characters and pattern if any. To override the default validation, a new table is created i.e. TXN_DATA_LANG_MAP. Snapshot of this table is as follows:

	TEMPLATE_ID	IDLANG	FIELDFORMAT	NAMFIELD
1	NARRATIVE	spa	[\u00C1\u00E1\u00C9\u00E9\u00CD\u00ED\u00D1\u00F1\u00D3]	...
2	STR_DESCRIPTION2	spa	[\u00C1\u00E1\u00C9\u00E9\u00CD\u00ED\u00D1\u00F1\u00D3]	...
3	BENEADDRESS	spa	[\u00C1\u00E1\u00C9\u00E9\u00CD\u00ED\u00D1\u00F1\u00D3]	...
4	ADRESS3	spa	[\u00C1\u00E1\u00C9\u00E9\u00CD\u00ED\u00D1\u00F1\u00D3]	...
5	STR_DESCRIPTION3	spa	[\u00C1\u00E1\u00C9\u00E9\u00CD\u00ED\u00D1\u00F1\u00D3]	...
6	ADRESS1	spa	[\u00C1\u00E1\u00C9\u00E9\u00CD\u00ED\u00D1\u00F1\u00D3]	...

This table has following columns:

TEMPLATE_ID: In this field we give the data dictionary type which is going to over ride.
 IDLANG : In this filed we give language identifier for which this data type is going to override.

FIELDFORMAT: In this field we specified language specific pattern.

NAMFIELD : This field is not necessary to override data dictionary type. So it will remain null.

For examples suppose you use NARRATIVE data type of a field and login in application in Spanish. So application checks if there is any entry in TXN_DATA_LANG_MAP table for this data type against Spanish language. In case it doesn't find any entry in TXN_DATA_LANG_MAP, it picks the default pattern from data_dictionary.

By this way ones can define their language specific field pattern. Please note this is only applicable for those fields which have data type as pattern. For String type we are cam override only white list characters present in this table.

5 Define language validation on TXN_DATA Level

Data level field validation is maintained in TXN_DATA. TXN_DATA has field level validation which one can override master pattern which is defined at DATA_DICTIONARY level. For this we use same table TXN_DATA_LANG_MAP but for TXN_DATA level column uses is modified:

TEMPLATE_ID	AT.GENERICPAYMENTSERVICEINTERFACE.GENERICPAYMENTSERVICE.INITIATEGENERICPAYMENTDETAILS.UDFFIELDS
IDLANG	spa
FIELDFORMAT	\u00C1\u00E1\u00C9\u00E9\u00CD\u00ED\u00D1\u00F1\u00D3\u00F3\u00DA\u00FA\u00DC\u00FC\u00AB\u00BB\u00BF\u00A1\u20...
NAMFIELD	fldnarrative

This table has following columns:

TEMPLATE_ID: This field holds the request id.

IDLANG : This field holds the language identifier.

FIELDFORMAT: This field holds language specific pattern.

NAMFIELD : This field contains reference field name of corresponding TXN_DATA element.

For example, to override narrative validation of own accounts transfer. An entry is made in TXN_DATA_LANG_MAP table as follows:

- Proper id request as per txn_data table for narrative field is put (e.g. OAT.GENERICPAYMENTSERVICEINTERFACE.GENERICPAYMENTSERVICE.INITIATEGENERICPAYMENTDETAILS.UDFFIELDS).
- Corresponding language identifier, field format and reference field name is added.

So validation engine checks if there is any entry in TXN_DATA_LANG_MAP table for this field for the particular language else it will look for the default validation in DATA_DICTIONARY table.

6 Some Guidelines

1. In XSL files do not define elements name against description (if you need dynamic names for elements). Always use code, not description.
2. Try to minimize use of pattern validator. Use only when it is required.
3. In some cases we find field format of TXN_DATA or DATA_DICTIONARY as regular expressions defined for String data type. Please note that field format is only applicable for pattern data type. It should not be used for String. In string we define some white listed characters with "#" (hash) separated.

For example: If you require a field which accept alphanumeric with some special char like "_" then define data type instead of using "Pattern Type" use "String Type" and also define this char ("_") as a white list char in field format.

Please find the sample bellow:

FIELDNAME	TD_PRO_CODE	...
FIELDDESC	Product Code	...
FIELDTYPE	S	
MINLENGTH	1	
MAXLENGTH	6	
FIELDFORMAT	#_#	...
COMMENTS		...

4. Define date type in TXN_DATA doesn't required field format, so leave it blank. Earlier Date format was on field level now date format is defined at language level in MSTLANG for particular languages.

7 Reference Documents

<i>Sr.No</i>	<i>Name of Document</i>
1	FCDB_DS_MULTILINGUAL.docx