Oracle FLEXCUBE Universal Banking ® 12.0.2

Development of Report Forms

August 2013

Contents

1. Preface

This document guides the developer in designing a Report form using Oracle FLEXCUBE Development Workbench for Universal Banking

1.1. Audience

This document is intended for FLEXCUBE Application developers/users that use Development Workbench to develop various FLEXCUBE components.

To Use this manual, you need conceptual and working knowledge of the below:

Proficiency	*Resources*
FLEXCUBE Functional Architecture	Training programs from Oracle Financial Software Services.
FLEXCUBE Technical Architecture	Training programs from Oracle Financial Software Services.
FLEXCUBE Object Naming conventions	Development Overview Guide
FLEXCUBE Screen Development	*04-Development_WorkBench _Screen_Development-I.docx*
Working knowledge of Web based applications	Self Acquired
Working knowledge of Oracle Database	Oracle Documentations
Working knowledge of PLSQL & SQL Language	Self Acquired
Working knowledge of XML files	Self Acquired

1.2. Related Documents

04-Development_WorkBench_Screen_Development-I.pdf

05-Development_WorkBench_Screen_Development-II.pdf

2. Introduction

This document provides information on:

- Chapter 1 , "Introduction"
- Chapter 1 , "Overview of FLEXCUBE Reports"
- Chapter 2 , "Designing Report Form"

3. Overview of FLEXCUBE Reports

Reports in FLEXCUBE UBS are used to fetch the data from FLEXCUBE database Schema based on the report Query criteria and render on screen or print.

FLEXCUBE UBS uses two software as reporting platforms.

- Oracle Business Intelligence Publisher (BIP)
- Oracle Business Intelligence Enterprise Edition (OBIEE)

This document describes the process of designing a Report Form for a BIP Report using Development Workbench. The process of designing RTF and XDO files are not explained in this manual.

Note:
This document assumes a given report RTF file and data template XML is already available.

4. Designing Report Form

1. Login to FLEXCUBE Development workbench

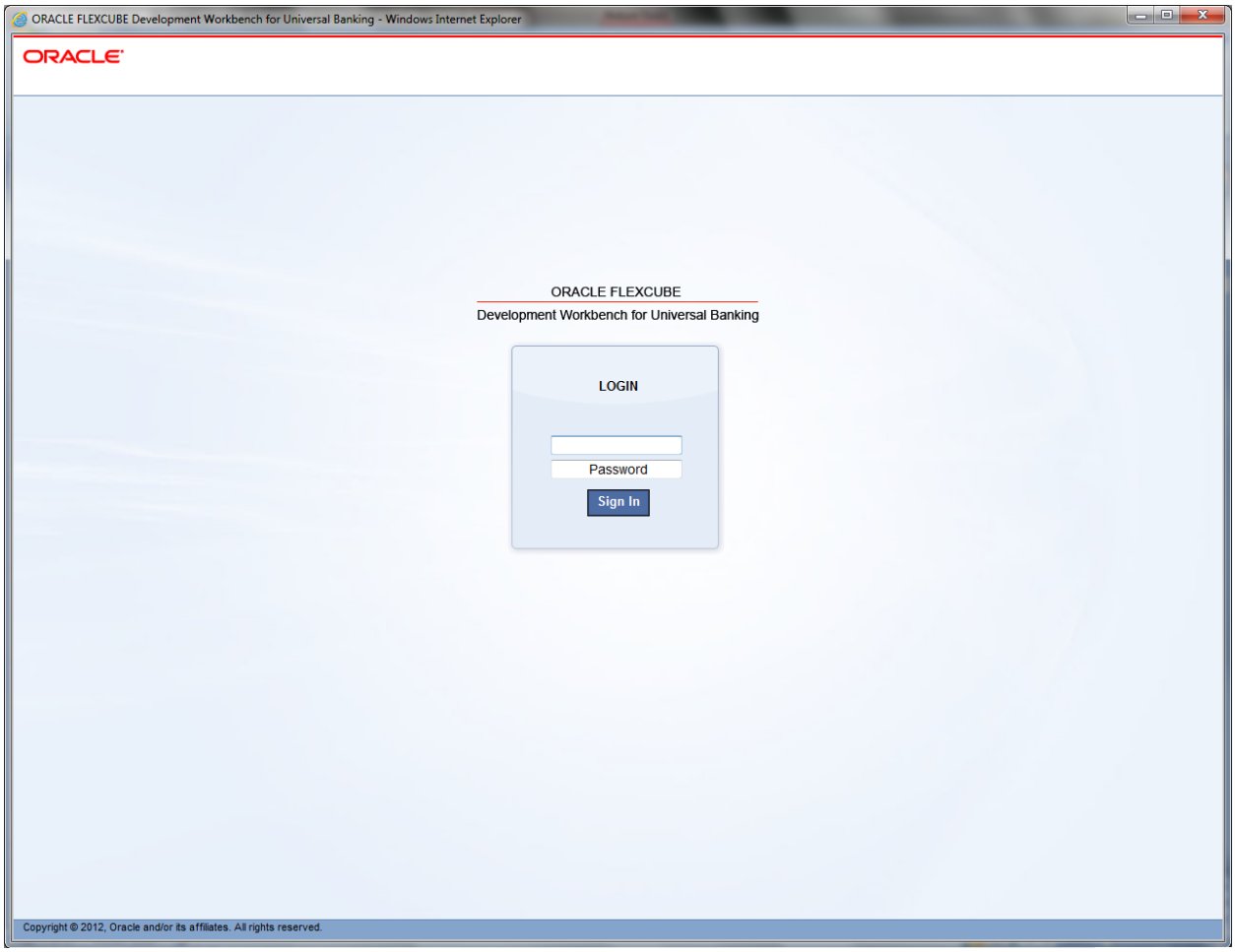


Fig 4.1 Login Page

2. From browser, select 'Function Generation'

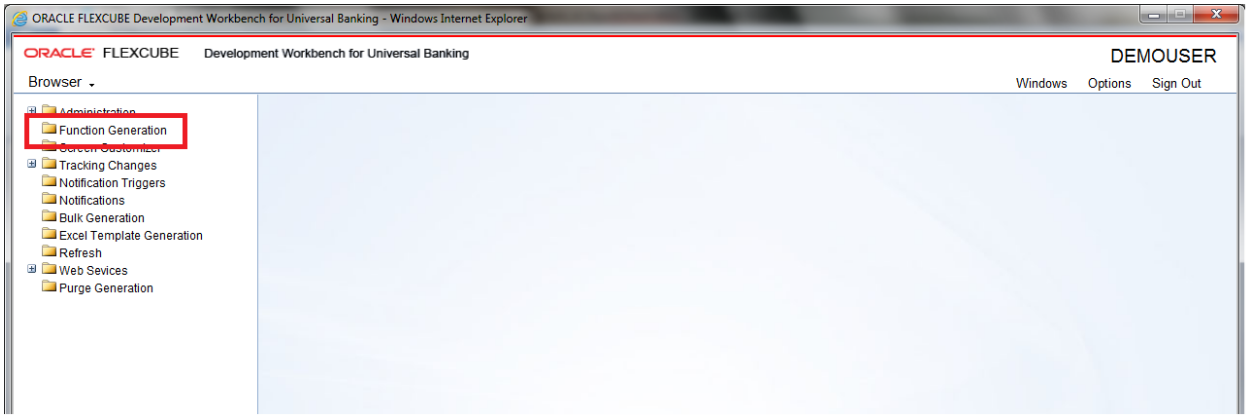


Fig 4.2: Function Generation Screen

3. Select ‘New’ from ‘Action’, and set the function type as ‘Report’ from the drop down list in ‘Function Category’. Type in the name of the screen; make sure that the third letter of the name is ‘R’.

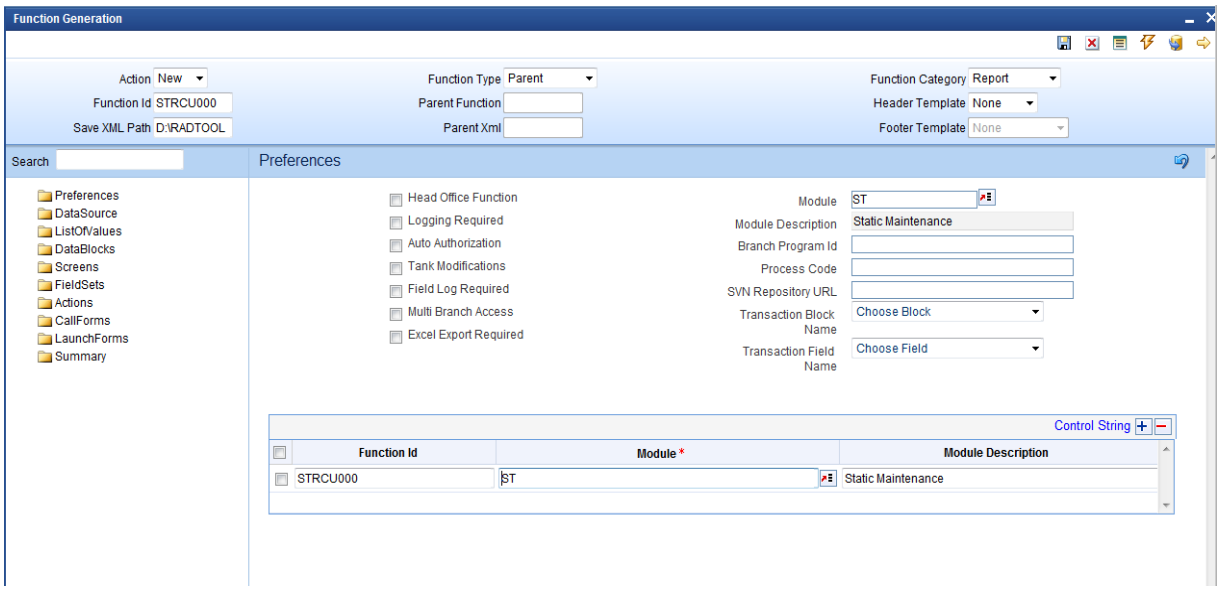


Fig 4.3: Report Screen design

4. Add data sources by right clicking on ‘DataSource’. In a report screen, the only table that should be added is ‘**CSTB_UI_COLUMNS**’. Add columns from the table as required. The number of columns to be added depends on the number of report parameters required in the report screen. Add as many fields as required in the report screen.

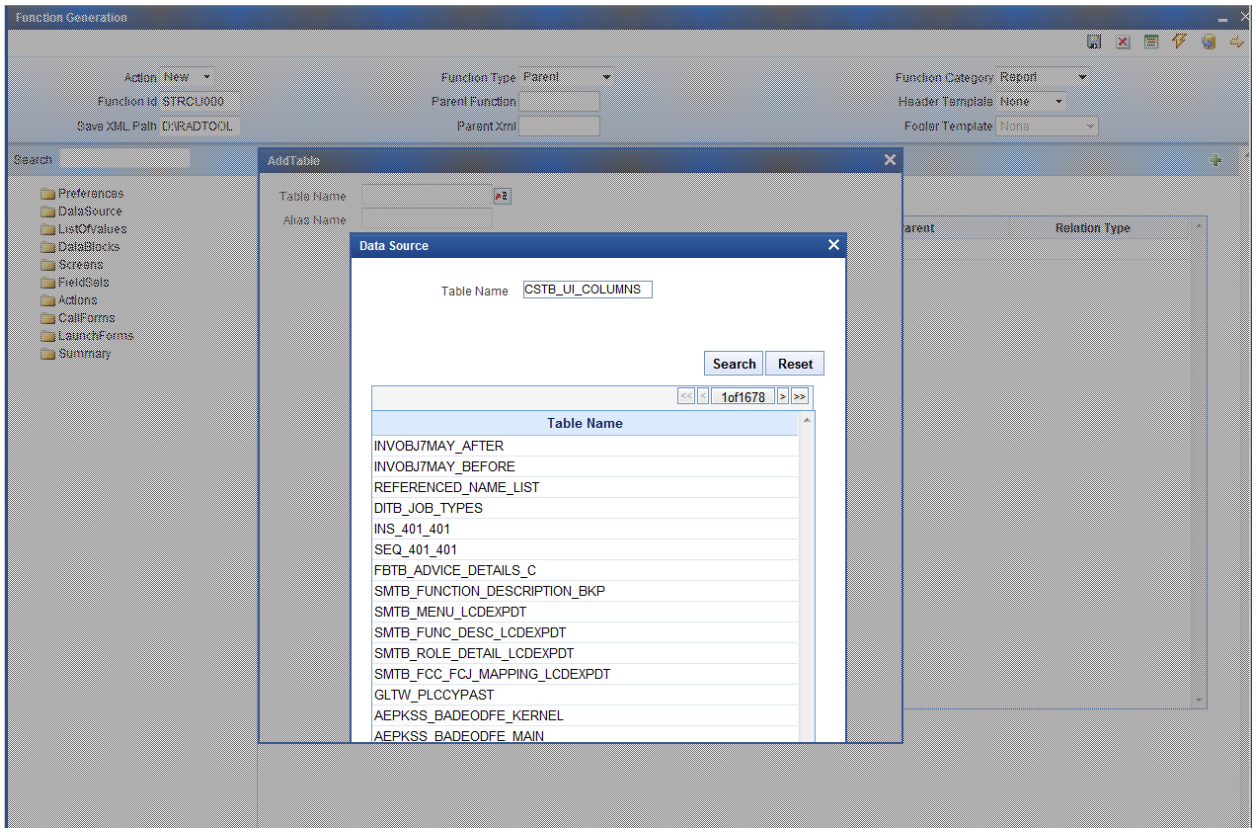


Fig 4.4: Adding Data Sources

5. Right click on data blocks and add block fields.

There are some generic report parameters which needed to be added in any report screen. They are:

PM_MINOR

PM_BRANCH_CODE

PM_BRANCH_DATE

PM_BRANCH_DESC

PM_CURRENT_USER

PM_LCY

PM_LANGUAGE

PM_MODULE

Note: While adding columns to the data source CSTB_UI_COLUMNS, take these generic parameters also into consideration

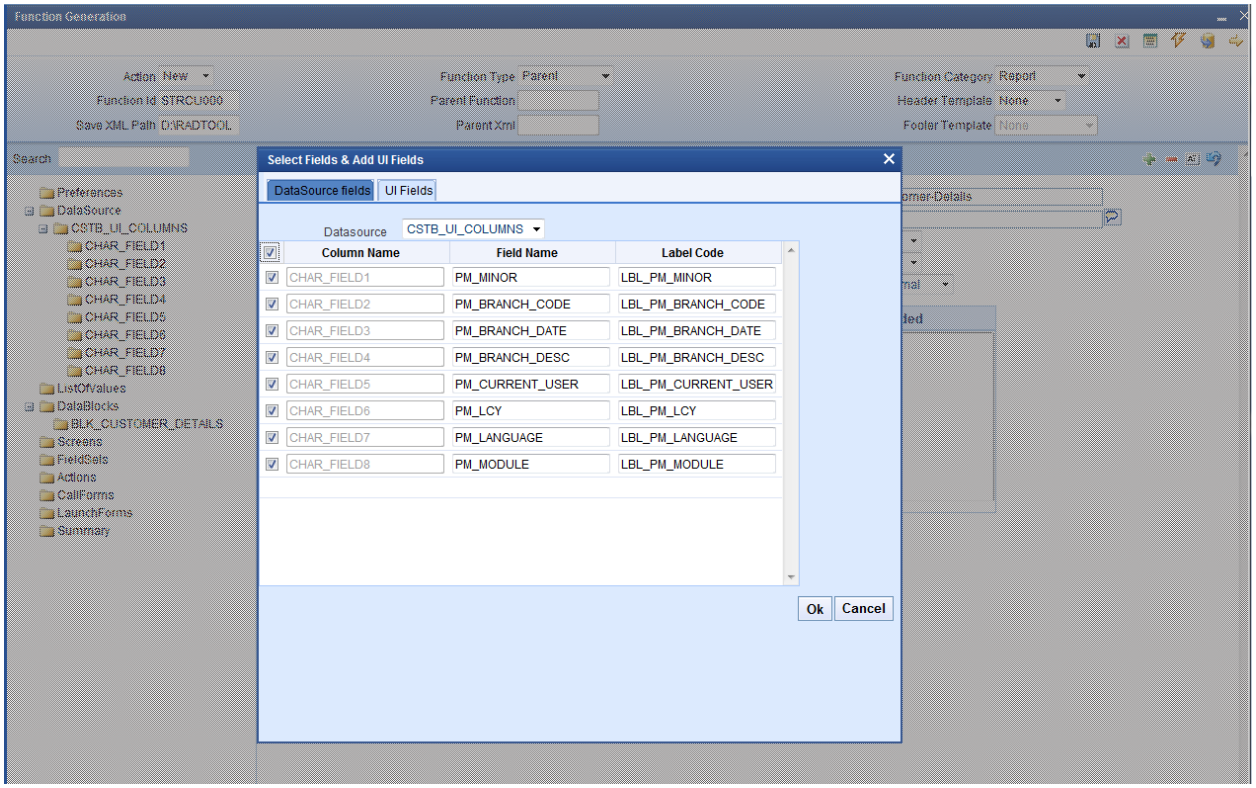


Fig 4.5: Adding Data Block Fields

The 'Report Parameter' checkbox has to be checked in case of report parameters. Also, 'Visible' has to be unchecked in all generic parameters. Also, default values are to be given to some of these fields.

For example,

PM_BRANCH_CODE specify the Default value as GLOBAL.CURRENT_BRANCH.
PM_BRANCH_DATE specify the Default value as GLOBAL.APPLICATION_DATE.
PM_CURRENT_USER specify the Default value as GLOBAL.USER_ID.

Development Workbench would automatically add the Report Options for all Report screens

These report options are

- Report Format
- Report Output (Print, Spool, View)
- Print At (Client/Server)

- Printer ID

6. To add new field set, right click on field set.

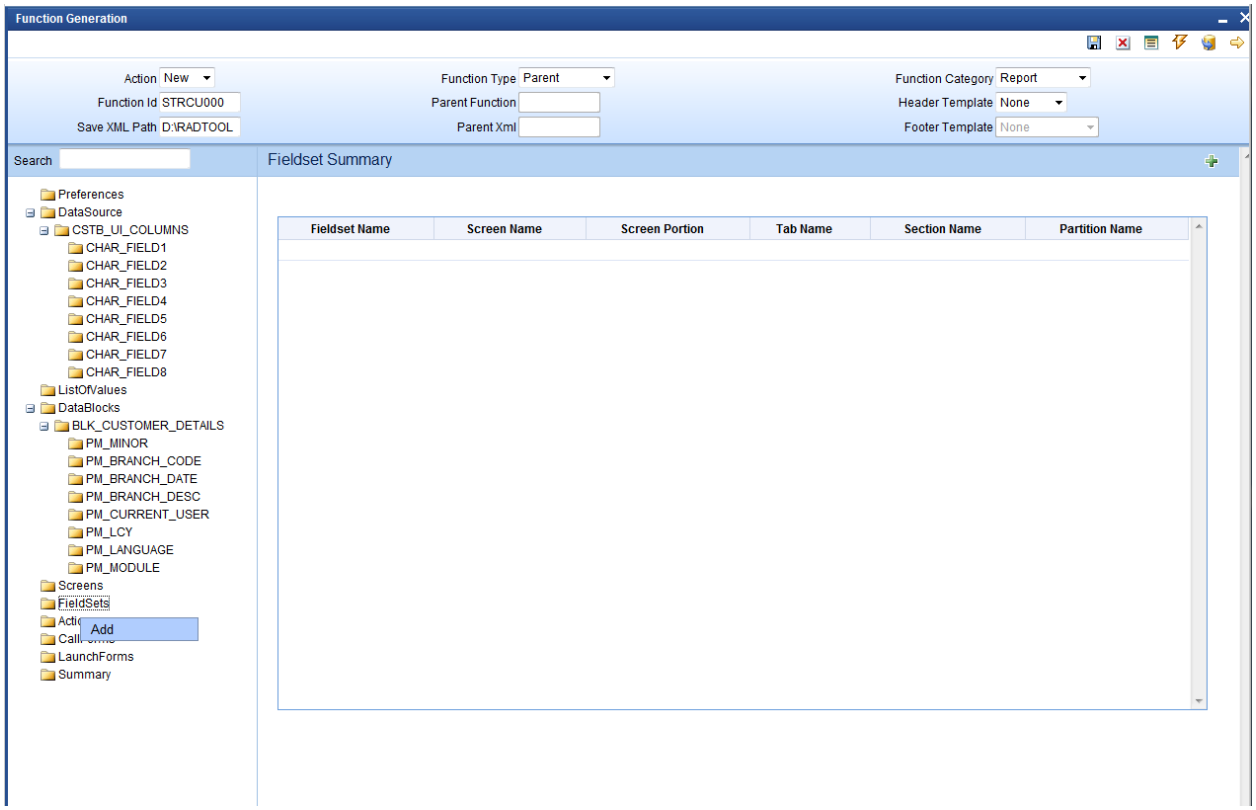


Fig 4.6: Adding field sets

7. Give field set properties:

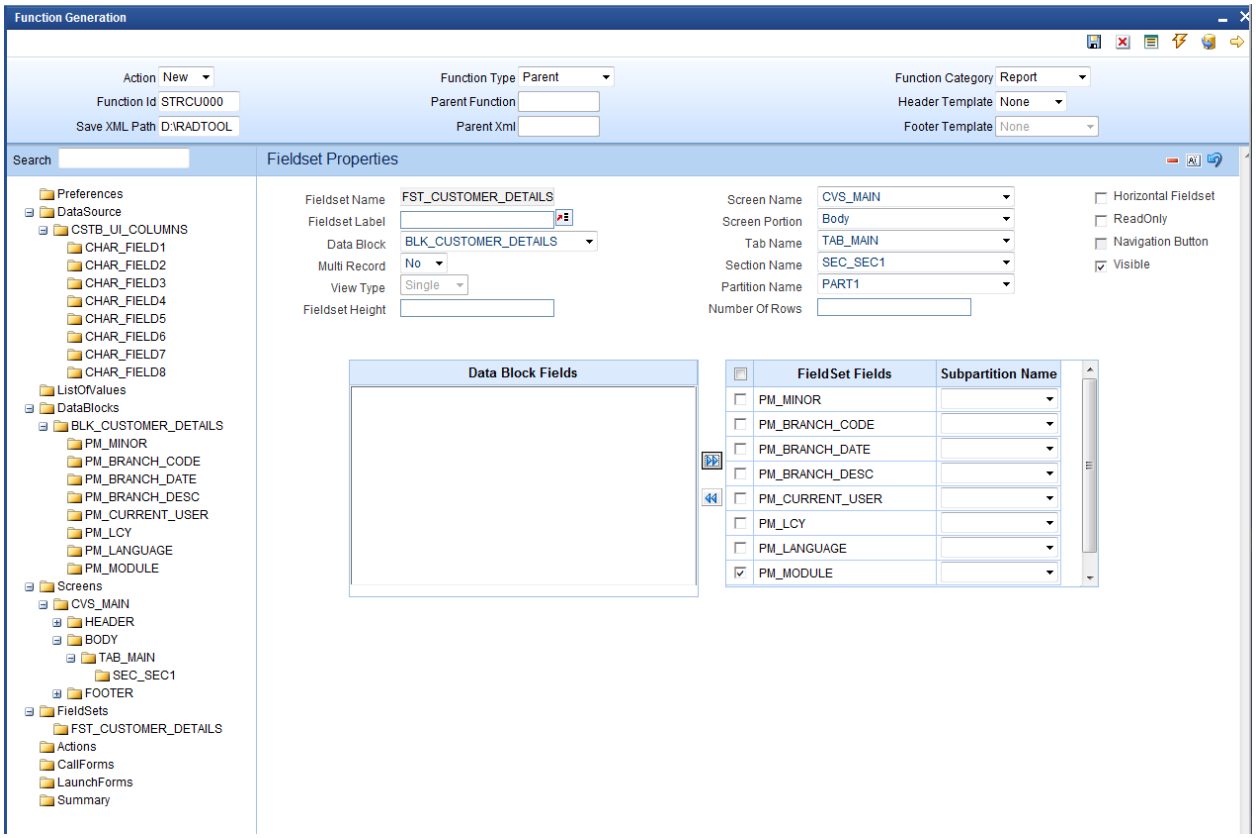


Fig 4.7: Providing Field set Properties

To preview the screen, right click on the screen name and select 'Preview'.

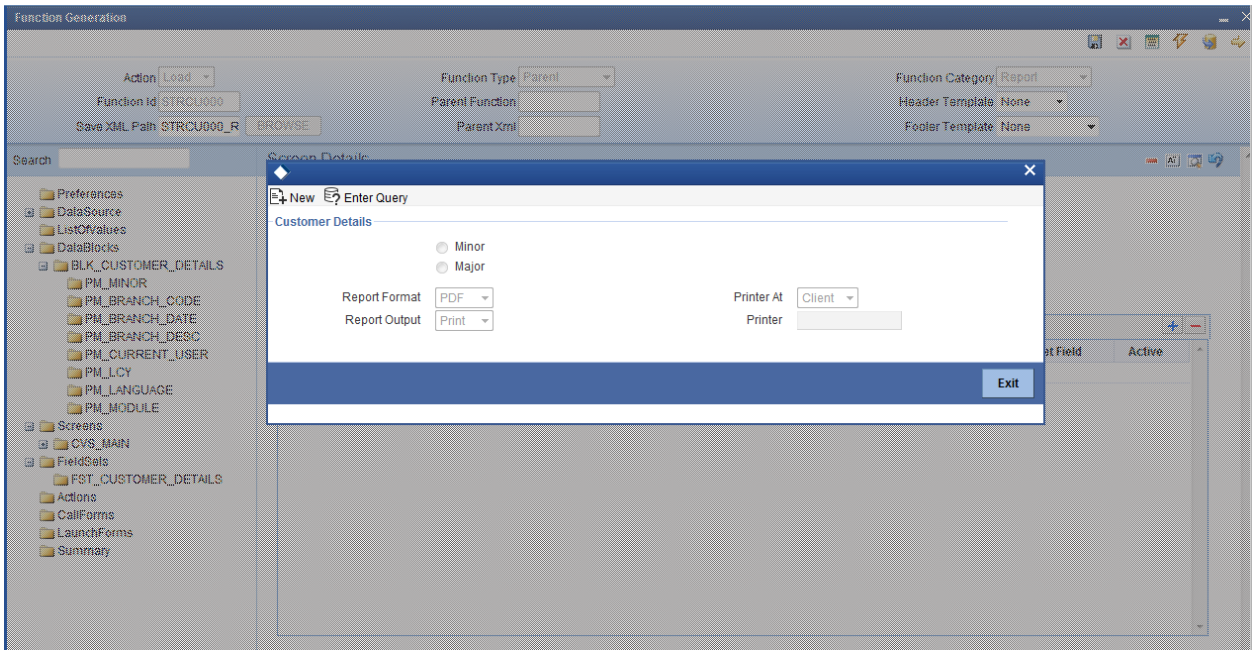


Fig 4.8: Preview of Report Form

8. Save and generate the files:

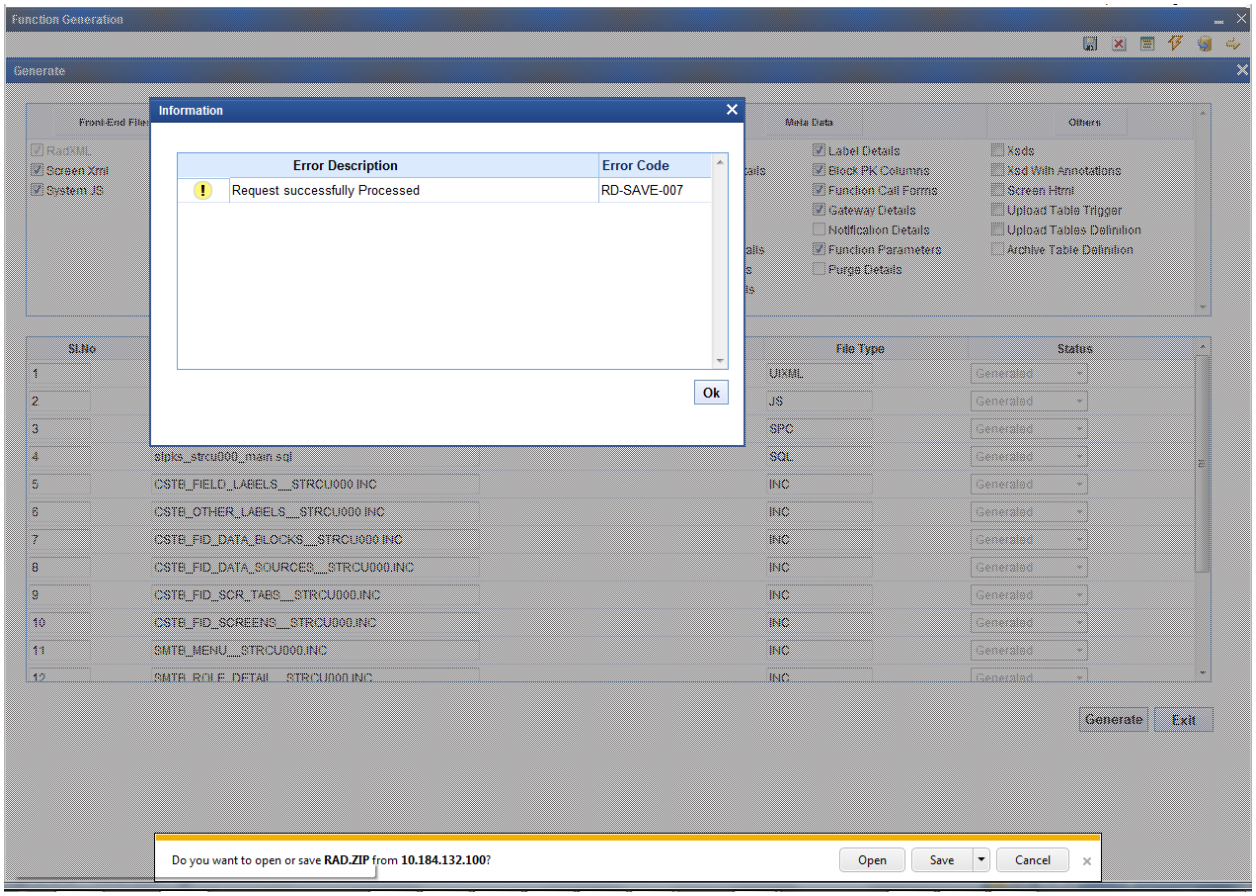


Fig 4.9: Generation of Files

9. Deploy the files

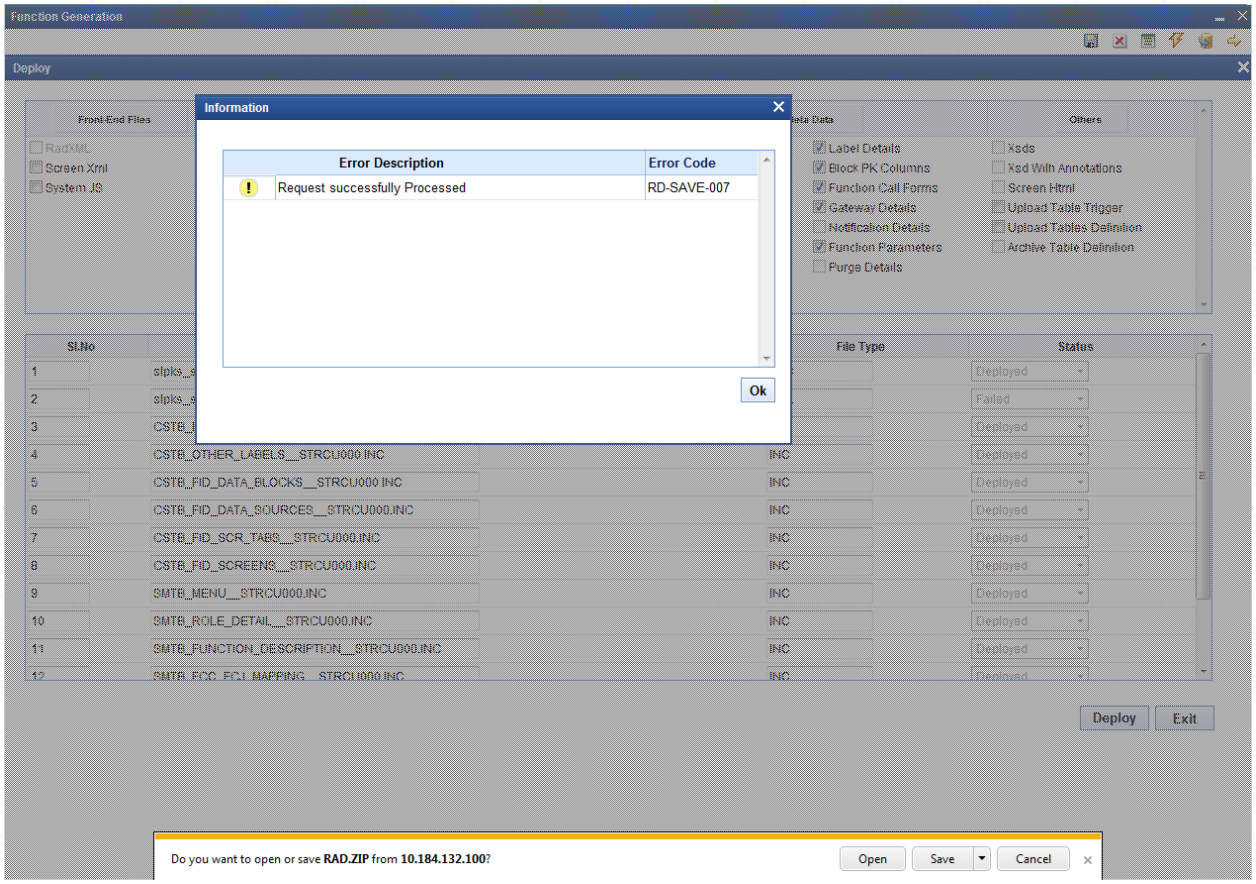


Fig 4.10: Deployed Files

Deploying the Report Screen:

The deployment can be done by clicking on the 'Deploy' button.

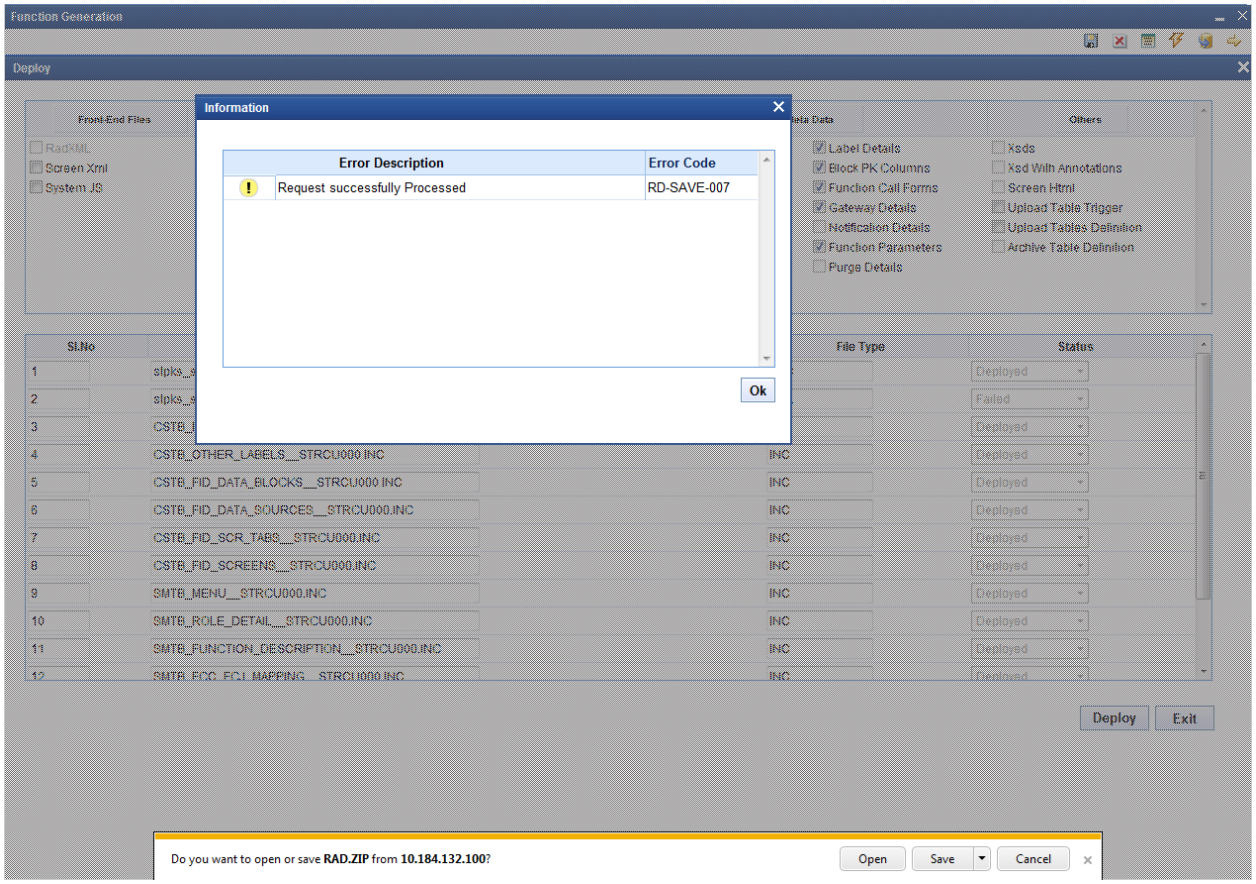
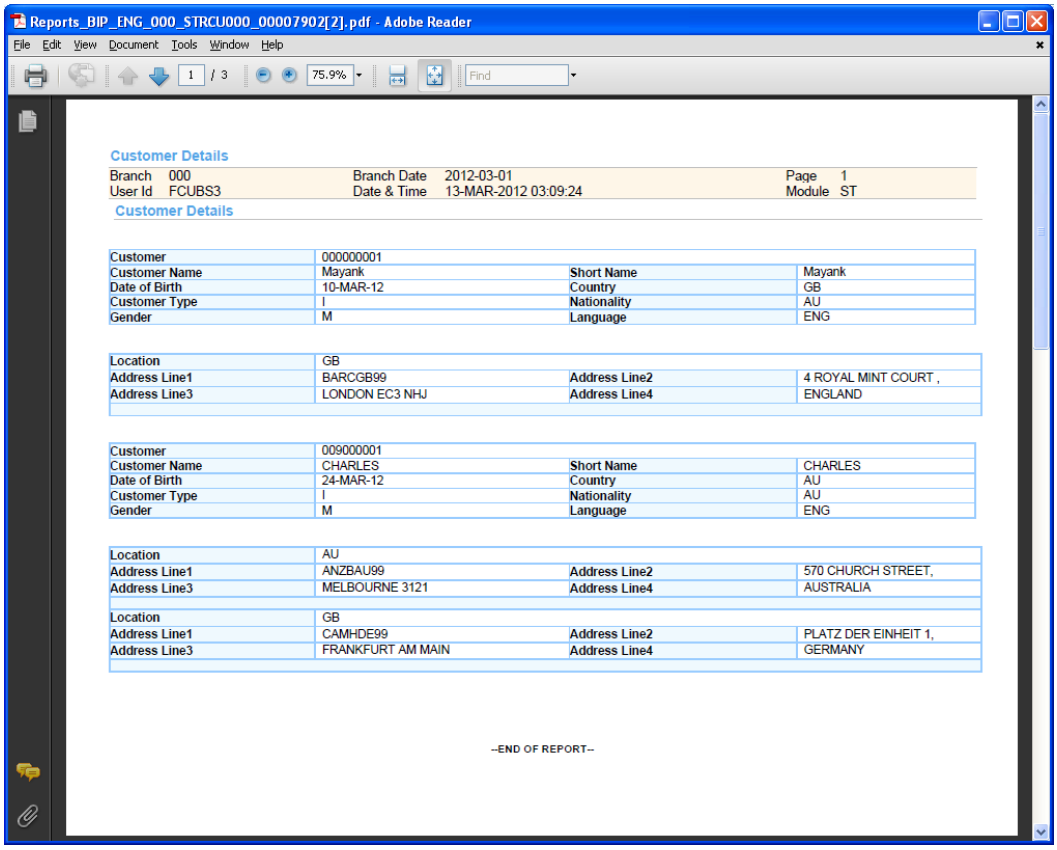


Fig 4.11: Deployment of Screens

A sample report screen:



5. RTF and XDO Files

The rtf file contains the specification for the layout of the report screen.

The xdo file contains the queries which will pick up the relevant values from the back end. The columns to be loaded also should be added in this file.

Both RTF and xdo files needs to be deployed in the BIP server for launching Reports from FLEXCUBE

6. Coding in Hook Package

- Code for deriving the bind values in xdo query has to be written in Hook package and the same has to be called on loading the xdo .Usually the function **Afterpform** is written in Hook package for this purpose.
- Any functions used in the xdo query is usually written in the Hook package .

Development of Report Forms
August 2013

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/