Child and Screen Childs - Concept and Design
Oracle FLEXCUBE Universal Banking
Release 12.4.0.0.0

Contents

1 Preface

This document describes the concept Child and Screen Child screens available in Oracle FLEXCUBE Development Workbench for Universal Banking and guides the developers on how to design child and screen child screens

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
Working knowledge of Web based applications	Self Acquired
Working knowledge of Oracle Database	Oracle Documentations
Working knowledge of PLSQL developer	Respective vendor documents
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.docx
05-Development_WorkBench _Screen_Development-II.docx
14-Development_of_Online_Forms.docx

2 Introduction

This document provides information on:

- Chapter 2 , "Introduction"
- Chapter 3 , "Child Screen"
- Chapter 4 , "Screen Child Screen"

3 Child Screen

Screens required in FLEXCUBE where the base functionality is same and the differences are either in the layout or in some additional processing i.e. there are some set of screens where majority of the functionality is same with some variance with existing screens.

Development Workbench supports developing such screens as the child screens of the base function with a facility to upgrade the child screens whenever the base or parent screen undergoes a change.

Example: Term Deposit Account Booking

Account for term deposit booking will have all the features of a normal customer account with some additional features. Thus it can be designed as the child screen of normal customer account maintenance

Workbench provides an option to design a child screen as a derivative of a parent screen. Workbench automatically inherits the parent screen and tracks the modifications made in the child screen.

Workbench does not allow the developer to delete elements inherited from the parent screen.

For child screens, in addition to release specific hooks for each logical stage, System also calls the corresponding hook from the parent function's programs. ***This allows the re-use of the common business logic among all the child functions.***

FLEXCUBE does not support multi level inheritance (i.e.child of child is not supported)

3.1 Screen Development

For developing a New Child screen,

- Select the action as New
- **Function Type** as Child
- **Parent Xml** field gets enabled
  Click on browse button and select the required parent function form the hard disk. On successful loading parent function will be get update with parent function id value.
- **Parent function** field will get defaulted to the name of the function id which has been loaded as the parent
- User has to enter Child function Id in **function Id** field. Standard naming conventions apply
- **Function category** will be defaulted with category of the parent. It cannot be modified

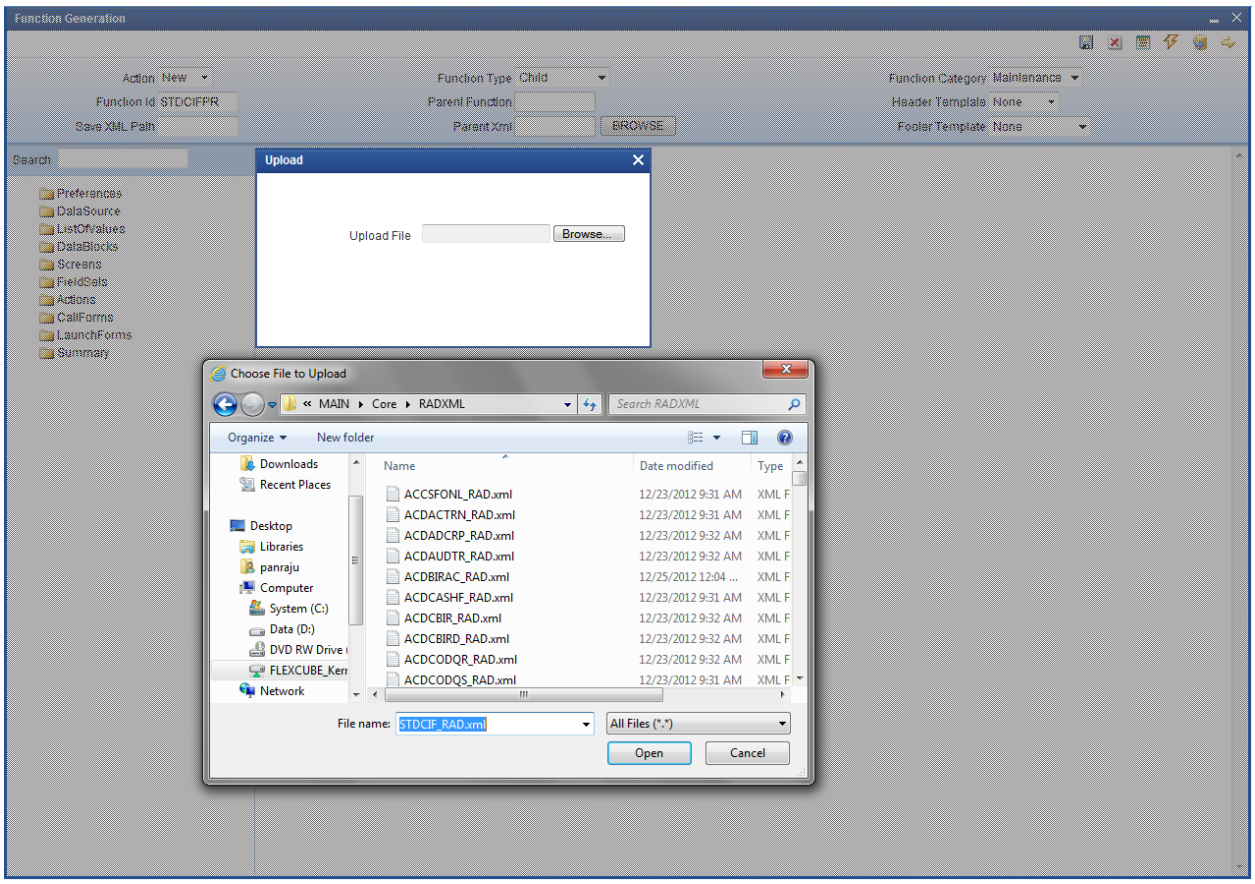

Fig 3.1: Creating Child Screen: Selecting Parent

Preferences:

Make sure the function id name in the preferences screen is that of the child screen and not of parent screen

On tab out of Preferences node function name gets changed automatically if not changed manually. Hence developer has to visit the Preferences node at least once before generating files for child function Id.

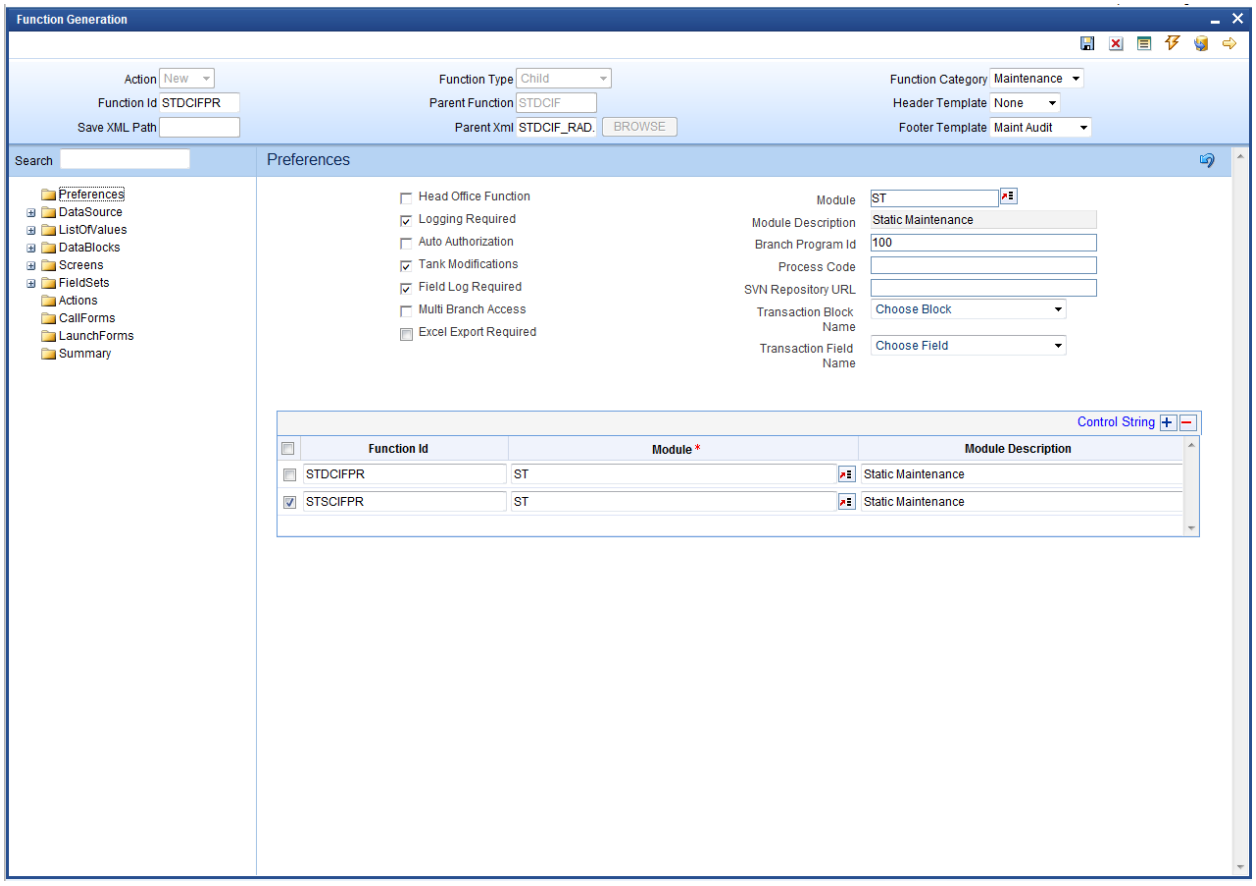

Fig 3.2: Preferences node of Child Screen

Child Screen Design is similar to normal Screen design. Refer *04-Development_WorkBench _Screen_Development-I.docx* for detailed explanation.

Note the following while designing a child screen.

- **Developer will not be allowed to delete or rename elements created in Parent function.**
- New data source or adding columns to existing data sources are allowed
- Addition of new LOVs and modification of existing LOVs are allowed
- Modifying properties of existing block fields(of parent) is allowed
- New Screens, tabs, section etc can be added by the developer
- Deletion and Renaming of screens, tabs, section etc are not allowed. Instead developer has the option to hide them
- Addition or removal of fields from field set is allowed. Properties of fields can also be modified. Note that the block to which field set is attached cannot be changed in child level
- New Field sets can be defined

- Developer can add new Launch Form and Call Form in the child screen. Existing launch Form/Call form can be made inactive, if not required in the child level.
- Amendable fields can be modified
- Summary screen can also be completely re designed in the child level

Note: If enhancement on parent screen happens parallel to child screen design developer can use child refresh option to upgrade latest changes in parent screen to child screen.

3.2 Generation of Files

Process of generation of files is similar to that of normal function id .
All the units generated for a normal function id will be generated for the child screen as well

In the script for SMTB_MENU, column PARENT_FUNCTION would contain the name of the parent function id

3.3 Extensible Development

Developer can add his code in hook packages and release specific JavaScript file.
Extensible Development process is similar to that followed in a maintenance or transaction screen. Refer respective documents for detailed explanation
Structure of the system packages will be same as for a normal maintenance or transaction function id

For child screens, in addition to release specific hooks for each logical stage, System also calls the corresponding hook from the parent function's programs. This allows the re-use of the common business logic among all the child functions.

Assume STDCUSAC is a normal maintenance screen and STDCUSTD as the child of this screen.

Below figure shows the snippet of code from ***fn_default_and_validate*** of the child main package

```
IF NOT stpks_stdcustd_Main.Fn_Skip_Master THEN
    Pr_Convert_Child_To_Master(p_stdcustd,l_STDCUSAC);
    Pr_Convert_Child_To_Master(p_Prev_stdcustd,l_Prev_STDCUSAC);
    Pr_Convert_Child_To_Master(p_Wrk_stdcustd,l_Wrk_STDCUSAC);
    Dbg('Calling stpks_stdcusac_Kernel.Fn_Pre_Default_And_Validate ');
    IF NOT stpks_stdcusac_Kernel.Fn_Pre_Default_And_Validate (p_Source,
        p_Source_Operation,
        p_Function_Id,
        p_Action_Code,
        p_function_Id  ,
        l_STDCUSAC,
        l_Prev_STDCUSAC,
        l_Wrk_STDCUSAC,
        p_Err_Code,
        p_Err_Params) THEN
        Dbg('Failed in stpks_stdcusac_Kernel.Fn_Pre_Default_And_Validate ');
        RETURN FALSE;
    END IF;
    Pr_Convert_Master_To_Child(l_Prev_STDCUSAC,p_Prev_stdcustd);
    Pr_Convert_Master_To_Child(l_Wrk_STDCUSAC,p_Wrk_stdcustd);
END IF;
```

Fig 3.3: Snippet from stpks_stdcustd_main.sql

Here u can find that the developer written defaulting and validation for the parent screen is being called before doing validations specific to child screen . Hence business logic for parent is re used. Developer has the option to skip doing parent validation for the child screens using skip handlers if he wishes so.

Note the functions ***Pr_convert_child_to_master*** and ***Pr_convert_master_to_child***. These procedures have to be used every time a call is made to the parent function id packages. This procedure prepares object types for the parent and child respectively

4 Screen Child

Screen child can be derived from a parent screen or a child screen. Screen child uses the packages for maintenance Functions from the parent itself. Only screen level modifications can be done in a Screen Child

It should be noted that Screen Child are not exactly the child of child screens (second level inheritance). **It can be the child of a parent or a child screen. Difference between child and screen child is that only screen level changes can be done in a screen Child. Business logic will remain the same as its parent.**

In child screen business logic can also be enhanced for the child function id.

4.1 Screen Development

For developing a New Screen Child screen,

- Select the action as New

- **Function Type** as Screen Child
- **Parent Xml** field gets enabled
  Click on browse button and select the required parent function form the hard disk. On successful loading parent function will be get update with parent function id value. **Parent function can be either a parent function id or a child function id**
- **Parent function** field will get defaulted to the name of the function id which has been loaded as the parent function
- User has to enter Screen Child function Id in **function Id** field. Standard naming conventions apply
- **Function category** will be defaulted with category of the parent. It cannot be modified

Fig 4.1: Screen Child Option to be selected

Only 4 nodes of the parent can be modified in a Screen Child

- Data Blocks, Screens ,Field Sets and Actions

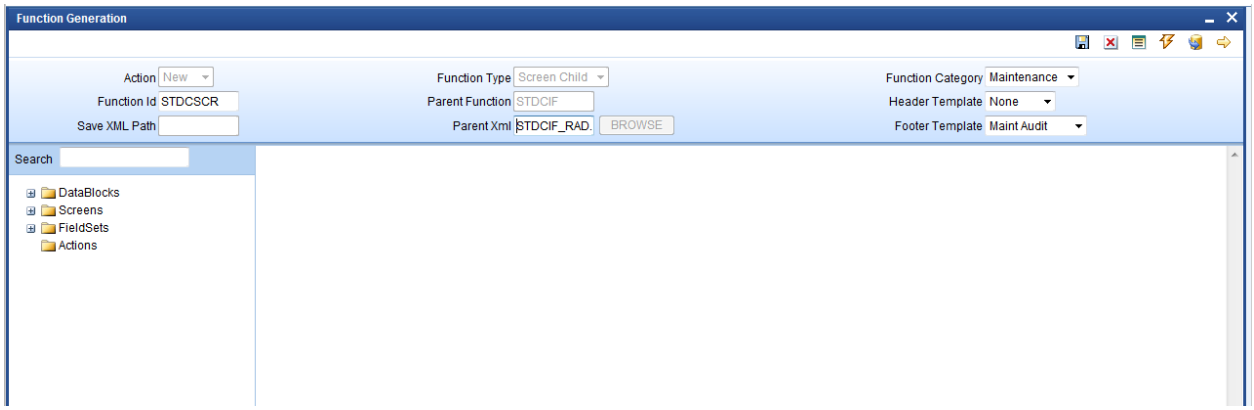

Fig 4.2: The Tree Available for Screen Child

Below are the possible operations that can be done on a screen child in Oracle FLEXCUBE Development Workbench

- Existing elements cannot be deleted
- New blocks or block fields cannot be added
- Only certain properties of the fields in the block can be modified
- Properties of Screens and Field sets can be modified
- New screens and Field Sets can be added
- Tabs, Sections and partitions can be added to the existing screens.
- Web Service and Operation can be configured for the Screen Child, but amendable fields cannot be modified.

Field level properties that can be modified are

- Field Label
- Field Size
- Maximum Decimals
- Default Value
- Preview Value
- Visible
- Read Only
- Calendar Text
- Popup Edit Required
- Uppercase Only
- Input by LOV only
- Not Required in XSD

Screen level modification can be done according to the requirement. Most of the properties at the screen level can be modified

Field set level modification also can be done as per requirement .Note that the block to which field set is attached cannot be changed at screen child for existing field sets. However all properties can be modified for new field sets.

Note: If parent screen enhancements are happening parallel to Screen Child design developer can use Screen child refresh option to upgrade latest parent changes to screen child .

4.2 Generated Files

Only Screen Xml (i.e. language specific xml) . Menu details, Label Details, XML Schema Definitions and Screen Html will be generated from the Workbench for Screen-Child Screen.

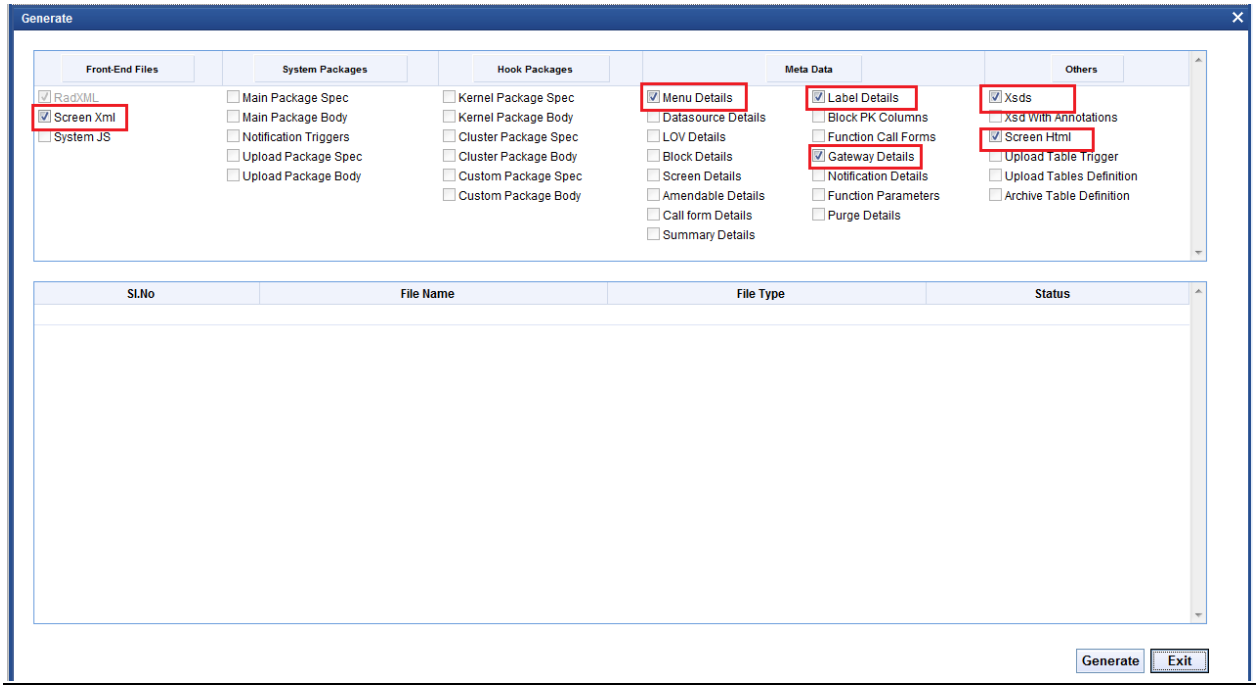

Fig 4.3: Generated Files

4.3 Extensible Development

Screen child uses the packages and JavaScript files of parent itself.

Child and Screen Childs - Concept and Design
[May] [2017]
Version 12.4.0.0.0

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/