

Oracle® Banking Payments

Extensibility Getting Started



Release 14.8.0.0.0
G32396-01
April 2025

ORACLE®

Oracle Banking Payments Extensibility Getting Started, Release 14.8.0.0.0

G32396-01

Copyright © 2017, 2025, 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

1	Preface	
	Purpose	1-1
	Audience	1-1
	Documentation Accessibility	1-1
	Critical Patches	1-1
	Diversity and Inclusion	1-1
	Conventions	1-2
2	Introduction	
	How to use this Guide	2-1
3	Extensibility Introduction	
	What is extensibility	3-1
	Industry Pattern	3-1
	Industry Approach	3-1
4	Oracle Banking Payments Extensibility	
	Business Areas	4-1
	Oracle Banking Payments Extensibility approach	4-1
	Oracle Banking Payments Extensibility User Roles	4-2
5	Oracle Banking Payments Extensible features	
	Screen changes	5-1
	New Screens	5-1
	Screen Modifications	5-1
	Amend field level attributes	5-1
	Style Sheet changes	5-1
	Language conversion of screens	5-2
	Functional	5-2
	User Defined fields at Maintenance	5-2

Processing Logic	5-2
Additional validation logic for a field or group of fields	5-2
Modify defaulting logic for fields	5-3
Online Enquire Screens of Contract extensibility	5-3

6 Extensibility development life cycle

Define Extensibility Requirement	6-1
Identify the Business area of extensibility	6-1
Identify the tools/framework to be used	6-2
Identify the file types & layers applicable	6-2
Develop changes	6-2
Test it in Oracle Banking Payments environment	6-3

Index

List of Figures

6-1 Stages

6-1

List of Tables

1-1 User Roles

1-1

1

Preface

Purpose

This guide is designed to help acquaint you with the Oracle Banking Payments application. This guide provides answers to specific features and procedures that the user need to be aware of the module to function successfully.

Audience

This manual is intended for the following User/User Roles:

Table 1-1 User Roles

Role	Function
Implementation & IT Staff	Implementation & Maintenance of the Software

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>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at [Critical Patches, Security Alerts and Bulletins](#). All critical patches should be applied in a timely manner to make sure effective security, as strongly recommended by [Oracle Software Security Assurance](#).

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

2

Introduction

How to use this Guide

The information in this guide includes:

- [Introduction](#)
- [Extensibility Introduction](#)
- [Oracle Banking Payments Extensibility](#)
- [Oracle Banking Payments Extensible features](#)
- [Extensibility development life cycle](#)

3

Extensibility Introduction

What is extensibility

Extensibility is an ability of the software system to allow and accept the significant extension of its capabilities without major rewriting of code or changes in its basic architecture. Extensible systems provide technology, tools, languages that designed so that developers can expand or add to its capabilities.

Industry Pattern

Following are the industry pattern to address the extensibility in software architecture

- Frameworks
- Configuration files
- Extension using scripts
- User specific extension software packages
- Object based programming where inheritance is used for extensibility

Industry Approach

Industry approaches to extensibility typically includes following:

- Tools to allow to extend the functionality of base product
- Program hooks to allow developers to insert their program routines
- Ability define new business events to address change in process
- Ability to create regional specific software changes
- Ability to add/remove fields at business messages

4

Oracle Banking Payments Extensibility

Business Areas

One of the primary goals of the Oracle Banking Payments architecture is that system should be able to be extendable in required business specific areas. Following are such areas where extensibility is required:

Business Area	Why extensibility required
Screen changes	User may want to keep some screens simple to improve training & operational efficiency
Language of Screens	User may wish to provide screens other than the default language of software
Business / legal requirements	Certain processing/calculation logic may be applied specific to region/country judiciary
Events & eco system participation	The software has to be part of bigger eco system by providing other integration/ notification mechanism
User configurable messages /reports	Software should provide mechanism to extract required information. System to be open to provide the same
Ad hoc exchange of information between systems	System should provide mechanism to exchange information with ad-hoc systems over the period time

Oracle Banking Payments Extensibility approach

Oracle Banking Payments provides the following approach to address the extensible requirement:

Pattern	Industry Approach	Oracle Banking Payments Approach
Framework	Tools and framework to extend the base product	RAD framework
Configuration files	XML files / Text files to configure	XML configuration files and Text based configuration files
Extension using Scripts	Scripts	Java Script based extensions to enable extension at user interface layer
User specific extension of software packag	Program hooks to allow extension logic call outs	Oracle Banking Payments call outs based on release type CUSTOM, CLUSTER Banking

Pattern	Industry Approach	Oracle Banking Payments Approach
User defined events	Ability to define new events/ message types	messaging architecture to define new XML message
Protocol tweaking	Configuration of protocol without software change	types Protocol tweaking Configuration of protocol without software change Oracle Banking Payments ISO8583 protocol definition in XML file that can be modifiable.
User/Regional specific processing logic	Ability to extend the core processing logic	Oracle Banking Payments UDF extensions

Oracle Banking Payments Extensibility User Roles

Oracle Banking Payments Extensibility development can be classified into 4 types based on the complexity and user competencies required:

- Application maintenance/definition of components**
 User expected to login into Oracle Banking Payments application and use certain function IDs to define the new components. This is typically applicable to Bank business user who requires new functionality.

 Example, user need to use function ID UDDMAINT to define new UDF field
- Configuration files**
 User expected to modify some of the parameters in configuration files. This may require restart of application or relevant applications. Typically this is required for application administrators in bank.

 Example, user may need to modify the ISO8583 protocol definition
- Tools based development**
 User expected to understand the given function ID working and required to extend the functionality by adding new data sources and fields. This is typically required by IT developer in bank.

 Example: User needs to change screen layout, to add new data blocks based on new tables added in database.
- Programming**
 User expected to achieve granular control and validations using programming extensions. User expected to know the language used thoroughly in this context. This is typically required by advanced developers in bank.

 Example, bank required to modify the defaulting and validation or modify the processing flow at specific call out points.

Developer role and extensible approach matrix

Given below matrix depicts the developer role and possible extensible approaches to apply:

Developer role	Maintenance/Definition	Configuration	Tools	Programming
Implementer Implementer could be OFSS staff or customer / partner staff who implements Oracle Banking Payments	Yes	Yes	Yes	Yes
Bank Application User Application users are the bank Oracle Banking Payments functional users	Yes			
Bank IT User Bank IT user could be system administrators and have technical skill to extend the Oracle Banking Payments		Yes	Yes	Yes

5

Oracle Banking Payments Extensible features

This section describes the extensible features available in Oracle Banking Payments.

Screen changes

This section describes features that are specific to Function ID (screens) extensibility. RAD tool is used for function ID extensibility.

New Screens

RAD tool used to develop the new screens depending upon the bank requirement. The screens are based on existing or new tables added in database.

Screen Modifications

Existing screens layouts can be modified using RAD tool to suite as follows:

- Hide fields that are not relevant to a given implementation
- Modify the placement of the fields (example moving from one tab to other tab)
- Add LOV to a given field
- Changing the data type
- Adding enumerations to a given field to restrict user inputs
- To increase the set fields (example adding the address line 5)

Amend field level attributes

Existing file level attributes can be modified to add below:

- Defaulting some value to reduce user input/errors.
- Restricting the maximum and minimum value
- Precision settings

Style Sheet changes

Oracle Banking Payments provides style editor to enable CSS changes to have following user specific UI elements design:

- Page template changes
- Dialog template changes
- Form elements look and feel
- Text fonts
- Tables look and feel

- Colors changes

Language conversion of screens

Oracle Banking Payments screens can be extended to support languages other than English.

Functional

User Defined fields at Maintenance

UDF framework enables the bank user to add the new field without changing any table structure. This is used in maintenance function IDs where new field required by bank user.

Processing Logic

Additional validation logic for a field or group of fields

Oracle Banking Payments provides the extension call outs in database layer. These extension call outs are extensible package and pre-named procedures to be used for extensibility. The base product will call this call outs during runtime with required PLSQL data type as parameters.

Example:

User wanted extends STDCIFCR function to add capital letter validation for the field "Customer name". This can be achieved as follows:

Edit the STPKS_STDCIFCR_CUSTOME.Fn_Pre_Default_Validate as below

```
FUNCTION Fn_Pre_Default_And_Validate
(p_Source IN VARCHAR2,
p_Source_Operation IN VARCHAR2,
p_Function_Id IN VARCHAR2,
p_Action_Code IN VARCHAR2,
p_Child_Function IN VARCHAR2,
p_stdcifcr IN stpks_stdCIFCR_Main.ty_stdCIFCR,
p_Prev_stdCIFCR IN OUT stpks_stdCIFCR_Main.ty_stdCIFCR,
p_Wrk_stdCIFCR IN OUT stpks_stdCIFCR_Main.ty_stdCIFCR,
p_Err_Code IN OUT VARCHAR2,
p_Err_Params IN OUT VARCHAR2)
RETURN BOOLEAN IS
BEGIN
  Dbg('In Fn_Pre_Default_And_Validate..');

  --extensibility code start
  p_Wrk_stdCIFCR := p_stdCIFCR;
  IF p_wrk_stdCIFCR.v_sttms_customer.CUSTOMER_NAME1 NOT IN
  (upper(p_wrk_stdCIFCR.v_sttms_customer.CUSTOMER_NAME1))
  THEN
    p_err_code := 'ST-OTHR-097';
    p_err_params := NULL;
    Dbg('Out of validation code-Sarva');
    RETURN FALSE;
  END IF;
```

```
--extensibility code ends
Dbg('Returning Success From fn_pre_default_and_validate..');

RETURN TRUE;
EXCEPTION
WHEN OTHERS THEN
Debug.Pr_Debug('**',
'In When Others of stpks_stdCIFCR_Custom.Fn_Pre_Default_And_Validate
..');
Debug.Pr_Debug('**', SQLERRM);
p_Err_Code := 'ST-OTHR-001';
p_Err_Params := NULL;
RETURN FALSE;
END Fn_Pre_Default_And_Validate;
```

 Note:

Open RAD XML for a given function ID using RAD tool to understand the data block and field name. This would give above complete path to access the field name. you can prefix “p_” to get function ID data type and “v_” to data block to get data block name.

Example: to know the card holder name element at runtime, use following template:

```
[function_id type].[data block name].[field name]
p_wrk_stdCIFCR.v_sttms_customer.CUSTOMER_NAME1
```

Modify defaulting logic for fields

Oracle Banking Payments call outs allows to change defaulting logic for elements using PLSQL data types.

 Note:

Refer example given in section 5.3.1 to know how to identify the element name

Online Enquire Screens of Contract extensibility

Online Enquire Screens of Contract extensibility



Note:

How to identify package name ?

Refer the RAD generated packages for CUSTOM and CLUSTER types to know the possible call outs available which has PLSQL data type parameters. To arrive at the package name using following template.

Template:

<Module code>PKS_<Function ID>_<Release type>

Example:

To get the CUSTOM release of function ID PADOVIEW which belongs to PA module, package would be

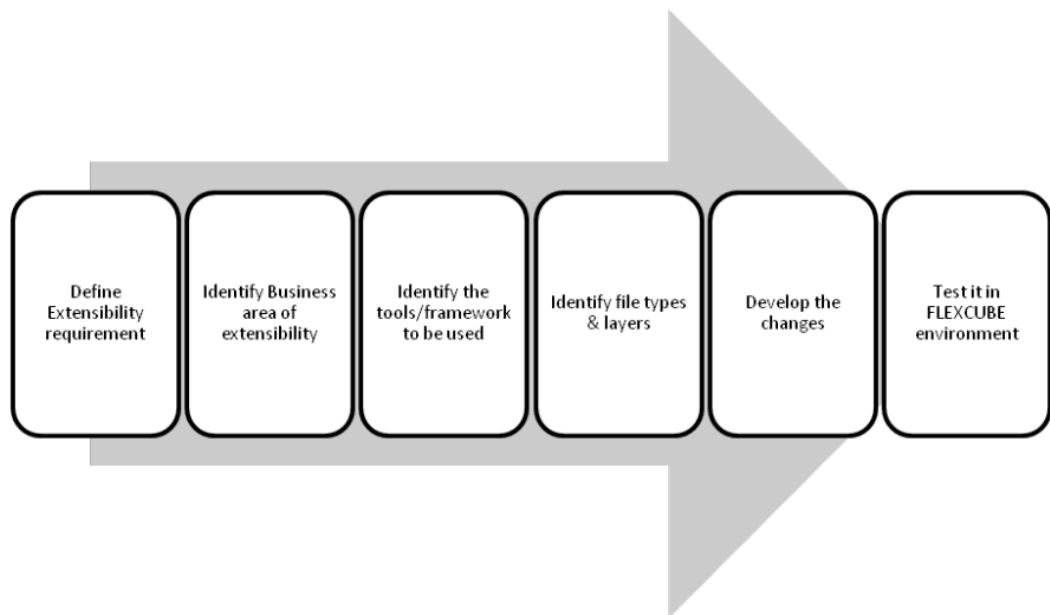
PAPKS_PADOVIEW_CUSTOM

6

Extensibility development life cycle

Extensibility development involves following stages. These stages are explained in detail further down the line.

Figure 6-1 Stages



Define Extensibility Requirement

Extensibility Requirements need to be clearly defined and documented. This requirement should describe the module, function ID (if applicable) and intended functionality required. This requirement should have justification of why extensibility needed compared with base functionality. It also should cover other alternatives to achieve the functionality without extensibility.

Identify the Business area of extensibility

Depending upon the Requirement, user needs to identify the Oracle Banking Payments business area that requires extensibility development. This includes:

- Function ID (New, modify existing, add fields, hide fields)
- Processing logic (defaulting , enriching, validating)
- UDF (New UDF fields for identified function IDs)
- Accounting Interface

- Batch (New batch function during EOD time or intraday)
- Report (new report or modify existing report query)
- Interface (New incoming or outgoing)

Identify the tools/framework to be used

Area	Oracle Banking Payments Tools/Framework
Function ID	• RAD • Style sheet editor
Processing logic	• PLSQL programming on RAD generated packages • PLSQL programming on core packages
Reports	• BIP report development
Interface	• Generic Interface framework

Identify the file types & layers applicable

The below table described the layer and file types developed for each extensibility business areas that involves software modification.

Area	Client Layer	Application Layer	Database Layer
Function ID	Java script files	UIXML	RAD generated CUSTOM/CLUSTER packages
Processing logic	-	-	RAD generated CUSTOM/CLUSTER packages Core ORACLE Banking PAYMENTS Packages
Reports	-	RTF file	RAD generated Report packages
Interface	NA	NA	NA

Develop changes

User can develop the required changes using respective tools documents.

Test it in Oracle Banking Payments environment

User need to copy the developed files to target environment and can test the developed functionality. Refer the Oracle Banking Payments installation manuals on how to deploy the changes.

Glossary

Index