# Oracle® Banking Payments

## Multi-byte Character Support

Release 14.7.0.0.0
G26150-02
February 2025

Oracle Banking Payments Multi-byte Character Support, Release 14.7.0.0.0

G26150-02

# Contents

# 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. |
| *italic* | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| `monospace` | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

# 2 Introduction

Oracle Banking Payments Solution supports multi-byte characters by means of setting session-level character length semantics during Payments installation using the NLS_LENGTH_SEMANTICS parameter of Oracle Database. Most of the application code is written without explicit length semantics (BYTE or CHAR) and, hence, adapts to the session-level semantics during compilation at the time of Payments installation.



## Background

NLS_LENGTH_SEMANTICS parameter allows you to specify the length of a column datatype/ PLSQL variable in terms of characters (CHAR) instead of the default BYTE. Typically, you would need this if you were to deal with (in PL/SQL code) and store data that contains multi-byte characters such as Japanese or Mandarin characters.

Oracle strongly recommends that you do not set NLS_LENGTH_SEMANTICS parameter on instance level (i.e. in the server parameter file) but only use it to alter session-level semantics as necessary. From MOS note 144808.1:

“Oracle advices to use explicit CHAR semantics in the SQL or PL/SQL syntax OR To make sure your application does an "ALTER SESSION SET NLS_LENGTH_SEMANTICS=CHAR;" when connecting if CHAR semantics is required but the semantic is not defined explicit in SQL.”

Based on this note and other available documentation for the parameter, we recommend an approach in the next section which should be followed in sites where multi-byte character support is required.

**Figure 2-1 STOP**


SR with oracle mentioned below is feasible. (SR 3-16364463401)

## Approach

In situations where PL/SQL code would interact with multi-byte character data and the same kind of data would be stored in Payments tables, follow below approach:

Please note the approach is suggested for 12c Database version only.

1. Complete the database installation and set up as per the Oracle Banking Payments Database Best Practices for your Payments release.
2. Immediately after creating the Payments application user (schema), create a logon trigger as the SYS user to alter each new session so that the session-level length semantics are

maintained as CHAR for the user at all times. For example, if the schema name was PAYMENTS and the trigger was named “PAYMENTS_LOGON”, the trigger would look like: CREATE OR REPLACE TRIGGER SYS.PAYMENTS_LOGON AFTER LOGON ON PAYMENTS.SCHEMA BEGIN EXECUTE IMMEDIATE 'alter session set nls_length_semantics=CHAR'; END;/

3. Proceed with the rest of the Payments installation as per the installation documents. In situations where an operation results in Payments objects being invalidated, make sure the objects are re-compiled either:
4. by the PAYMENTS schema user using one of the following options:
   - Manually using ALTER… COMPILE statements
   - SYS.UTL_RECOMP.RECOMP_PARALLEL procedure
   - SYS.UTL_RECOMP.RECOMP_SERIAL procedure

   Or,
5. by a user with appropriate privileges on SYS.UTL_RECOMP package (such as the SYS user) using one of the following options only:
   - SYS.UTL_RECOMP.RECOMP_PARALLEL procedure
   - SYS.UTL_RECOMP.RECOMP_SERIAL procedure

   Oracle Banking Payments strongly advises against manually compiling PAYMENTS schema objects (using ALTER…COMPILE) as a non-Payments application user when length semantics of CHAR are being enforced. Such an attempt may change the NLS_LENGTH_SEMANTICS parameter setting of the PL/SQL objects to BYTE and this may result in runtime exceptions within the application, such as: “ORA-06502- PL/SQL: numeric or value error: character string buffer too small”.

**Figure 2-2 STOP**


If the PAYMENTS schema is ever re-compiled using SYS.DBMS_UTILITY.COMPILE_SCHEMA in an environment that is using CHAR semantics, then it must be ensured that for the user executing the procedure, the session-level NLS_LENGTH_SEMANTICS parameter is set to CHAR. This is because while SYS.UTL_RECOMP.x procedures re-use existing PL/SQL object settings, SYS.DBMS_UTILITY.COMPILE_SCHEMA compiles the objects with the NLS settings of the invoking user’s session and may alter the PL/SQL object settings of the objects (including length semantics).