

Oracle ACE

Tech Superstars Unite

Get worldwide recognition as an Oracle ACE



Oracle.com profile page



Swag, certification exam credit & event passes



Exclusive content



Networking events



Your own Oracle cloud account



Travel support



Learn more at:
ace.oracle.com

 [@oracleace](https://twitter.com/oracleace)

 [Linkedin.com/groups/1796302](https://www.linkedin.com/groups/1796302)

 [@oracleace.bsky.social](https://bsky.app/profile/oracleace.bsky.social)



ORACLE

What Else in APEX 26.1?

Menno Hoogendijk

Consulting Member of Technical Staff

Oracle APEX

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Agenda

Interactive Report
Improvements

Developer Usability
Improvements

Other New Features

Interactive Report Enhancements

Enhancement 1/6: Declarative Row Selection

This feature allows developers to enable row selection in Interactive Reports without requiring manual checkbox implementation or JavaScript workarounds.

☰ New Features

🔍 fsecara ▾

Declarative Row Selection

Standard IR - Multi Selection Standard IR - Single Selection Standard IR - Hide Control IR Detail View - Multi Selection

Detail View Multi Selection
100288:100291

🔍 Search



Actions ▾



JSB

Johann Sebastian Bach

Gender: Male, Born: 1685, Died: 1750

30.167197



LVB

Ludwig van Beethoven

Gender: Male, Born: 1770, Died: 1827

30.109331



CVO

Cleopatra VII of Egypt

Gender: Female, Born: -69, Died: -30

30.036677



GG

Galileo Galilei

Gender: Male, Born: 1564, Died: 1642

29.96003

Enhancement 2/6: Invoke Interactive Report Dialog

This feature allows developers to invoke specific Interactive Report action dialogs such as Select Columns, Download, Chart, etc through dynamic actions, enabling custom button placement and greater UI flexibility beyond the standard IR Actions menu.

The screenshot displays a web application interface with a table of data and an open 'Chart' dialog. The table has columns: Id, Name, Email, Initials, Managerno, Longitude, Latitude, and Avatar Image. The 'Chart' dialog is centered over the table and contains the following elements:

- Chart Type Selection:** Four icons representing different chart types: Bar, Line with Area, Pie, and Line.
- Label:** A dropdown menu with '- Select Column -' and an adjacent text input field for 'Axis Title for Label'.
- Value:** A dropdown menu with '- Select Column -' and an adjacent text input field for 'Axis Title for Value'.
- Function:** A dropdown menu with '- Select Function -'.
- Orientation:** A dropdown menu with 'Vertical' selected.
- Sort:** A dropdown menu with 'Default' selected.
- Buttons:** 'Cancel' and 'Apply' buttons at the bottom right of the dialog.

Enhancement 3/6: Maximum Rows to Display

This feature allows developers to set a maximum number of rows displayed to users after all Interactive Report filters, sorting or other settings are applied, ensuring users see the most relevant results based on their interactions.

Preferred over Maximum Rows to Process.

The screenshot shows the Oracle APEX App Builder interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Data Reporter', and 'Gallery'. The current page is 'Page 8: Maximum Rows to Display'. The left sidebar shows the component tree with 'Maximum Rows To Display' selected under 'Body'. The right pane shows the configuration for this component, with the 'Maximum Rows to Display' property set to 50. Other visible properties include 'Type' (Row Ranges X to Y), 'Display Position' (Bottom - Right), 'Performance' (Lazy Loading), and 'Messages' (When No Data Found).

Enhancement 4/6: CSS Class to IR Columns

This feature allows developers to assign a CSS classes directly to Interactive Report columns, providing precise styling control and consistent formatting without requiring HTML expression or workarounds.

The screenshot displays the Oracle APEX App Builder interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Data Reporter', and 'Gallery'. The current page is 'Application 103 \ Page 9: Background Column Color'. The left sidebar shows a tree view of components, with 'Column Background' > 'Columns' > 'START_DATE' selected. The main panel shows the 'Column' configuration for the selected column. The 'Appearance' section is expanded, showing 'CSS Classes' set to 'u-color-18'. The 'Accessibility' section has 'Value Identifies Row' disabled. The 'Column Formatting' section is also visible.

Enhancement 5/6: Filtering and Sorting for Template Components

This feature allows developers to enable sorting and filtering on Template Component columns in Interactive Report, removing previous restrictions and providing full control over column functionality.

☰ New Features 🔍 fsecara ▾

Declarative Dynamic Actions

Show All ☐ Menu Button ☐ Content Row ☐ **IR Badges** ☐ Cards

▾ ☐ 🔍 Status = 'On-Hold' × ☑ 🔍 Status = 'Open' ×

Task Name	End Date	Start Date	Status ↑	Budget
Arrange for vacation coverage	14-SEP-2039	16-JUL-2039	↑ ↓ ✕	\$7,000.00
Customize Software Projects software	04-OCT-2039	29-AUG-2040	🔍 Filter...	\$1,000.00
Investigate new Virus Protection software	27-SEP-2039	12-SEP-2039	Closed	\$1,500.00
Get RFPs for new server	17-SEP-2039	27-AUG-2039	On-Hold	\$1,000.00
Collect mission-critical spreadsheets	30-OCT-2039	29-AUG-2039	Open	\$4,000.00
Create applications from spreadsheets	12-FEB-2040	29-AUG-2039	Pending	\$10,000.00
Lock spreadsheets	12-FEB-2040	29-AUG-2039	! Open	\$800.00
Send links to previous spreadsheet owners	14-FEB-2040	30-AUG-2039	! Open	\$1,500.00
Customize solutions	14-NOV-2039	22-AUG-2039	! Open	\$4,000.00

Enhancement 6/6: Unlimited Row Output

This feature removes the 32kb per-row limit output limitation in Interactive Report, enabling display of complex reports with numerous columns, large text values and extensive templating, note that this is not available for Pivot.

☰ New Features 🔍 fsecara ▾

Unlimited Row Output

Lots of Columns Lots of Rows

🔍 Search Actions ▾

Cc	C001	C002	C003	C004	C005	C006	C007	C008	C009	C010	C011	C012	C013	C014	C015	C016	C017	C018	C019	C020	C021	C022	C023	C024	C025	C026	C027	C028
0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	2	2	2	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
0	3	3	3	0	0	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
0	4	4	4	0	1	1	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
0	5	5	5	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
0	6	6	6	1	1	2	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
0	7	7	7	1	1	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	
0	8	8	8	1	2	2	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	
0	9	9	9	1	2	3	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	
0	10	10	10	2	2	3	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	1
0	11	11	11	2	2	3	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	1
0	12	12	12	2	3	4	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	1



Demo:

Interactive Report Enhancements



Developer Usability Improvements

Trigger Actions

This feature empowers developers to build robust client-side functionality with row context access.

Simplified development by allowing direct, **declarative** attachment of dynamic actions, eliminating the need for custom JavaScript.

Consistent functionality across Template Component Actions, Menu Buttons, and standard page components.



Declarative Menus and Trigger Actions

The screenshot displays the Oracle APEX App Builder interface for configuring a declarative dynamic action. The top navigation bar includes 'APEX App Builder', 'SQL Workshop', 'Data Reporter', and 'Gallery'. The current page is 'Application 103 \ Page 4: Declarative Dynamic Actions'. The left sidebar shows a tree view with 'Change Status' expanded to 'Menus', which includes 'Open Project', 'Close Project', 'On-Hold Project', and 'Triggered Actions'. Under 'Triggered Actions', 'Show Error Message' is selected. The main workspace shows a 'Layout' view of a page with regions like 'REGION CONTENT', 'SUB REGIONS', 'NEXT', 'Badges', 'SORT ORDER', 'PREVIOUS', 'REGION BODY', and 'REGION CONTENT'. A 'Regions' palette at the bottom provides various UI components. The right-hand 'Trigger Action' configuration panel is active, showing a 'Settings' section with a message: 'A task with the status &STATUS. cannot be updated, &STATUS. already in use.' and a 'Clear Errors' toggle. The 'Execution' section shows a 'Sequence' of 40. The 'Client-side Condition' section is set to 'JavaScript expression' with the code: `$v('STATUS') === 'On-Hold'`. The 'Server-side Condition' section is currently set to '- Select -'.



Declarative Menus and Trigger Actions

New Features

1 error has occurred
• A task with the status On-Hold cannot be updated, On-Hold already in use.

Declarative Dynamic Actions

Menu Button **Content Row** Badges Cards

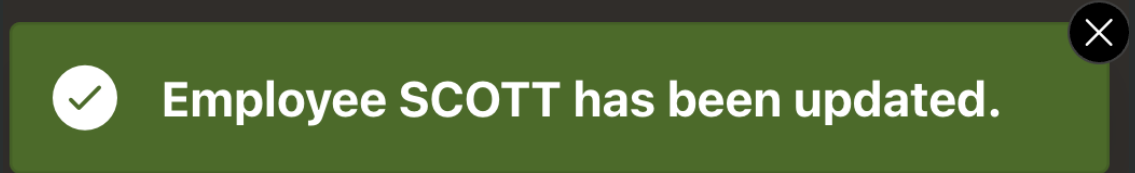
- Task **Arrange for vacation coverage**
Project: Maintain Support Systems
On-Hold
- Task **Investigate new Virus Protection software**
Project: Maintain Support Systems
On-Hold Project
- Task **Identify point solutions required**
Project: Load Software
Closed
- Task **Install in development**
Project: Load Software
Closed
- Task **Customize solutions**
Project: Load Software
Open

Task

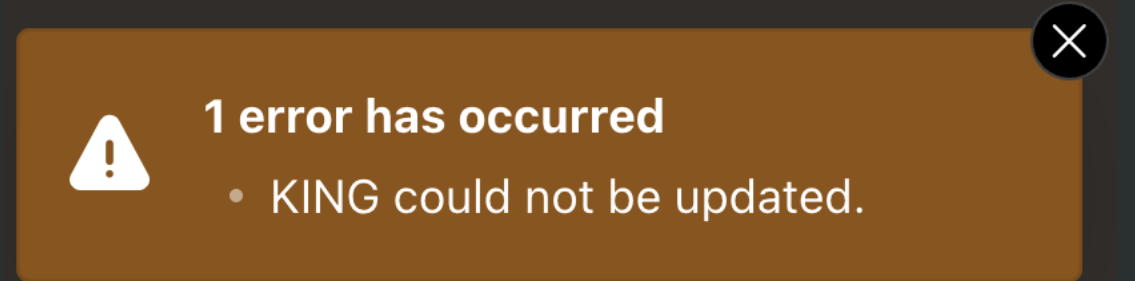
MAIN App 103 Page 4 Session Debug Quick Edit Customize

New Dynamic Actions

- Show Success Message
- Show Error Message
- Clear Errors



A green notification bar with a white checkmark icon on the left and a close button (X) on the right. The text inside reads: "Employee SCOTT has been updated."



An orange notification bar with a white warning triangle icon on the left and a close button (X) on the right. The text inside reads: "1 error has occurred" followed by a bulleted list item: "• KING could not be updated."

Declarative Spinner

A new **Show Processing** switch in Dynamic Actions makes it easy to display a spinner for long-running PL/SQL code.

No custom JavaScript needed!

This brings consistent and user-friendly feedback to server-side actions, streamlining both development and the end-user experience.

The screenshot shows the 'Trigger Action' configuration interface. It includes a search filter, an 'Identification' section with 'Name' (Execute Code) and 'Action' (Execute Serv), and a 'Settings' section with 'Language' (PL/SQL). Below these is a 'PL/SQL Code' section containing a PL/SQL block. At the bottom, there are fields for 'Items to Submit' and 'Items to Return', and a 'Show Processing' toggle switch which is currently turned on and highlighted with a red box.

```
begin
  for l_emp ( select empno
              from emp
              where deptno =
:P2_DEPTNO )
  loop
    if needs_salary_raise(
l_empno.empno ) then
      update emp
      set sal = sal * 1.03
      where empno =
```



Running Modal Dialog Pages

Run Directly from Page Designer:

- Modal Dialog Pages
- Page 0

✓ **Changes saved. Dialog pages cannot be run directly from Page Designer. Review this page by running a page that launches this dialog.** ✕



Sample Data

- Enables **quick component creation** by using built-in sample data without needing to define a data source.
- Supports **tailored sample data per component type** to ensure appropriate visuals, with flexibility for plug-in developers to define their own demo queries.
- Helps developers **build UI mockups faster** and independently of backend data availability.
- Lets UI developers focus on layout while data source developers define actual queries or REST sources later.

Create Classic Report

Page Definition

Page Number: 10

Name: [Text Input]

Page Mode: Normal | Modal Dialog | Drawer

Include Form Page: [Toggle Off]

Data Source

Data Source: Sample Data

Navigation

Use Breadcrumb: [Toggle On]

Breadcrumb Parent Entry: [Dropdown]

[Back] [Cancel] [Create Page]

Demo: Developer Usability Improvements





Other New Features

Message Based App Translations

APEX now features a powerful, **text message–based translation system**! Developers can easily **seed, export, and import** translatable text strings.

Enabling quick setup for multilingual apps without the need for shadow applications.

Effortlessly manage translations in the APEX Builder or via XLIFF/CSV files, so delivering applications in multiple languages is simpler and more flexible than ever before!

Text Message Translations	
	Manage Text Messages Create and manage text messages for your application.
	Convert to Text Messages Convert and update text across your application into Text Messages.
	Export Text Messages Download text messages in XLIFF or CSV format for translations.
	Import Text Messages Upload translated text messages in XLIFF or CSV format to your application.



Content Security Policy (CSP) Enhancements

APEX 26.1 makes significant progress toward stronger Content Security Policy (CSP) support, reducing the need for `unsafe-inline` and `unsafe-hashes` across modern components.

A new class-based approach replaces inline styles for dynamic behaviour, helping applications work with stricter CSP configurations.

These improvements enhance the security posture of APEX applications and support adoption of modern CSP standards, while ongoing validation and guidance continue to expand compatibility across APEX features.

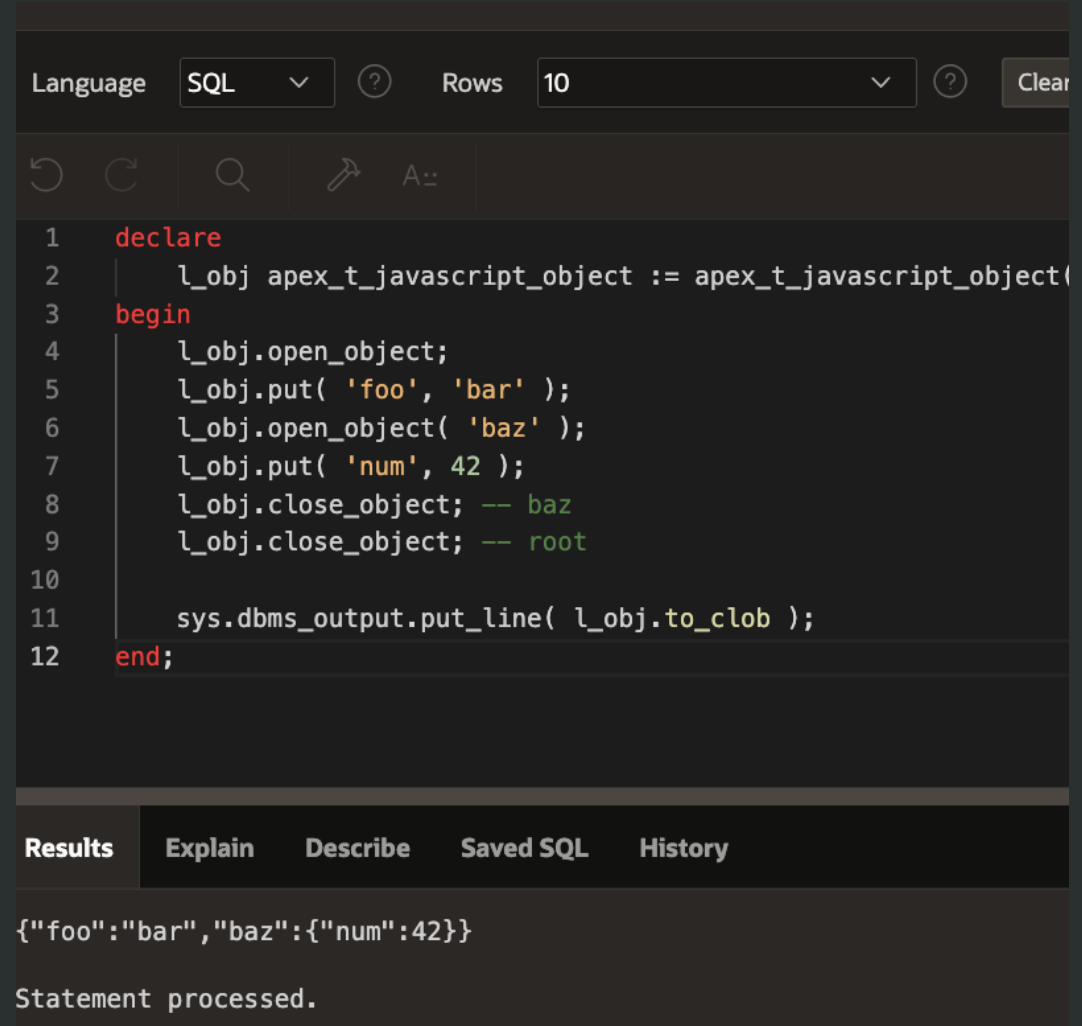
26ai Boolean Support

APEX now fully supports the Boolean SQL datatype (Oracle DB 26ai+), enabling Boolean columns, values, and processing in UI, session state, and APIs.

APEX_T_JAVASCRIPT_OBJECT

The **APEX_T_JAVASCRIPT_OBJECT** Public API is new public object type for constructing JavaScript objects with various properties to be passed directly to the browser with methods like:

- **open_object/close_object**
- **open_array/close_array**
- **put**
- **append**
- **put_function**



The screenshot shows the Oracle SQL Developer interface. At the top, the 'Language' is set to 'SQL' and 'Rows' is set to '10'. Below the toolbar, a PL/SQL script is displayed with line numbers 1 through 12. The script declares a local object, opens it, puts 'foo' and 'bar', opens an array 'baz', puts 'num' and 42, closes the array, and finally outputs the object's JSON representation. Below the script, the 'Results' tab is active, showing the output: {"foo":"bar","baz":{"num":42}} and the message 'Statement processed.'

```
1 declare
2     l_obj apex_t_javascript_object := apex_t_javascript_object(
3 begin
4     l_obj.open_object;
5     l_obj.put( 'foo', 'bar' );
6     l_obj.open_object( 'baz' );
7     l_obj.put( 'num', 42 );
8     l_obj.close_object; -- baz
9     l_obj.close_object; -- root
10
11     sys.dbms_output.put_line( l_obj.to_clob );
12 end;
```

Results Explain Describe Saved SQL History

```
{"foo":"bar","baz":{"num":42}}
```

Statement processed.

Demo: Other New Features



Thank you

Menno Hoogendijk