

# Tree Select Plugin for Oracle APEX

User Manual Developed by MOHAMMAD ALQURAN

Version 1.0 – November 2025

## 1. Introduction

The **Tree Select Plugin for Oracle APEX** lets you display a hierarchical tree with checkboxes inside a region. End-users can expand and collapse branches, check or uncheck items, search by label, and have their selection stored in a page item for use in reports, charts, and processes.

This manual is aimed at APEX developers who want to add the plugin to their applications, configure it, and understand how to integrate it with other components using Dynamic Actions and CSS.

## 2. Installing the Plugin

1. In APEX, go to **Shared Components** → **Plug-ins** and import the plugin export file for *Tree Select Plugin for Oracle APEX*.
2. Upload the JavaScript (`xx_tree_select.js`) and CSS (`xx_tree_select.css`) files as **Static Application Files**.
3. In the plugin definition, ensure the JavaScript File URL and CSS File URL attributes point to the plugin files (for example `#PLUGIN_FILES#xx_tree_select.js` and `#PLUGIN_FILES#xx_tree_select.css`), so they are automatically included on pages that use the plugin.

After installation, the plugin becomes available as a new region type in the region type selector.

## 3. Creating a Tree Select Region

1. Create or open a page where you want to use the tree.
2. Add a new region and choose **Tree Select Plugin for Oracle APEX** as the region type.
3. Give the region a clear **Static ID** (for example, `empTreeRegion`).
4. Configure the data source attributes:
  - o **Static JSON** — paste a JSON array of nodes with `id`, `parentId` and `label`.
  - o **SQL / Table** — supply a table or SQL query that returns `id`, `parent_id`, and `label`.
5. Set the **Store Item** attribute to the name of a page item (e.g. `P10_SELECTED_NODES`) where the selected ids will be stored.

## 4. Configuring Plugin Attributes

In the plugin region attributes, you'll find options like:

- **Allow Parent Selection** — can parent nodes be checked?
- **Allow Leaf Selection** — can leaf nodes be checked?
- **Tri-State** — should parents show a mixed (indeterminate) state when children differ?
- **Default Expand Level** — how many levels to open initially (e.g. 1 or 2).
- **Search Mode** — `NONE`, `CLIENT`, or `SERVER`.
- **Store Format** — `CSV` or `JSON`.

- **Visual Options** — compact rows, show icons, dark theme, etc.

Adjust these attributes to match your functional and visual requirements. Most behavioral differences (e.g. tri-state vs simple checkboxes, client vs server search) are controlled through these settings.

## 5. Using the Toolbar and Search

The plugin can render a toolbar and search field, depending on your settings:

- **Search** — filters nodes by their labels.
- **Expand all / Collapse all** — open or close all branches.
- **Check all / Uncheck all** — toggle all enabled nodes in the current view. Holding **SHIFT** while clicking Check All can be configured to load all children first and then apply the operation.

Users can also click individual toggles to expand a single branch, and check or uncheck a single row using the checkbox.

## 6. Understanding Selection Behavior

Each tree node can be in one of three states (when tri-state is enabled):

- **Checked** — the node is selected.
- **Unchecked** — the node is not selected.
- **Mixed (indeterminate)** — some children are checked and some are not.

The plugin stores the current selection in the page item configured as the **Store Item**:

- If the **Store Format** is `csv`, the item contains a comma-separated list of selected node ids (e.g. `10,20,30`).
- If the format is `JSON`, the item contains a JSON array of ids (e.g. `["10", "20", "30"]`).

You can use this item in report filters, processes, validations, or any server-side logic that needs to know which nodes are selected.

## 7. Using Dynamic Actions

Dynamic Actions (DAs) make the tree interactive with the rest of your page. Common patterns:

- **DA on item change** — create a DA that fires when the store item value changes (e.g. `P10_SELECTED_NODES`), and have it refresh a report region or chart.
- **DA on custom event** — if you have JavaScript code that triggers a custom event when the selection changes, you can define a DA that responds to that event on the tree region.
- **DA on region refresh** — after the tree region is refreshed (for example, after a submit), you can re-run your setup or perform other logic.

Combined with the tree, these patterns enable faceted search: users choose categories in the tree, and your reports automatically refresh to show matching rows.

## 8. Styling and Customizing the Look

The Tree Select plugin uses its own CSS file to define default styling, but you remain in control of the final appearance:

- Adjust colors to match your application theme.
- Change row spacing or font size to create a compact or relaxed layout.
- Customize connector lines between nodes.
- Add accent classes to mark important nodes with colored bars or icons.

To override styles, add your CSS either in the page CSS or in a shared application CSS file. Make sure your overrides load *after* the plugin CSS so they take effect.

## 9. Example Use Cases

Here are some practical ways to use the Tree Select Plugin:

- **Faceted search** — let users choose multiple categories or tags, then filter reports on the selected values.
- **Security / role assignment** — show privileges in a hierarchy and store the selected ones for a user or role.
- **Organizational hierarchies** — show departments and teams, and allow managers to select entire branches.
- **Product catalogs** — let users explore product categories and choose subsets for analysis or export.

## 10. Troubleshooting

### Tree does not appear

- Give the plugin region a Static ID (for example `empTreeRegion`) so you can target it from custom CSS or Dynamic Actions. The plugin does not require a specific Static ID, but each region should have a unique one.
- Verify that the JavaScript and CSS files are referenced correctly as plugin files (for example via `#PLUGIN_FILES#` URLs).
- Confirm that your data source (JSON or SQL) returns rows.

### Search and wildcards in Client vs Server mode

- In SERVER mode, search uses SQL `LIKE` semantics on the server side, and in CLIENT mode the plugin emulates the same behavior in JavaScript. Wildcards `%` (any sequence of characters) and `_` (single character) are supported consistently in both modes, so users can rely on the same pattern rules regardless of configuration.

### Selection is not stored in the item

- Check that the Store Item attribute points to an existing page item.
- Make sure the item is not protected or reset unintentionally by page processes.

## 11. Compatibility and Upgrade Notes

The Tree Select Plugin is designed for current Oracle APEX versions using Universal Theme (APEX 19.2 and newer). When you upgrade APEX:

- Retest pages that use the plugin, especially AJAX and search.
- Check for any changes in Universal Theme that affect spacing or colors.
- If necessary, update the plugin's PL/SQL code to use new attribute helper APIs (such as `apex_plugin_util.get_varchar2_attribute`) for long-term compatibility.

## Features and Capabilities Overview

### Tree Functionality

- Full tri-state checkbox tree integrated with Oracle APEX page items and Dynamic Actions.
- Parent / child recompute logic that respects disabled nodes and supports mixed states.
- Dynamic expand and collapse per node, plus toolbar buttons to expand/collapse all.

- Client and Server search modes, both supporting SQL-style wildcards (% and \_).
- Shift-click style bulk operations (Check All / Uncheck All) for fast selection changes.
- Optimized behavior for large data sets with optional lazy loading of child nodes.

## User Interface & Styling

- Toolbar with expand, collapse, check, uncheck, and search controls.
- Flexible layout that supports faceted-style pages (search above, controls on the side of the tree).
- Per-node icons using the ICON column, fully themeable via CSS.
- Universal Theme-friendly colors and typography so the tree blends with the rest of the application.
- Accessible markup using role="tree", aria-expanded, and aria-hidden where appropriate.
- Self-contained dialog behavior that avoids interfering with the page's scroll or global overlays.

## Integration & Configuration

- Implemented as a declarative APEX region plugin, configured entirely from the builder.
- Supports the plugin's custom Dynamic Actions such as "Tree Select - Changed", "Tree Select - Search", and "Tree Select - Loaded".
- Works with multiple data source types (SQL Query, SQL Table, Static JSON, and REST-based sources via APEX).
- Designed for modern APEX 21+ with Universal Theme, and can be configured to work with older APEX versions when needed.
- No external JavaScript libraries required; the plugin ships with its own JS and CSS bundle.

In short, the Tree Select Plugin provides a rich, maintainable, and APEX-native way to present and manage hierarchical data with checkboxes, while remaining flexible enough to adapt to many different use cases (filtering reports, faceted search, navigation trees, and more).

## 12. Future Enhancements

Possible enhancements that may appear in future versions of the plugin include:

- **Expand / Collapse State Memory** — remember which branches were expanded between sessions and restore them automatically.
- **Improved Accessibility & Keyboard Navigation** — additional ARIA refinements and keyboard shortcuts.
- **Optional Drag-and-Drop Reordering** — interactively re-order nodes with server persistence.

Features such as lazy loading for large trees and per-node icons (from the ICON column) are already available in the current version.

## 13. Credits

The Tree Select Plugin for Oracle APEX and this documentation were developed by **MOHAMMAD ALQURAN**. It is intended as a reusable building block for rich hierarchical selection interfaces within APEX applications.

**Tree Select Plugin for Oracle APEX**Developed by **Mohammad Alquran**Contact: [moh.alquraan@gmail.com](mailto:moh.alquraan@gmail.com)[Donate](#)[Chat on WhatsApp](#)