

Add or disable option in cascading select

On this page

- [Purpose](#)
- [Configuration Parameters](#)
- [Usage Examples](#)
- [Related Features](#)

Purpose

Add or Disable option in Cascading Select post-function allows dynamically **adding new options**, **re-enabling disabled options** and **disabling exiting options** in Cascading Select custom fields. It supports adding/re-enabling/disabling **root options** or **child options**.

Configuration Parameters

This screenshot shows configuration parameters for **Add or Disable option in Cascading Select** post-function:

Cascading select list field: Cascading Select List field that will receive the new option.	Country and City - [Select List (cascading)]	?
Option level: Level of the cascading select field where the new option will be inserted.	<input checked="" type="radio"/> Root option <input type="radio"/> Child option	
Option Name: Source field containing the name of the new option to be inserted.	Ephemeral string 1 - [Text]	
Action: Action to be carried out by the post-function on the option with input name.	<input checked="" type="radio"/> Insert new option, or do nothing if it already exists. <input type="radio"/> Insert new option, or re-enable it if it already exists and is disabled . <input type="radio"/> Insert new option, or insert a duplicated one if it already exists. <input type="radio"/> Insert new option, or insert a duplicated one if it already exists and is disabled . <input type="radio"/> Disable existing option.	
Sort options alphabetically:	<input type="checkbox"/>	If checked, options are sorted alphabetically, otherwise the new option is added at the end of the options list.
Conditional execution: Optional boolean expression that should be satisfied in order to actually execute the post-function. (Syntax Specification)	<pre>1</pre> <p>Leave the field empty for executing the post-function unconditionally. Collection of Examples [Line 1 / Col 1]</p> <p><u>Logical connectives</u>: and, or and not. Alternatively you can also use &, and !. <u>Comparison operators</u>: =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and != can be used with <i>strings</i>, <i>multi-valued fields</i> and <i>lists</i>. <u>Logical literals</u>: true and false. Literal null is used with = and != to check whether a field is initialized, e.g. {00012} != null checks whether <i>Due Date</i> is initialized.</p> <p>String Field Code Injector: Summary - [Text] - %{00000}</p> <p>Numeric/Date Field Code Injector: Original estimate (minutes) - [Number] - {00068}</p> <p>Check Syntax</p>	

Ephemeral string virtual fields are typically used to store the name of the new option, and the name of the root option where the child option is going to be inserted added or re-enabled. These fields are previously set by post-function [Copy parsed text to a field](#).

Note: **Disable option** feature is available since version **2.2.26**.

Usage Examples

Related Features

- [Add or disable option in select list, radio button, multi-select list or checkbox](#)