

Add a Behaviour to a MLCS field

Multi Level Cascade Select field is not officially supported by Adaptavist Script Runner plugin and its component. In case of a Script Runner Behaviour though, it is possible to apply a workaround that will allow to evaluate MLCS values using Behaviours even if the field is not supported.

Here below, we're going to see how to hide fields depending on the value selected of the first level of your MLCS field. The main idea is to trigger, via script, a "change" event on a field supported by Behaviours and to create then a Behaviour that handles everything we want to accomplish.

1) For our purposes, let's define the following fields that must be configured on the same screen the Behaviour is going to run:

- a Multi Level Cascade Select field called "Department" and has an ID of 10100 with three options on the first level
- a text field called "Employee" and has an ID of 11300 (this field will be used for the workaround).
- three Group Picker fields to assign an OnCall Team to the Issue, one for each option of the "Department" field. Depending on the first-level value of "Department", one of these fields will be shown

2) On the form (create / edit / transition screen) you'd like your Behaviour to be ran, choose one of the fields available and in its own Field Configuration (related to the Project and Issue Type desired) add this javascript

```
<script type="text/javascript">
  jQuery(document).ready(function($) {
    JIRA.bind(JIRA.Events.NEW_CONTENT_ADDED, function (e, context) {
      AJS.$(function () {
        //get first-level item of your mlcs field
        var mlcs = document.getElementById("customfield_10100:0");
        mlcs.onchange=function() {
          //when the first-level item changes, trigger a change on Employee field
          AJS.$("#customfield_11300").trigger("change");
        }
      });
    });
  });
</script>
```

3) When picking the MLCS field on the form, the suffix ":0" will represent the selection of the first level (this is easy to check if you go into the "Inspect" mode in your browser, so you can check the HTML of the page)

4) Go in the Behaviour section of the Script Runner plugin and create a new Behaviour adding a new mapping depending on your needs, then enter the Behaviour configuration and add a serverside script to the "Employee" field. The script will be something like this

```
/* ##### "Employee" field's Behaviour ##### */

//hide fields available for every Department selection
getFieldByName("Department 1 On-call team").setHidden(true);
getFieldByName("Department 2 On-call team").setHidden(true);
getFieldByName("Department 3 On-call team").setHidden(true);

//access to first-level MLCS value
FormField mlcs = getFieldById("customfield_11000:0");
String mlcsVal = mlcs.getValue();
log.debug("Department's First-Level value is " + mlcsVal);

//test value selected using optionId and then show desired the field
if("14100".equals(mlcsVal)){
  getFieldByName("Department 1 On-call team").setHidden(false);
} else if("14101".equals(mlcsVal)){
  getFieldByName("Department 2 On-call team").setHidden(false);
} else if("14102".equals(mlcsVal)){
  getFieldByName("Department 3 On-call team").setHidden(false);
}
```

5) Three different fields, each one related to a Department selected on the first level, will be hidden every time the Behaviour is executed (these three lines can also be added as a Initialiser script). Then, the first level of the Department field will be selected, its value will be logged and then the test will begin: depending on which value has been selected, one of the fields will be displayed.

Important: the value to test in the "if" is not the string value of the options but the IDs associated. The IDs can be retrieved using the MLCS configuration page, opening each option (as if you would have some more child options) and then looking at the "parentOptionId" value in the URL