

# Check if all levels are filled in

If you set an MLCS field as mandatory in the Field Configuration Scheme, JIRA will only check if the first level is filled in.

To check if all levels are filled in, you can use the following scripts.

## For JIRA 7.x

```
import org.apache.log4j.Logger
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.issue.CustomFieldManager;
import com.atlassian.jira.issue.fields.CustomField;
import com.opensymphony.workflow.InvalidInputException;

CustomFieldManager customFieldManager = ComponentAccessor.getCustomFieldManager()
//10003 is the ID of the MLCS custom field
CustomField mlcs = customFieldManager.getCustomFieldObject(new Long('11167'))
// log.info("DEBUG MESSAGE: mlcs = $mlcs")
def mlcsValues = issue.getCustomFieldValue(mlcs)
// log.info("DEBUG MESSAGE: mlcsValues = $mlcsValues")
def levels = 0
if (mlcsValues != null){
    mlcsValues.each { i ->
        if (! i.toString().startsWith("com.sourcesense.jira.plugin.customfield.option.
SpecialOptionFactory")){
            levels++
        }
    }
}
// Validate the MLCS field has at least 2 levels populated and throw an exception if not
if (levels < 2){
    InvalidInputException iie = new InvalidInputException("PLEASE NOTE: At least 2 drop-downs are required for
" + mlcs)
    log.info("PLEASE NOTE: At least 2 drop-downs are required for " + mlcs)
    throw iie
}
```

## For JIRA 6.x

```

import org.apache.log4j.Logger
import com.atlassian.jira.issue.MutableIssue
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.issue.customfields.manager.OptionsManager;
import com.atlassian.jira.issue.CustomFieldManager;
import com.atlassian.jira.issue.fields.CustomField;
import com.atlassian.jira.issue.customfields.option.Option;
import com.atlassian.jira.issue.Issue;
import com.google.common.collect.Lists;
import com.atlassian.jira.issue.IssueManager;
import com.atlassian.jira.security.JiraAuthenticationContext;
import java.util.ArrayList;
import com.atlassian.crowd.embedded.api.User;
import com.atlassian.jira.event.type.EventDispatchOption;
import com.opensymphony.workflow.InvalidInputException;

def log = Logger.getInstance("com.onresolve.jira.groovy.PostFunction")

CustomFieldManager customFieldManager = ComponentAccessor.getCustomFieldManager()
//10003 is the ID of the MLCS custom field
CustomField mlcs = customFieldManager.getCustomFieldObject("customfield_10003")
ArrayList<String> mlcsValues = (ArrayList<String>) issue.getCustomFieldValue(mlcs)

if (mlcsValues != null) {

    String[] mlcsValuesList = mlcsValues.toArray()
    // If the MLCS field has 3 levels...
    if (mlcsValuesList.length != 3){
        InvalidInputException iie = new InvalidInputException("Please fill in all levels" + mlcs)
        log.info("Please fill in all levels " + mlcs)
        throw iie
    }
}

```

#### For JIRA 8.x

```

import org.apache.log4j.Logger
import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.issue.CustomFieldManager;
import com.atlassian.jira.issue.fields.CustomField;
import com.opensymphony.workflow.InvalidInputException;

CustomFieldManager customFieldManager = ComponentAccessor.getCustomFieldManager()
//10401 is the ID of the MLCS custom field
CustomField mlcs = customFieldManager.getCustomFieldObject("customfield_10401")
// log.info("DEBUG MESSAGE: mlcs = $mlcs")
def mlcsValues = issue.getCustomFieldValue(mlcs)
// log.info("DEBUG MESSAGE: mlcsValues = $mlcsValues")
def levels = 0
if (mlcsValues != null){
mlcsValues.each { i ->
if (! i.toString().startsWith("com.sourcesense.jira.plugin.customfield.option.SpecialOptionFactory")){
levels++
}
}
}
// Validate the MLCS field has at least 2 levels populated and throw an exception if not
//change the number of levels with the one you need
if (levels < 2){
log.info("PLEASE NOTE: At least 2 drop-downs are required for " + mlcs)
throw invalidInputException = new InvalidInputException("customfield_10401","MLCS is required")
}
}

```

