

Split 3 levels MLCS values into three different fields

```
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;

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

JiraAuthenticationContext jAC = ComponentAccessor.getJiraAuthenticationContext();
IssueManager issueManager = ComponentAccessor.getIssueManager();
User user = jAC.getUser().getDirectoryUser();

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

String[] mlcsValuesList = mlcsValues.toArray()
//10000, 100001, 100002 are the IDs of the fields you want to copy values into
CustomField level1CF = customFieldManager.getCustomFieldObject("customfield_10000")
CustomField level2CF = customFieldManager.getCustomFieldObject("customfield_10001")
CustomField level3CF = customFieldManager.getCustomFieldObject("customfield_10002")
issue.setCustomFieldValue(level1CF, mlcsValuesList[0])
issue.setCustomFieldValue(level2CF, mlcsValuesList[1])
issue.setCustomFieldValue(level3CF, mlcsValuesList[2])
issueManager.updateIssue(user, issue, EventDispatchOption.DO_NOT_DISPATCH, false)
```