Managing Inventories on issue transition - Example 1

It is possible to track quantity of an asset with post functions. Please follow the steps of the following example.

1- Define a Quantity attribute
2- Add it to the desired form
3- Create an asset and set a Quantity value (i.e. 50)
4- Now there is an asset with a Quantity value 50
5- Define the post function to update quantity
   - Post function Groovy Script
6- Test it on Post Function Create Screen and publish workflow
   - Change status of the issue to execute post function.
   - Before: quantity=50
   - After, quantity=49

In this example, we’ll demonstrate to decrease quantity value of an asset which is attached to a Jira issue on Done workflow transition.

1- Define a Quantity attribute

![Edit Attribute](Image)

2- Add it to the desired form

![Edit Form Phone](Image)

3- Create an asset and set a Quantity value (i.e. 50)
4- Now there is an asset with a Quantity value 50

5- Define the post function to update quantity

Select Add "[AIP] - Update Asset workflow postfunction" to the workflows to update an asset object on transition. And select Quantity field. Leave form parameter as blank to set any matching form.
import inventoryplugin.entity.JipInventoryItem;
import org.apache.commons.lang3.StringUtils;

// as field is a Text type we need to convert value to int and return as String
String getNewQuantityValueOfAsset() {
    def stringValue = StringUtils.trim(aipUtils.getAttributeValueAsStringByName(asset, 'Quantity'));
    if (stringValue != null && stringValue.isInteger()) {
        int intValue = stringValue as Integer;
        intValue--;
        return intValue as String;
    } else {
        return stringValue;
    }
}

// "null" result won't update asset. Return '' if you want to clear attribute value
if(assetStatus=='removed') return null;
return getNewQuantityValueOfAsset();

// aipUtils.getAttributeValueAsStringByName returns the Quantity attribute value of the asset.
"return null" result won't update asset. If you want to clear attribute value use "return ''"

6- Test it on Post Function Create Screen and publish workflow
7- Change status of the Issue to execute post function.

In this example post function is configured to for Done transition. Quantity is set from 50 to 49.

Before: quantity=50

After, quantity=49
When the last task is done, the story can be automatically closed >> Drag this task to "Done" too