Conditions: Usage

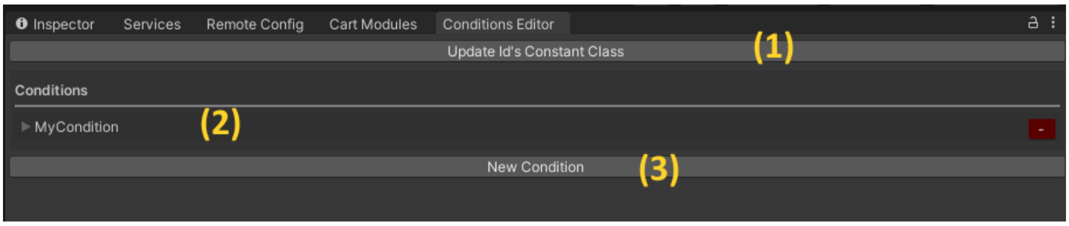
The conditions system is designed to let you define `bool` checks based on runtime criteria. The library has some criteria pre-defined for you to use. You define all the conditions in the editor and then implement them into your code base for use.

Editor window

Conditions are defined and edited in the conditions editor window. This can be found under:

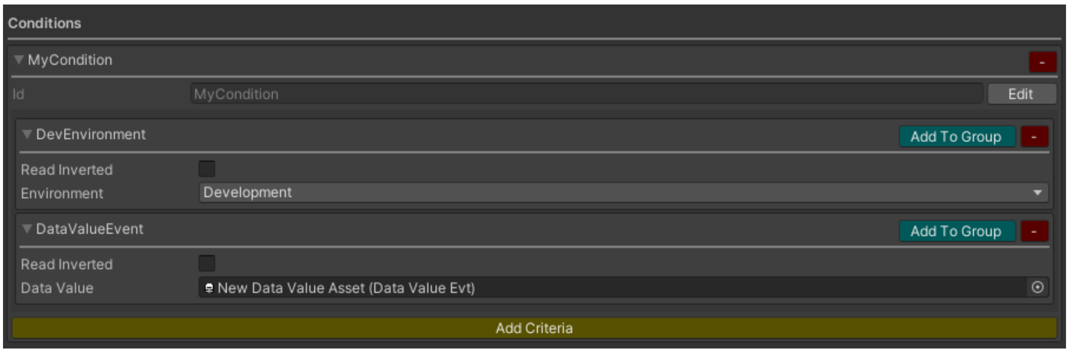
```
Tools/Carter Games/The Cart/Modules/Conditions/Editor
```

From the window you have 3 main sections:



1. Press to update a constants class which is handy for referencing condition id’s without any issues.
2. The area that shows all the conditions you have defined in dropdowns.
3. Press to make a new condition.

Condition Criteria

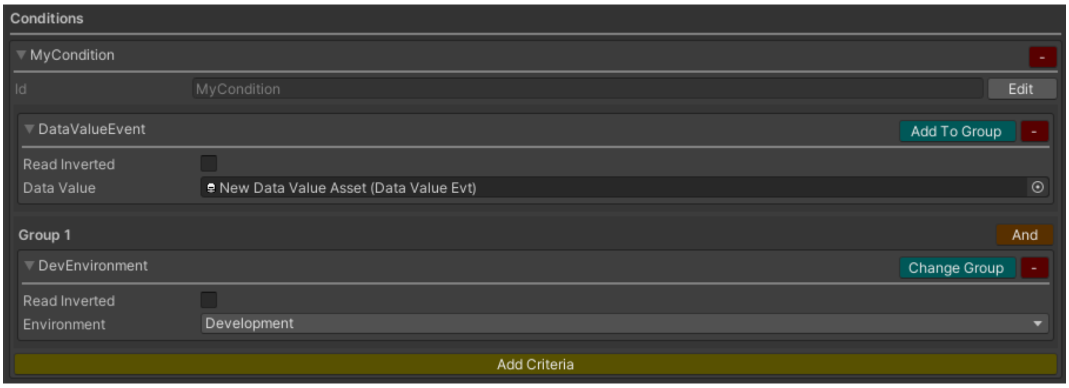


Id: MyCondition

Read Inverted: □
Environment: Development

Read Inverted: □
Data Value: New Data Value Asset (Data Value Evt)

With a condition open you can see all the criteria for that condition and their settings. Each criteria can be read in an inverted state should that be needed. If you want to group criteria together you can press the add group button and select the group you wish to add that criteria to.



Id: MyCondition

Read Inverted: ☐

Data Value: New Data Value Asset (Data Value Evt)

Read Inverted: ☐

Environment: Development

When a criteria is in a group it'll appear in a separate section. Each group can be toggled in its passing check. This can be:

- And
  - All criteria in the group must pass for the group to pass.
- Or
  - Any criteria passing in the group will pass the group.

You can toggle the group passing state method by pressing the orange current state button on the group.

If the criteria is not in a group, it assumes the criteria passing check is an `And` style check.

Listening for a condition

In code, you can listen for any condition you have defined. You use the `ConditionManager` class to do this which is a static class. An example of using the condition defined above:

```
private void OnEnable()
{
    ConditionManager.RegisterListener(ConditionIds.MyCondition, OnConditionChanged
}


private void OnConditionChanged()
{
    if (ConditionManager.IsValid(ConditionIds.MyCondition)
    {
        Debug.Log("My condition was valid.");
    }
}
```