**How to use option in info**

**This document covers new options introduced in the Mod Installer version 20230401-1700.**

There is a way to make a package customizable by a user. You can add as many options as you want.

I take the rope package as an example.

# Info.xml

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<PackageInfo>

  <Games>

<Game>FC5</Game>

</Games>

  <DefaultInclude>false</DefaultInclude>

  <Name>FC4 Grappling Rope</Name>

<Description></Description>

  <Category>8</Category>

  <Options>

    <Option ID="RpColor" name="Rope color">

      <option name="FC4 color" value="1,0.55,0,1" />

      <option name="Red" value="1,0,0,1" />

      <option name="Blue" value="0,0,1,1" />

      <option name="Green" value="0,1,0,1" />

    </Option>

  </Options>

</PackageInfo>

So, first we define **Options** node and then add child **Option** nodes and add **ID** and **name** attributes. Value of the attribute can be anything and it will be displayed in the Mod Installer when you select the package. The **ID** attribute is for usage of the **Option**, because you can have many **Option** nodes.

Then we can add options, the count of **option** nodes is not limited.

The **option** node has two attributes – **name** and **value**. **name** is name of the option and will be displayed in Mod Installer. **value** is the value of the option which will be used, more later.

There is also the possibility of using more than one value of one option (option value index) - for example if you need to change several fields in configs.

Definition is pretty simple. Every next value will have number at end, so:

  <option name="Option 1" value="0" value1="No" value2="False" value3="none" />

As you can see, every value has always +1, there is no limit for values.

Note: if you define more values, you must set same amount of values for every option:

  <Options>

    <Option ID="MyOpt" name="An option">

      <option name="Option 1" value="0" value1="No" value2="False" value3="none" />

      <option name="Option 2" value="1" value1="Yes" value2="True" value3="one" />

    </Option>

  </Options>

## Note: **Option** node can be added also to info\_replace.xml

# How to use the Option

Now there are two places where the Option can be used – first is LUA scripts and second are replaces.

## LUA scripts

Usage is simple:

local selector = \*\*ModInstallerOption\*\*MyOpt\*\*

if selector == 1 then

  self.allowVehicles = true

end

When Mod Installer installing packages, the \*\*ModInstallerOption\*\*MyOpt\*\* will be replaced with select **option**’s node **value.** The OptionID part is replaced with **Option**‘s **ID** attribute.

For more values, usage is simple:

local selector = \*\*ModInstallerOption\*\*MyOpt\*\*2\*\*

if selector == 1 then

  self.allowVehicles = true

end

This means the **value** to variable **selector** will be added from **value** with number **2**.

## Replaces

You add special string to the field value, it looks like this **{ option ID | value index }** - the option index is optional, if not set, default is first (without the number at end).

Example:

    <Replace RequiredFile="graphics\\_materials\grassam-m-20140219095318.material.bin">

      <object hash="2A979110">

        <primaryKey hash="5E237E06">A55B20EE</primaryKey>

        <field hash="1D775834" type="Int32">{MyOpt}</field>

        <field hash="87CA5B02" type="Boolean">{MyOpt|2}</field>

        <field hash="0B6A78CA" type="String">{MyOpt|3}</field>

      </object>

    </Replace>

So, there are used three values, let’s describe how it will work:

* **{MyOpt}** will be replaced with option MyOpt’s first value - **value**
* **{MyOpt|2}** will be replaced with option MyOpt’s second value - **value2**
* **{MyOpt|3}** will be replaced with option MyOpt’s three value - **value3**

You can also use options as part of string:

  <field hash="0B6A78CA" type="String">Some long text with option put here: {MyOpt}</field>

## Using in hash, name and type attributes

Options can be used even in changing hash and name attributes. They are defined as in-string variable:

  <object hash="BB96607D">

   <field hash="{MyOpt}" type="Enum">1</field>

  </object>

  <object name="ManagedObject">

   <field name="{MyOpt}" type="Enum">1</field>

  </object>

  <object hash="BB96607D" name="ManagedObject">

  <field hash="55356C84" name="selRpgLootRollableMask" type="{MyOpt}">1</field>

  </object>

As you can see, the option is defined between { and }, you define just the option ID. There is also the possibility of defining the value index.

  <object hash="BB96607D">

   <field hash="{MyOpt|2}" type="Enum">1</field>

  </object>

It can be used even as partial name:

  <object name="ManagedObject">

   <field name="selRpgLootRollable{MyOpt}" type="Enum">1</field>

  </object>

## Using in field arrays

For more info about field arrays look at **How to use replace package**.

Example of usage for variants of replacing field arrays.

<object hash="2A979110" name="parameter">

<primaryKey hash="5E237E06" name="name" value-ComputeHash32="BaseColorArray" type="BinHex">74C1B63C</primaryKey>

<field hash="1D775834" name="value" type="Vector4">

<Color>{MyOpt},1,1,1</Color>

<Color>2,1,1,{MyOpt}</Color>

</field>

</object>

<object hash="2A979110" name="parameter">

<primaryKey hash="5E237E06" name="name" value-ComputeHash32="TilingArray" type="BinHex">24C77693</primaryKey>

<field hash="1D775834" name="value" type="Vector4" arrayIndex="10">{MyOpt},1000,1000,1000</field>

</object>

## Using in templates IDs

For more info about templates look to **How to use templates in info**.

Options can be used even in template IDs, so you can apply different templates based on selected option’s value.

<object name="TableEntries">

<template id="{MyOpt}" templateValueID="7005422307597406" />

</object>

<object name="TableEntries">

<template id="MyTemplate{MyOpt}" templateValueID="7005422307597406" />

</object>

## Using in templateValue

For this you need to understand **How to use templates in info**.

You can even use Options to set values for templates.

We have defined an *option* with *ID* **MyOpt**, so defining the *templateValue* with option will look like:

  <template id="1" templateValueID="{MyOpt}" />

And also with option value index:

  <template id="1" templateValueID="{MyOpt|2}" />

This means that *templateValue* with name **ID** will receive a selected value from *option* with *ID* **MyOpt**.

So without *option* it looks like:

<template id="MyTemplate" templateValueID="A Value" />

and with *option* with *ID* **MyOpt**:

<template id="MyTemplate" templateValueID="{MyOpt}" />

# Option can be used in info\_replace.xml for selecting different Replaces

If you want to make more values to be changed, you can use *Option* for selecting different *Replace* for one file, as an example package you can take Encounter Spawns.

Format is: **option ID | option selected value | option value index** where the option value index is optional.

If you want to make this, just define *Option* as always, then make more *Replace* for one file (or you can use more files) like this:

  <Replaces>

    <Replace RequiredFile="databases\generic\encounterspawnsettings.ndb" Option="EnemyCount|0">

      <object hash="59F2984F">

. . .

      </object>

    </Replace>

    <Replace RequiredFile="databases\generic\encounterspawnsettings.ndb" Option="EnemyCount|1">

      <object hash="59F2984F">

. . .

      </object>

    </Replace>

    <Replace RequiredFile="databases\generic\encounterspawnsettings.ndb" Option="EnemyCount|1|3">

      <object hash="59F2984F">

. . .

      </object>

    </Replace>

  </Replaces>

Then the *Replace* will be selected on the selected option in Mod Installer.

This Option definition supports special vars. If you put a special character before the attribute value, you receive a special selector based on the selected option’s value.

So, possible characters:

* **!** (this means the Replace will used ONLY if the option’s value does NOT equal the defined value)
* **>** (this means the Replace will used ONLY if the option’s value is BIGGER than the defined value - WARNING - option’s value MUST be number)
* **<** (this means the Replace will used ONLY if the option’s value is SMALLER than the defined value - WARNING - option’s value MUST be number)

And how to use them?

  <Replaces>

    <Replace RequiredFile="databases\generic\encounterspawnsettings.ndb" Option="EnemyCount|!0">

. . .

    </Replace>

  </Replaces>

  <Replaces>

    <Replace RequiredFile="databases\generic\encounterspawnsettings.ndb" Option="EnemyCount|>0">

. . .

    </Replace>

  </Replaces>

  <Replaces>

    <Replace RequiredFile="databases\generic\encounterspawnsettings.ndb" Option="EnemyCount|<0">

. . .

    </Replace>

  </Replaces>

Note: don’t forget XML doesn’t like <> chars in values of attributes, so you must escape it:

* for < use &lt;
* for > use &gt;

# Option for selecting which file will be packed

*Option* can be used also for selecting which file will be packed to the game. It can be used for example for model replacing or textures replacing, simply for files where can’t be used info\_replace.xml or one of methods above.

This can be used only if in info.xml is defined the *Pairs* node. Alternative structure of \*.a3 package doesn’t support this.

Again - same format: **option ID | option selected value | option value index** where the option value index is optional

Here is as an example info.xml from Gamescom Demo Loading Screen (abbreviated):

  <Option name="Resolution of video">

    <option name="720p" value="720p" />

    <option name="1080p" value="1080p" />

    <option name="1440p" value="1440p" />

    <option name="2160p" value="2160p" />

  </Option>

  <Pairs>

    <Pair>

      <Option>VideoRes|720p</Option>

      <Source>fc\_loading\_720p.bik</Source>

      <Target>ui\singleplayer\video\fc\_loading\_regionfaith.bik</Target>

    </Pair>

    <Pair>

      <Option>VideoRes|1080p</Option>

      <Source>fc\_loading\_1080p.bik</Source>

      <Target>ui\singleplayer\video\fc\_loading\_regionfaith.bik</Target>

    </Pair>

    <Pair>

      <Option>VideoRes|1440p</Option>

      <Source>fc\_loading\_1440p.bik</Source>

      <Target>ui\singleplayer\video\fc\_loading\_regionfaith.bik</Target>

    </Pair>

    <Pair>

      <Option>VideoRes|2160p</Option>

      <Source>fc\_loading\_2160p.bik</Source>

      <Target>ui\singleplayer\video\fc\_loading\_regionjohn.bik</Target>

    </Pair>

    <Pair>

      <Source>singleplayer.feu</Source>

      <Target>ui\singleplayer\menu\loading\gamemodes\singleplayer.feu</Target>

    </Pair>

  </Pairs>

Each *Pair* which has defined *Option* will be selected on selected option in Mod Installer. Other *Pair* which have a different *Option* than the selected one in Mod Installer will be ignored during packing.