**How to use replace package**

# Structure of info\_replace.xml

This is simple replace in itemdescriptor.ndb file. This example adds support for A-99 weapon for NPC.

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

<PackageInfoReplace>

  <Games>

<Game>FC5</Game>

</Games>

  <DefaultInclude>true</DefaultInclude>

  <Name>NPC Weapons Support</Name>

  <Description>Adds NPC support for weapons. NPC now can use some weapons that weren't created for using by NPC.</Description>

  <Category>7</Category>

  <Replaces>

    <Replace RequiredFile="databases/generic/itemdescriptor.ndb">

      <object hash="59F2984F">

        <primaryKey hash="8EDB0295">7366C26342072000</primaryKey> <!-- Weapons/Handguns/A-99 -->

        <object hash="452629B8">

          <field hash="CBF95ACC" addNode="1" type="String">weaponproperties:fcx.autopistols.a-99</field>

          <field hash="6993CC7C">2B568CE1</field>

        </object>

      </object>

    </Replace>

  </Replaces>

</PackageInfoReplace>

There are some standard tags as **DefaultInclude**, **Description**, etc…

Now the new tags:

* **Replaces** – there are defined replace actions, you can add as many as you want, see more below
* **Replace** – this is replace for one or only in one file. It needs attribute **RequiredFile** which is complete file name of file in which will be the replace created.

Also, take care of new root tag name – **PackageInfoReplace**

Now inside Replace tag there are all fields which will be changed. So, now how to make them?

# Replacing a value in FCB

As an example I took adding NPC support to A-99 weapon. We need to change:

* text\_ tmplWeaponNpcPropertiesName – hash CBF95ACC
* tmplWeaponNpcPropertiesName – hash 6993CC7C

So, now open converted FCB file. Use FCBConverter for converting files from fcb to xml.

Converted item descriptor for A-99 looks like this:

<object hash="59F2984F" name="Generic">

  <field hash="CF68E402" name="text\_hid\_DTCTH\_ClassName" type="String">CItemDescriptor</field>

  <field hash="25368426" name="hid\_DTCTH\_ClassName" type="ComputeHash32">CItemDescriptor</field>

  <field hash="8EDB0295" name="disLibItemId" type="BinHex">7366C26342072000</field>

  <field hash="D91FFFD3" name="disDbObjectName" type="String">Weapons/Handguns/A-99</field>

  <field hash="61FC6B02" name="hidSingleObject" type="BinHex">00</field>

  <field hash="1BC69C64" name="selCategory" type="BinHex">01000000</field>

  <object hash="452629B8" name="InventoryItemData">

    <field hash="CF68E402" name="text\_hid\_DTCTH\_ClassName" type="BinHex">CWeaponDescriptorData</field>

    <field hash="25368426" name="hid\_DTCTH\_ClassName" type="BinHex">CWeaponDescriptorData</field>

    <field hash="AB0AAEB9" name="selItemAddedStatisticEnum" type="BinHex">FFFFFFFF</field>

    <field hash="1BF7FB9D" name="bSharedItem" type="BinHex">00</field>

    <field hash="A788401A" name="bCanEquipOnReload" type="BinHex">01</field>

    <field hash="298E6C00" name="text\_tmplWeaponPropertiesName" type="String">weaponproperties:fcx.autopistols.a-99</field>

    <field hash="E45A0A1B" name="tmplWeaponPropertiesName" type="ComputeHash32">weaponproperties:fcx.autopistols.a-99</field>

    <field hash="6993CC7C" name="tmplWeaponNpcPropertiesName" type="BinHex">FFFFFFFF</field>

    <object hash="F9A7F4CA" name="itemdescriptorUpgradedItem">

      <field hash="BD865AD7" name="hid\_DbRefId" type="BinHex">0000000000000000</field>

      <field hash="CB3EDEF1" name="text\_hid\_DbRefT" type="String">IItemDescriptor</field>

      <field hash="57C4D76B" name="hid\_DbRefT" type="ComputeHash32">IItemDescriptor</field>

    </object>

    <object hash="94E7819E" name="UIData">

      <field hash="E95B2A50" name="IconName" type="String">A99</field>

      <field hash="BCD4E3F4" name="locidDisplayName" type="Int32">195230</field>

      <field hash="C10B3D5B" name="selLootFeedbackType" type="BinHex">00000000</field>

    </object>

    <object hash="43AB715D" name="itembagBagItem">

      <field hash="BD865AD7" name="hid\_DbRefId" type="BinHex">0000000000000000</field>

      <field hash="CB3EDEF1" name="text\_hid\_DbRefT" type="String">IItemDescriptor</field>

      <field hash="57C4D76B" name="hid\_DbRefT" type="ComputeHash32">IItemDescriptor</field>

    </object>

    <object hash="3EFA8AE3" name="arkEquipmentArchetype">

      <field hash="561D06B4" name="hid\_ArchRefId" type="BinHex">11BB78EC53072000</field>

    </object>

  </object>

  <object hash="AC5D9819" name="LootProperties">

    <object hash="D7C11237" name="ltdrpcontainerLootDropContainer">

      <field hash="BD865AD7" name="hid\_DbRefId" type="BinHex">0000000000000000</field>

      <field hash="CB3EDEF1" name="text\_hid\_DbRefT" type="String">CFCXLootDroppedContainer</field>

      <field hash="57C4D76B" name="hid\_DbRefT" type="ComputeHash32">CFCXLootDroppedContainer</field>

    </object>

  </object>

</object>

The program needs to know which field with hash **6993CC7C** you want to replace, because if you search the hash, you will see there are many fields with this hash, so you must define it’s parent. In this case the best is to select whole weapon descriptor entry, so you do this:

<object hash="59F2984F" name="Generic">

  <primaryKey hash="8EDB0295">7366C26342072000</primaryKey>

</object>

Tag **primaryKey** means that the program will search only objects with hash **59F2984F** which has in child fields field with hash **8EDB0295** and its value is **7366C26342072000**

You can also add more **primaryKey** tags if you want to replace more config entries with same values.

Now the other, you define full path to a field which you want to replace, so like this:

<object hash="59F2984F" name="Generic">

  <primaryKey hash="8EDB0295">7366C26342072000</primaryKey>

  <object hash="452629B8" name="InventoryItemData">

    <field hash="CBF95ACC" addNode="1" type="String">weaponproperties:fcx.autopistols.a-99</field>

    <field hash="6993CC7C">2B568CE1</field>

  </object>

</object>

Attribute **addNode** with value 1 means that the program add new field with defined hash, type and value.

Field with hash **6993CC7C** doesn’t have defined type. Why? Because the field exists in config entry, so you don’t have to add it to replace. Only if you add new fields, you must define it.

In **primaryKey** you can use **hash** or **name**. The value of **primaryKey** can be any type, simply add attribute **type** and set type.

Also, you can use only **name** attribute, whole replace can be **hash**-less.

<object name="Generic">

  <primaryKey name="disLibItemId" type="Id64">9015180977661555</primaryKey>

  <object name="InventoryItemData">

    <field name="text\_tmplWeaponNpcPropertiesName" addNode="1" type="String">weaponproperties:fcx.autopistols.a-99</field>

    <field name="tmplWeaponNpcPropertiesName" type="ComputeHash32">weaponproperties:fcx.autopistols.a-99</field>

  </object>

</object>

# Removing whole config entry

Sometimes you want to remove a config entry. You can do it like this:

<object name="GenericParent">

<object hash="59F2984F" name="Generic">

   <primaryKey hash="8EDB0295" removeParent="1">7366C26342072000</primaryKey>

</object>

</object>

Result:

<object name="GenericParent">

</object>

This will delete whole object with hash **59F2984F** which has child field with hash **8EDB0295** and value **7366C26342072000**

You define **primaryKey** and add attribute **removeParent="1"**. If you add **removeParent="2"** then two upper parents (objects) will be deleted.

# Cleaning whole object

To clean the whole *object*, for example for adding new *fields*, use *clearNodeBeforeInsert*. As the name says, it will clean the *object* node and after that it can add new *fields* or *objects*, or even *templates*. Value of the attribute is always 1.

Example:

<object hash="59F2984F" name="Generic">

  <primaryKey hash="8EDB0295">7366C26342072000</primaryKey>

  <object hash="452629B8" name="InventoryItemData" clearNodeBeforeInsert="1">

    <field hash="CBF95ACC" type="String">weaponproperties:fcx.autopistols.a-99</field>

    <field hash="6993CC7C">2B568CE1</field>

  </object>

</object>

will result into:

<object hash="59F2984F" name="Generic">

  <field hash="8EDB0295">7366C26342072000</field>

  <object hash="452629B8" name="InventoryItemData">

  </object>

</object>

# Making a duplicate of an object

This can be useful if you want to add a new entry with the same values but only change one or two *fields*. Notice the *newHash* attribute which will set the new *object*’s *hash*.

Example:

This is original config entry:

<object hash="59F2984F" name="Generic">

  <field hash="8EDB0295">7366C26342072000</field>

  <object hash="452629B8" name="InventoryItemData">

    <field hash="CBF95ACC" type="String">weaponproperties:fcx.autopistols.a-99</field>

    <field hash="6993CC7C">2B568CE1</field>

  </object>

</object>

This is prepared replace:

<object hash="59F2984F" name="Generic">

  <primaryKey hash="8EDB0295">7366C26342072000</primaryKey>

  <object hash="452629B8" name="InventoryItemData" makeObjectDuplicate="1" newHash="C8AA02BC">

    <field hash="CBF95ACC" type="String">new value</field>

  </object>

</object>

will result into:

<object hash="59F2984F" name="Generic">

  <field hash="8EDB0295">7366C26342072000</field>

  <object hash="452629B8" name="InventoryItemData">

    <field hash="CBF95ACC" type="String">weaponproperties:fcx.autopistols.a-99</field>

    <field hash="6993CC7C">2B568CE1</field>

  </object>

  <object hash="452629B8" name="InventoryItemData">

    <field hash="CBF95ACC" type="String">new value</field>

    <field hash="6993CC7C">2B568CE1</field>

  </object>

</object>

Notice the changed value of the *field*.

# Making a duplicate of a field

Same as duplicate of an *object* but for a *field*. Its value is ignored - it’s taken from the source *field*, so it can be used only as duplication.

    <field hash="CBF95ACC" newHash="AF6B90CA" makeFieldDuplicate="1" />

# Advanced - sorting / reordering objects

Some config entries require to be sorted according to some *field* values. For this action there are attributes *reorder\_object*, *reorder\_by* and *reorder\_type*. Those attributes must be used together and they’re applied to *object*.

Example:

<object hash="B5456221">

<object hash="FEFEC7DE">

<field hash="ED668878" type="BinHex">01000000</field>

<field name="SomeFieldName" type="String">SomeValue - Tree</field>

</object>

<object hash="FEFEC7DE">

<field hash="ED668878" type="BinHex">00000000</field>

<field name="SomeFieldName" type="String">SomeValue - House</field>

</object>

<object hash="FEFEC7DE">

<field hash="ED668878" type="BinHex">02000000</field>

<field name="SomeFieldName" type="String">SomeValue - Bench</field>

</object>

</object>

<object hash="B5456221" reorder\_object="FEFEC7DE" reorder\_by="ED668878" reorder\_type="desc" />

Result:

<object hash="B5456221">

<object hash="FEFEC7DE">

<field hash="ED668878" type="BinHex">02000000</field>

<field name="SomeFieldName" type="String">SomeValue - Bench</field>

</object>

<object hash="FEFEC7DE">

<field hash="ED668878" type="BinHex">01000000</field>

<field name="SomeFieldName" type="String">SomeValue - Tree</field>

</object>

<object hash="FEFEC7DE">

<field hash="ED668878" type="BinHex">00000000</field>

<field name="SomeFieldName" type="String">SomeValue - House</field>

</object>

</object>

So, how to define it:

* reorder\_object - it’s a hash of children *objects* which will be sorted
* reorder\_by - it’s a *field* in the child *object*, its value is taken to sorting
* reorder\_type - type of sorting, it can be **asc** or **desc**

# Duplicating parent object of primaryKey

This is somehow the same as other *makeObjectDuplicate* and *makeFieldDuplicate*, but this one is used only with *primaryKey*. The attribute name is *makeParentDuplicate*, value is always 1.

Example:

<object hash="59F2984F">

<primaryKey hash="8EDB0295" makeParentDuplicate="1">2827558C65072000</primaryKey>

<field hash="8EDB0295" type="Id64">9015331982223145</field>

<object hash="452629B8">

<object hash="3EFA8AE3">

<field hash="561D06B4" type="Id64">9015331982089933</field>

</object>

</object>

</object>

As you can see, it will make duplicate of *object* 59F2984F and change value of *field* 8EDB0295 which is also used as *primaryKey*. In this case it makes duplicate and also does some replacing in the newly duplicated *object*.

# Find fields which contains string via primaryKey

There can be situations when you want to find several fields which start with the same string but end with different strings, in this case this feature can be useful. Attribute name is *contains*, value is always 1.

<object hash="BB96607D">

<primaryKey hash="D91FFFD3" contains="1" type="String">Databases\UI\ShoppingItems\_Bowmore\Clothing</primaryKey>

</object>

This will find all *objects* BB96607D which have a *field* D91FFFD3 with a string which contains “Databases\UI\ShoppingItems\_Bowmore\Clothing”.

# First-only child primaryKey occurrence

If you want to find *primaryKey* only in the first child of an *object*, you can use *primaryKeyExact*. Classic *primaryKey* searching *fields* in all descendants of parent *object*.

<object hash="507DA888" name="ActivityConditionsData">

<object hash="E4807B9E" name="actdescActivity">

<primaryKeyExact hash="BD865AD7" name="hid\_DbRefId" removeParent="2">40B9B35368072000</primaryKeyExact>

</object>

</object>

The reason for usage can be that there can be two *fields* with the same *hash* and value in child *objects* and if you want to select only the first one, then this is the solution.

# Replacing fields where cannot be primaryKey set

Sometimes there are config entries which have the same names, but they are like arrays. Like this:

<object hash="90E76EC8" name="HealthRegenerationDelays">

  <object hash="599C3960" name="DifficultyLevel">

    <field hash="CCB806A0" name="RegenDelay" type="Float32">3</field>

  </object>

  <object hash="599C3960" name="DifficultyLevel">

    <field hash="CCB806A0" name="RegenDelay" type="Float32">5</field>

  </object>

  <object hash="599C3960" name="DifficultyLevel">

    <field hash="CCB806A0" name="RegenDelay" type="Float32">10</field>

  </object>

  <object hash="599C3960" name="DifficultyLevel">

    <field hash="CCB806A0" name="RegenDelay" type="Float32">13</field>

  </object>

  <object hash="599C3960" name="DifficultyLevel">

    <field hash="CCB806A0" name="RegenDelay" value-Int32="0" type="BinHex">00000000</field>

  </object>

</object>

And we want to replace field which has value 13. So we make this:

<object hash="90E76EC8" name="HealthRegenerationDelays">

  <object hash="599C3960" name="DifficultyLevel" count="3">

    <field hash="CCB806A0" name="RegenDelay" type="Float32">13</field>

  </object>

</object>

I added attribute **count="3"**. It means that fourth (because it counts from zero) object will be selected. And now we have selected what we wanted.

This example also shows that I didn’t use **primaryKey** even in object with hash **90E76EC8**. And why? Because if you search for hash **90E76EC8** in nomadobjecttemplates\_rt.fcb you will see that it is there sixth times and I want to replace all sixth objects.

And now a little more. What if you want to replace only one of the six objects? Do you know how to do it? If not, look below.

1. We must find parent object of object with hash **90E76EC8.**
2. It is an object with hash **C9762625** named as **Health**. But this is also sixth times there. So we must go upper.
3. Next parent is an object with hash **B96328AF** named as **StatsAndCounters.** But still same. Go upper.
4. Finally we found object with hash **6E167DD5** named **Template**. It has child with hash **C821B0C6** named **FullName**. Value of this field is unique – this is what we need.

So, here is complete replace for only one specific object of six.

<object hash="6E167DD5" name="Template">

  <!-- CountersService:Counters.CountersService -->

  <primaryKey hash="C821B0C6" name="FullName">436F756E74657273536572766963653A436F756E746572732E436F756E746572735365727669636500</primaryKey>

  <object hash="B96328AF" name="StatsAndCounters">

    <object hash="C9762625" name="Health">

      <object hash="90E76EC8" name="HealthRegenerationDelays">

        <object hash="599C3960" name="DifficultyLevel">

          <field hash="CCB806A0" name="RegenDelay" type="Float32">3</field>

        </object>

        <object hash="599C3960" name="DifficultyLevel">

          <field hash="CCB806A0" name="RegenDelay" type="Float32">5</field>

        </object>

        <object hash="599C3960" name="DifficultyLevel">

          <field hash="CCB806A0" name="RegenDelay" type="Float32">10</field>

        </object>

        <object hash="599C3960" name="DifficultyLevel">

          <field hash="CCB806A0" name="RegenDelay" type="Float32">13</field>

        </object>

        <object hash="599C3960" name="DifficultyLevel">

          <field hash="CCB806A0" name="RegenDelay" value-Int32="0" type="BinHex">00000000</field>

        </object>

      </object>

    </object>

  </object>

</object>

Also, all strings in **primaryKey** must be in hex.

# What if primaryKey will not exist?

Nothing will happen. **primaryKey** is like search, if nothing is found, then whole parent **object** of **primaryKey** is ignored.

# Using replace for field array

There are some config fields with arrays. It means there are multiple values like several Vector3 values.

There are two ways to use replace on array fields - first one is replacing the whole field with new values, the second one is replacing one item in the array.

This is a sample config entry taken from material. It’s a field with a Vector4 array defining a color.

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

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

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

<Color>0.29765308,0.32038102,0.35865393,1</Color>

<Color>0.05637399,0.08690139,0.061907507,1</Color>

<Color>1,1,1,1</Color>

<Color>1,1,1,1</Color>

<Color>1,1,1,1</Color>

<Color>1,1,1,1</Color>

<Color>1,1,1,1</Color>

<Color>1,1,1,1</Color>

<Color>1,1,1,1</Color>

<Color>1,1,1,1</Color>

<Color>1,1,1,1</Color>

<Color>1,1,1,1</Color>

</field>

</object>

Here is an example for the first way. We want to replace whole array with own values, so we make a replace:

<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>0.29765308,0.32038102,0.35865393,1</Color>

<Color>0.05637399,0.08690139,0.061907507,1</Color>

<Color>1,1,1,1</Color>

<Color>2,1,1,1</Color>

<Color>3,1,1,1</Color>

<Color>4,1,1,1</Color>

<Color>5,1,1,1</Color>

<Color>6,1,1,1</Color>

<Color>7,1,1,1</Color>

<Color>8,1,1,1</Color>

<Color>9,1,1,1</Color>

<Color>10,1,1,1</Color>

</field>

</object>

Notice that **type** was added to the field. It’s required because the Mod Installer needs to know what type of array items is used.

This first way is useful when we want to change every item of the array.

Now let’s try the second way. It’s useful when we want to replace just a single item from an array.

<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">1000,1000,1000,1000</field>

</object>

Notice that **type** was added to the field. It’s required because the Mod Installer needs to know what type of array items is used.

Also notice that **arrayIndex** attribute was added. It’s required for defining what item we want to replace. It’s counted from zero so it means the first item has index 0.

In this example we’re replacing item with index 10 which is the one before the last one in the sample config entry.