**How to use templates in info**

Sometimes when we need to add more same entries, but just with one, two or three values different, then we don’t have to add whole entry, but we can use Templates.

An example of normal, longer use:

<object hash="59F2984F">

  <primaryKey hash="8EDB0295">D2E24FBB52072000</primaryKey>

  <object hash="1142A285">

    <object hash="9E036BCA">

      <field hash="2AB6B940" type="BinHex">2E237B0B52072000</field>

      <field hash="359A33C3" type="BinHex">0A000000</field>

      <field hash="D52715D0" type="BinHex">0000803F</field>

      <field hash="CECC0C92" type="BinHex">0000803F</field>

      <object hash="F401329E">

        <field hash="BD865AD7" type="BinHex">B742B1B24B072000</field>

        <field hash="CB3EDEF1" type="String">CPreConditionKnowledge</field>

        <field hash="57C4D76B" type="BinHex">F2163D9B</field>

      </object>

    </object>

    <object hash="9E036BCA">

      <field hash="2AB6B940" type="BinHex">BD3542A453072000</field>

      <field hash="359A33C3" type="BinHex">0A000000</field>

      <field hash="D52715D0" type="BinHex">0000803F</field>

      <field hash="CECC0C92" type="BinHex">0000803F</field>

      <object hash="F401329E">

        <field hash="BD865AD7" type="BinHex">B742B1B24B072000</field>

        <field hash="CB3EDEF1" type="String">CPreConditionKnowledge</field>

        <field hash="57C4D76B" type="BinHex">F2163D9B</field>

      </object>

    </object>

    <object hash="9E036BCA">

      <field hash="2AB6B940" type="BinHex">713BA82A5A072000</field>

      <field hash="359A33C3" type="BinHex">0A000000</field>

      <field hash="D52715D0" type="BinHex">0000803F</field>

      <field hash="CECC0C92" type="BinHex">0000803F</field>

      <object hash="F401329E">

        <field hash="BD865AD7" type="BinHex">B742B1B24B072000</field>

        <field hash="CB3EDEF1" type="String">CPreConditionKnowledge</field>

        <field hash="57C4D76B" type="BinHex">F2163D9B</field>

      </object>

    </object>

  </object>

</object>

This is not so long, but there are other, a very long entries, for example in patch.dbt.fcb.

So, all three entries are the same, only field with hash 2AB6B940 (ID) is different.

Now, we define new tag named *Templates* in root of info\_replace.xml:

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

<PackageInfoReplace>

  <Games>

<Game>FC5</Game>

</Games>

  <DefaultInclude>false</DefaultInclude>

  <Name>Faith After Liberation</Name>

  <Description>Faith also appear after Henbane is liberated.</Description>

  <Category>15</Category>

  <Replaces>

    ...

  </Replaces>

  <Templates>

  </Templates>

</PackageInfoReplace>

Now we add our first template. It need attribute named id, which is needed for further use, it can be anything – number, string (case sensitive):

<Template id="myTemplate">

  <object hash="9E036BCA">

    <field hash="2AB6B940" templateValue="ID" type="BinHex">a default value</field>

    <field hash="359A33C3" type="BinHex">0A000000</field>

    <field hash="D52715D0" type="BinHex">0000803F</field>

    <field hash="CECC0C92" type="BinHex">0000803F</field>

    <object hash="F401329E">

      <field hash="BD865AD7" type="BinHex">B742B1B24B072000</field>

      <field hash="CB3EDEF1" type="String">CPreConditionKnowledge</field>

      <field hash="57C4D76B" type="BinHex">F2163D9B</field>

    </object>

  </object>

</Template>

And we put it into *Templates*

You can see I added to a field with hash 2AB6B940 new attribute named *templateValue*. It means this field will be used as a variable, so its value will change in every use of template, more later. Value of the attribute can be also anything, if it will be string, then it is case sensitive.

You can add as many *templateValue* as you want, so basically every field can have its own *templateValue.*

Now how to use the template?

I changed the first example, so it will now have a template. It means I delete all objects with hash 9E036BCA. So:

<object hash="59F2984F">

  <primaryKey hash="8EDB0295">D2E24FBB52072000</primaryKey>

  <object hash="1142A285">

    <template id="1" templateValueID="2E237B0B52072000" />

    <template id="1" templateValueID="BD3542A453072000" />

    <template id="1" templateValueID="713BA82A5A072000" />

  </object>

</object>

You just define tag named *template* and add to it attribute named *id* which is name, id of our template and special tag *templateValue****ID***. FIrst part is *templateValue* – this identifying that we want to use a variable, but you must also define which variable in template you want to change, so you add the same which you defined in template in *templateValue* as its value, it was ID, so then *templateValueID.*

And value of *templateValueID* is what you want, in this case we wanted different ID.

By this way we saved many lines and also xml is more readable and clearer. Also, you can add new entries by this way without using non-templates way with adding *addNode=“1“*

………….

Note: if your template has variables and when you define the template and you don’t define a value of a variable, then default will be used.

Example:

This is a defined variable **ID** with no default value, so if you don’t define the variable when calling a *template*, it will raise an error during processing.

  <template id="myTemplate" />

...

<Template id="myTemplate">

   <object hash="9E036BCA">

<field hash="BD865AD7" templateValue="ID" type="Id64" />

   </object>

</Template>

So, two ways - define default value or set the variable **ID** in calling *template*, or both:

  <template id="myTemplate" templateValueID="9125963258740256" />

...

<Template id="myTemplate">

   <object hash="9E036BCA">

<field hash="BD865AD7" templateValue="ID" type="Id64">0</field>

   </object>

</Template>

The default value can be useful if you’re using the template in more places and only once you want to change the value of the variable.

………….

Whole info\_replace will look like:

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

<PackageInfoReplace>

  <Games>

<Game>FC5</Game>

</Games>

  <DefaultInclude>false</DefaultInclude>

  <Name>Faith After Liberation</Name>

  <Description>Faith also appear after Henbane is liberated.</Description>

  <Category>15</Category>

  <Replaces>

    <Replace RequiredFile="databases\generic\encounterprefablist.ndb">

      <object hash="59F2984F">

        <primaryKey hash="8EDB0295">D2E24FBB52072000</primaryKey> <!-- FCX\_Encounters/Thematics/Selena\_Specter -->

        <object hash="1142A285">

          <template id="1" templateValueID="2E237B0B52072000" />

          <template id="1" templateValueID="BD3542A453072000" />

          <template id="1" templateValueID="713BA82A5A072000" />

        </object>

      </object>

    </Replace>

  </Replaces>

  <Templates>

    <Template id="1">

      <object hash="9E036BCA">

        <field hash="2AB6B940" templateValue="ID" type="BinHex" />

        <field hash="359A33C3" type="BinHex">0A000000</field>

        <field hash="D52715D0" type="BinHex">0000803F</field>

        <field hash="CECC0C92" type="BinHex">0000803F</field>

        <object hash="F401329E">

          <field hash="BD865AD7" type="BinHex">B742B1B24B072000</field>

          <field hash="CB3EDEF1" type="String">CPreConditionKnowledge</field>

          <field hash="57C4D76B" type="BinHex">F2163D9B</field>

        </object>

      </object>

    </Template>

  </Templates>

</PackageInfoReplace>

# Processing templates

**This is a special advanced usage. You must be 100% sure before using it.**

If you use it, MI will process the *template*, which means by default MI doesn’t check any MI attributes in the *template* like *count*, *clearNodeBeforeInsert*, *removeObject*, etc…

So if you need it, add attribute *process=”1”* to the *template* definition:

  <Templates>

    <Template id="1" process="1">

      <object hash="9E036BCA">

        <field hash="2AB6B940" templateValue="ID" type="BinHex" />

        <field hash="359A33C3" type="BinHex">0A000000</field>

        <field hash="D52715D0" type="BinHex">0000803F</field>

        <field hash="CECC0C92" type="BinHex">0000803F</field>

        <object hash="F401329E">

          <field hash="BD865AD7" type="BinHex">B742B1B24B072000</field>

          <field hash="CB3EDEF1" type="String">CPreConditionKnowledge</field>

          <field hash="57C4D76B" type="BinHex">F2163D9B</field>

        </object>

      </object>

    </Template>

  </Templates>

**Note: if you use *Options* in *template*, it will work without using *process*.**