Andrew Downes
andownes@gmail.com

Expose Additional QuantLib Enumerations to QuantLibXL

This document explains how to expose additional QuantLib enumerated types to Excel. It is assumed that you have already compiled the Full build of QuantLibXL.

1. The Enumeration

In this example we use a new, trivial, example of an enumerated type. We assume the code for the enumeration is contained in the file:

`QuantLib\ql\newfiles\newenums.hpp`

This file is added to the root directory of the `QuantLib` project. The code for the enumeration is below:

```
#ifndef new_enums_h
#define new_enums_h

namespace QuantLib {

enum NewEnumeration {
      NO_NEW_ENUMERATION,
      ENUM_VALUE_1,
      ENUM_VALUE_2
};

} // namespace

#endif
```

2. Adding the Enumeration to QuantLibXL

Open the file:

`QuantLibAddin\gensrc\metadata\enumerations\enumeratedtypes.xml`

Between `<EnumeratedTypeGroups>` and `</EnumeratedTypeGroups>` add the code:

```
    <EnumeratedTypeGroup type='QuantLib::NewEnumeration'>
      <includeFile>oh/enumerations/typefactory.hpp</includeFile>
      <constructor>true</constructor>
      <EnumeratedTypes>
        <EnumeratedType>
          <string>NO_NEW_ENUMERATION</string>
          <value>QuantLib::NO_NEW_ENUMERATION</value>
        </EnumeratedType>
        <EnumeratedType>
          <string>ENUM_VALUE_1</string>
          <value>QuantLib::ENUM_VALUE_1</value>
        </EnumeratedType>
        <EnumeratedType>
          <string>ENUM_VALUE_2</string>
          <value>QuantLib::ENUM_VALUE_2</value>
        </EnumeratedType>
      </EnumeratedTypes>
    </EnumeratedTypeGroup>
```

The line `<constructor>true</constructor>` will be discussed in more detail in the final section, for all QuantLib enumerations that you wish to expose to Excel the line should remain as above.

Edit the file QuantLibAddin\gensrc\metadata\types\types.xml and add the new data type NewEnumeration:

```
<root>
  <DataTypes>
    <DataType
defaultSuperType='enumeration'>QuantLib::Average::Type</DataType>
...
    <DataType
defaultSuperType='enumeration'>QuantLib::NewEnumeration</DataType>
...
    <DataType
defaultSuperType='underlyingClass'>QuantLib::Leg</DataType>
  </DataTypes>
</root>
```

Edit the file QuantLibAddin\gensrc\stubs\stub.enum.types and add the file containing the new enumeration:

```
#include <qlo/qladdindefines.hpp>
...
#include <ql/newfiles/newenums.hpp>
```

3. Using the Enumeration in QuantLibXL

To use this enumeration in a function/constructor call from excel, add the parameter to the function/constructor in the appropriate .xml file by using:

```xml
    <Member name='qlMyFunction' type='QuantLib::MyClass'>
      <description>function description</description>
      <libraryFunction>myFunction</libraryFunction>
      <SupportedPlatforms>
        <SupportedPlatform name='Excel' calcInWizard='false'/>
      </SupportedPlatforms>
      <ParameterList>
        <Parameters>
          <Parameter name='NewEnum' exampleValue='ENUM_VALUE_1'>
            <type>QuantLib::NewEnumeration</type>
            <tensorRank>scalar</tensorRank>
            <description>new enumeration</description>
          </Parameter>
        </Parameters>
      </ParameterList>
      <ReturnValue>
        <type>double</type>
        <tensorRank>scalar</tensorRank>
      </ReturnValue>
    </Member>
```

The enumeration can now be used in excel by entering a string. For the above example we write:

```
=qlMyFunction(ENUM_VALUE_1)
```

4. The constructor flag

As mentioned, for all QuantLib enumerated types that you wish to expose to Excel the `<constructor>` tag should remain set to true. This associates the string "`ENUM_VALUE_1`" in Excel with the QuantLib type `QuantLib::NewEnumeration(QuantLib::ENUM_VALUE_1)`.

However, you may wish to associate a string in Excel directly with a QuantLib class. An example of this is given by currency inputs to functions, which can be written simply as (for example) "AUD", rather than going through the process of constructing an Australian dollar currency class.

To achieve a similar result, set the `<constructor>` tag to false. In the above example, this would associate the string "`ENUM_VALUE_1`" in Excel with the QuantLib type `QuantLib::ENUM_VALUE_1` – the string in the `<string>` tag is replaced directly with the type in the `<value>` tag. Hence, as in the currency example, if you wanted the string AUD to give an AUDCurrency class, we would set the `<constructor>` tag to false, set the `<string>` tag to AUD and the `<value>` tag to QuantLib::AUDCurrency().