Generate XML programatically in Qt

This code snippet shows how to generate XML using QXmlStreamWriter. The `QXmlStreamWriter` class provides an XML writer with a simple streaming API.

Overview

`QXmlStreamWriter` is the pendent to `QXmlStreamReader` for writing XML. Like its related class, it operates on a `QIODevice` specified with `setDevice()`. Element tags are opened with `writeStartElement()` followed by `writeAttribute()` or `writeAttributes()`, element content, and then `writeEndElement()`. A shorter form `writeEmptyElement()` can be used to write empty elements, followed by `writeAttributes()`.

This snippet can be self-signed.

Headers files

```
#include <QXmlStreamWriter>

class GenerateXML : public QMainWindow
{
    Q_OBJECT

public:
 GenerateXML(QWidget *parent = 0);
    ~GenerateXML();
 //call this function to create xml
    void CreateXMLFile();

private:
    Ui::GenerateXMLClass ui;
 //variable having data from which we have to generate xml.
 QMap<QString,QString> firstname;
 QMap<QString,QString> surname;
 QMap<QString,QString> phoneNumber;
};
```

Source

```
#include "GenerateXML.h"
#include <QMap>
#include <QFile>
#include <QMessageBox>
#include <QXmlStreamWriter>
#include <QMapIterator>

// code for constructor

GenerateXML::GenerateXML(QWidget *parent)
    : QMainWindow(parent)
{
 ui.setupUi(this);
 setWindowTitle("Generate XML");

 /* Assuming user rerived data from which he wants to creare XML.
 * lets we have data of 5 user to create XML.
 */
 firstname.insertMulti("firstname", "abc");
 firstname.insertMulti("firstname", "ghi");
 firstname.insertMulti("firstname", "mno");
 firstname.insertMulti("firstname", "stu");
 firstname.insertMulti("firstname", "yza");

 surname.insertMulti("surname", "def");
 surname.insertMulti("surname", "jkl");
 surname.insertMulti("surname", "pqr");
 surname.insertMulti("surname", "vwx");
 surname.insertMulti("surname", "bcd");

 phoneNumber.insertMulti("number", "12345");
 phoneNumber.insertMulti("number", "67890");
 phoneNumber.insertMulti("number", "12356");
 phoneNumber.insertMulti("number", "87654");
 phoneNumber.insertMulti("number", "756776");

 /* call method to generate XML */
 CreateXMLFile();
}

void GenerateXML::CreateXMLFile()
{
    /* For testing purpose, we generate in C drives,
 QFile file("c://contacts.xml");

 /*open a file */
 if (!file.open(QIODevice::WriteOnly))
 {
 /* show wrror message if not able to open file */
 QMessageBox::warning(0, "Read only", "The file is in read only mode");
 }
 else
 {
  /*if file is successfully opened then create XML*/
  QXmlStreamWriter* xmlWriter = new QXmlStreamWriter();
  /* set device (here file)to streamwriter */
  xmlWriter->setDevice(&file);
  /* Writes a document start with the XML version number version. */
  xmlWriter->writeStartDocument();
```


(C) Copyright Nokia 2013. All rights reserved.

```
   /* Writes a start element with name,
   * Subsequent calls to writeAttribute() will add attributes to this element.
    here we creating a tag <persons>*/
   xmlWriter->writeStartElement("persons");

   /* create mapiterator to iterate through maps*/
   QMapIterator<QString, QString> i(firstname);
   QMapIterator<QString, QString> j(surname);
   QMapIterator<QString, QString> k(phoneNumber);

   /*this while loop generates <person> tag with attributes and CDATA in each execution.
   * currently we have 5 element in map so loop creates 5 tags.
   */
   while (i.hasNext() && j.hasNext() && k.hasNext())
      {
    i.next(); j.next(); k.next();
    /* create tag person */
    xmlWriter->writeStartElement("person");
    /*add one attribute and its value*/
    xmlWriter->writeAttribute(i.key(), i.value());
    /*add one attribute and its value*/
    xmlWriter->writeAttribute(j.key(), j.value());
    /*add one attribute and its value*/
    xmlWriter->writeAttribute(k.key(), k.value());
    /*add character data to tag person */
    xmlWriter->writeCharacters ("kamlesh");
    /*close tag person  */
    xmlWriter->writeEndElement();

      }
      /*end tag persons*/
   xmlWriter->writeEndElement();
   /*end document */
   xmlWriter->writeEndDocument();
               delete xmlWriter;
 }
}

// code for destructor
GenerateXML::~GenerateXML()
{

}
```

Postconditions

The code snippet is expected to generate XML as shown below.

```
   <?xml version="1.0" encoding="UTF-8" ?>
 - <persons>
   <person firstname="yza" surname="bcd" number="756776">kamlesh</person>
   <person firstname="stu" surname="vwx" number="87654">kamlesh</person>
   <person firstname="mno" surname="pqr" number="12356">kamlesh</person>
   <person firstname="ghi" surname="jkl" number="67890">kamlesh</person>
   <person firstname="abc" surname="def" number="12345">kamlesh</person>
   </persons>
```

External links

- QXmlStreamWriter: QXmlStreamWriter reference

Internal links

- QXmlStreamReader to parse XML in Qt

Code Example

- File:QtGenerateXML.zip will generate XML and is tested on Nokia 5800 XpressMusic.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.