



ID	...	Creation date	June 13, 2009
Platform	S60 3rd Edition, FP1, FP2 S60 5th Edition	Tested on devices	Nokia 5800 XpressMusic
Category	Qt for Symbian	Subcategory	

Keywords (APIs, classes, methods, functions): QDomStreamWriter

Overview

This code snippets shows how to generate XML using QDomStreamWriter. The QDomStreamWriter class provides an XML writer with a simple streaming API.

QDomStreamWriter is the pendent to QDomStreamReader 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. As it does not use any API which require developer/certified signing.

Preconditions

- Download and Install latest version [Qt for Symbian - Installation packages](#) which has links on how to install the latest version

Headers files

```
#include <QDomStreamWriter>

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.
```

Generate_XML_programatically_in_Qt_for_Symbian

```
<QString, QMap> firstname;  
<QString, QMap> surname;  
<QString, QMap> phoneNumber;  
};
```

Source

```
#include "GenerateXML.h"  
#include <QMap>  
#include <QFile>  
#include <QMessageBox>  
#include <QXmlStreamWriter>  
#include <QMapIterator>  
  
GenerateXML::GenerateXML(QWidget *parent)  
    : QMainWindow(parent)  
{  
    setupUi(this);  
    setWindowTitle("Generate XML");  
  
    /* Assuming user received data from which he wants to create XML.  
     * lets we have data of 5 user to create XML.  
     */  
    firstNameMulti("firstname", "abc");  
    firstNameMulti("firstname", "ghi");  
    firstNameMulti("firstname", "mno");  
    firstNameMulti("firstname", "stu");  
    firstNameMulti("firstname", "yza");  
  
    surnameMulti("surname", "def");  
    surnameMulti("surname", "jkl");  
    surnameMulti("surname", "pqr");  
    surnameMulti("surname", "vwx");  
    surnameMulti("surname", "bcd");  
  
    phoneNumberMulti("number", "12345");  
    phoneNumberMulti("number", "67890");  
    phoneNumberMulti("number", "12356");  
    phoneNumberMulti("number", "87654");  
    phoneNumberMulti("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 error message if not able to open file */  
        QMessageBox(0, "Read only", "The file is in read only mode");  
    }  
    else  
    {  
        /*if file is successfully opened then create XML*/  
        QXmlStreamWriter writer(new QXmlStreamWriter());
```

Generate_XML_programatically_in_Qt_for_Symbian

```
/* set device (here file)to streamwriter */
    ->setDevice(&file);
/* Writes a document start with the XML version number version. */
    ->writeStartDocument();
/* Writes a start element with name,
    * Subsequent calls to writeAttribute() will add attributes to this element.
    here we creating a tag <persons>*/
    ->writeStartElement("persons");

/* create mapiterator to iterate through maps*/
    <QString> i(firstname);
    <QString> j(surname);
    <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())
{
    next(); j.next(); k.next();
/* create tag person */
    ->writeStartElement("person");
/*add one attribute and its value*/
    ->writeAttribute("key1", i.value());
/*add one attribute and its value*/
    ->writeAttribute("key2", j.value());
/*add one attribute and its value*/
    ->writeAttribute("key3", k.value());
/*add character data to tag person */
    ->writeCharacterData("kamlesh");
/*close tag person */
    ->writeEndElement();
}
/*end tag persons*/
    ->writeEndElement();
/*end document */
    ->writeEndDocument();
    delete xmlWriter;
}
}

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>
```

Generate_XML_programatically_in_Qt_for_Symbian

```
<person firstname="abc" surname="def" number="12345">kamlesh</person>  
</persons>
```

External links

- [QXmlStreamWriter: QXmlStreamWriter reference](#)

Internal links

- [Using QXmlStreamReader to parse XML](#)

Code Example

- The [Code Example](#) will generate XML and is tested on Nokia 5800 XpressMusic.