RSS Feed reader for All Georgia Tech Newsfeeds from the Georgia Tech news center

Members: Taeyong Shin, Woojoo Shin, Soo Kwon Ha

Introduction

The goal of this project is to build RSS (Rich Site Summary) feed reader to display the “All Georgia Tech News” feed found on the Georgia Tech News Center (http://www.news.gatech.edu/rss). Building an RSS feed reader had three main parts of development. First, an XML parser had to be developed so that the data from the Georgia Tech News feed can be formatted for display. Secondly, a server connection had to be made so that XML data can be received from the internet. Lastly, a graphic user interface had to be developed that was easy for a user to use.

This project makes use of a software called Qt to develop a GUI, to read and parse XML (Extensible Markup Language) files and to implement a HTTP (Hypertext Transfer Protocol) connection. The project focuses on important aspects of application development such as GUI development, event handling, and server implementation. Lastly, to allow the project to be compiled across various platforms, QMAKE was used to compile the final project.

The final product of this project returns an executable file that displays the Georgia Tech News RSS Feed. This system has an automatic refresh capability, which refreshes the system every five minutes to make sure that new articles have not been posted. The system also has a manual refresh capability that allows the user to refresh the RSS feed whenever they desire. The QMAKE compiler compiles to reflect the native look of an operating system which gives the application a more natural look on any system.

How To Set Up the Project

To set up the project homebrew and qt must be installed to build and compile the system. First, open up the terminal and copy the following command to download homebrew: /usr/bin/ruby

-e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)". Then using the newly installed homebrew package manager, download qt using the command: brew install qt on the terminal. To allow the qmake command to work on the computer, qt must first be linked to the system. To link qt to the system run the following command on the terminal : brew link qt --force. After these steps the program can now be built and compiled. Unzip the rss-gui.zip file to uncompress the final project. Then using the terminal cd into the rss-gui folder. From here use the following command to build and compile the system on the terminal: qmake && make. After the program is compiled there should be an executable file in the rss-gui file called “rss-gui”. Double click the executable file and the program should start running right away.

Qt overview

Qt is a cross-platform software framework for creating embedded GUI. The Qt GUI module provides classes for integration, 2D graphics, imaging, fonts, text and handlings of events. Higher level API’s like At Widgets and Qt Quick provide various technologies for creating user interfaces, helping users easily access to display function of the program. In this project specifically, we incorporated the Core, XML, and Network modules to allow for asynchronous threading, XML reading and parsing, and HTTP connections.

GUI Development with QT

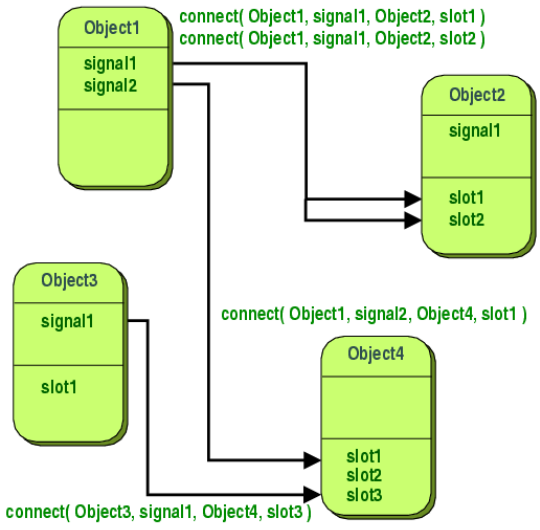

Figure 1. Diagram of Signal and Slot Method

The main reason that Qt was chosen for this project was because of its aptitude for GUI development. A major advantage of using Qt for GUI development was that it offers a graphical interface for GUI development. Instead of having to code every part of the GUI by hand, click and drop options were readily available for GUI development. Due to this advantage, more time could be used for development of the back-end side of the project to allow for a stronger focus on the C++ aspects of this project.

Furthermore, because Qt also offers event handling capabilities features such as automatic

and manual refresh could be easily developed. Qt uses a signals and slots method (shown in **Figure 1.**) as an alternative to the callback technique for event handling. In Qt, when an event occurs a signal is emitted, “signaling” to the system that the event has occurred. Then the signal is received by a “slot”, which then triggers a function as a response to the signal. For this project, a timer signal is used to allow automatic refresh of the RSS feed, and a push button signal is used to allow for a manual refresh of the RSS feed. For automatic refresh, a timer was set with an interval of five minutes. When the interval had passed the timer would “time out” and send a signal indicating this event. For manual refresh, a push button being clicked is the trigger of a signal emission. When these signals are received by the slot containing the refresh function, the load_rss() method is returned to begin the refresh process.

XML with QT

The Georgia Tech News Center’s RSS news feeds are given in XML format which required the use of the XML module of Qt for proper handling. XML is a simple self-descriptive text format. It is designed to carry data with the focus on describing what that data is. For example, in the case of the Georgia Tech News Center’s RSS news feeds, the XML data contains information such as, the title of the news article, a link to the article, a description of the article,

```
    <title>Celebrating Small</title>
    <link>http://www.news.gatech.edu/2018/05/03/celebrating-small</link>
    <description>  &lt;span class=&quot;date-display-single&quot; property=&quot;dc:date&quot; datatype=&quot;xsd:dateTime&quot;
content=&quot;2018-05-03T00:00:00-04:00&quot;&gt;May 3, 2018&lt;/span&gt;  Atlanta, GA  Graduating from Georgia Tech is a big accomplishment,
but one way to commemorate it is very small.  &lt;a href=&quot;/old-mercury-news-categories/institute-and-campus&quot;
typeof=&quot;skos:Concept&quot; property=&quot;rdfs:label skos:prefLabel&quot; datatype=&quot;&quot;&gt;Institute and Campus&lt;/a&gt;
</description>
     <pubDate>Thu, 03 May 2018 17:12:31 +0000</pubDate>
```

Figure 2. Georgia Tech News XML Example

and the date on which the article was published, as shown in **Figure 2**.

The Qt XML module allows the user to easily extract such information from the a given XML file. In this project, the QXmlStreamReader from the QtCore module was used to parse the data. Qt offers three different options for parsing an XML file. The QDomDocument from the QtXml module, models the entire XML document as a tree data structure of objects. The QXmlQuery from the QtXmlPatterns module creates an XQuery object to search the XML and return a formatted result set. For this project, however, the QXmlStreamReader was chose due to its efficiency. The QXmlStreamReader parses XML data by reporting an XML document as a stream of tokens. The QXmlStreamReader iterates through a loop and pulls these tokens from the reader, one after another. Similar to how one may parse through a .csv (Comma-

separated values) file, the readNext() method is called, which reads the input stream until it completes the token, then returns the data type of the token, whether it be the title, link, description, or publish date of the news article.

Server with QT

To receive information from the Georgia Tech News Center's RSS feed an internet connection had to be made. The Qt Network Module provides a set of APIs for programming applications that use TCP/IP (Transmission Control Protocol/Internet Protocol). Operations such as requests, cookies, and sending data over HTTP are offered by the Qt Network Module.

The QNetworkAccessManager from the QtNetwork module holds the common configuration and settings for the requests it sends and receives the replies. The QNetworkAccessManager signals the system as to the status of a request so that when the signal is received by the slot the information from the server can be used for further application. In this project, when the QNetworkAccessManager receives that the XML data transmission request has finished the QXMLStreamReader is able to parse the data to pass on to the graphic interface which then returns the data for the user to see. The QtNetwork module also offers other signals such as an incorrect URL, or any other problems which may cause problems with network connections.

Work distribution

Taeyong Shin worked the systematic function of RSS feeds. He worked on parsing the XML files so that they could be passed on to the final system, he developed auto refresh functionality of the RSS feed reader. He used the timer code to update new information every 5 minutes. Also, he was responsible for researching the capabilities of the Qt software and to report to the group as to how to approach the software with our goals in mind.

Files Developed:

news.cpp

rss.cpp

rsshandler.cpp

util.cpp

Woojoo Shin worked on general display and layout features on the GUI. He constructed the graphical user interface (GUI) of programs, organized by their relationships with various operating systems and created the main Window to display the RSS news feed from the Internet. He researched how events were handled by Qt so he could developed the manual refresh functionality. Also he was in charge of analyzing and fixing code errors.

Files Developed:

newsmodel.cpp

decoratednewsmodel.cpp

mainwindow.ui

SooKwon made an outline about how to proceed RSS feed project and he collected materials to use for code. He was the one responsible for bringing the idea of using Qt for our project. He was also responsible for connecting the RSS feed to the designated server (Georgia Tech news center) and collecting the information from the website. He researched the QtNetwork module and developed code so that we could receive XML data from the servers.

Files Developed:

imagecache.cpp

mainwindow.cpp

rsshandler.cpp

Reference

Company, T. Q. (n.d.). Qt | Cross-platform software development for embedded & desktop. Retrieved from https://www.qt.io/

Georgia Institute of Technology. (n.d.). Retrieved from http://www.news.gatech.edu/rss

Google Code Archive – Lon-term storage for Google Code Project Hosting. (n.d.). Retrieved from http://code.google.com/archive/p/feed-reader-lib/