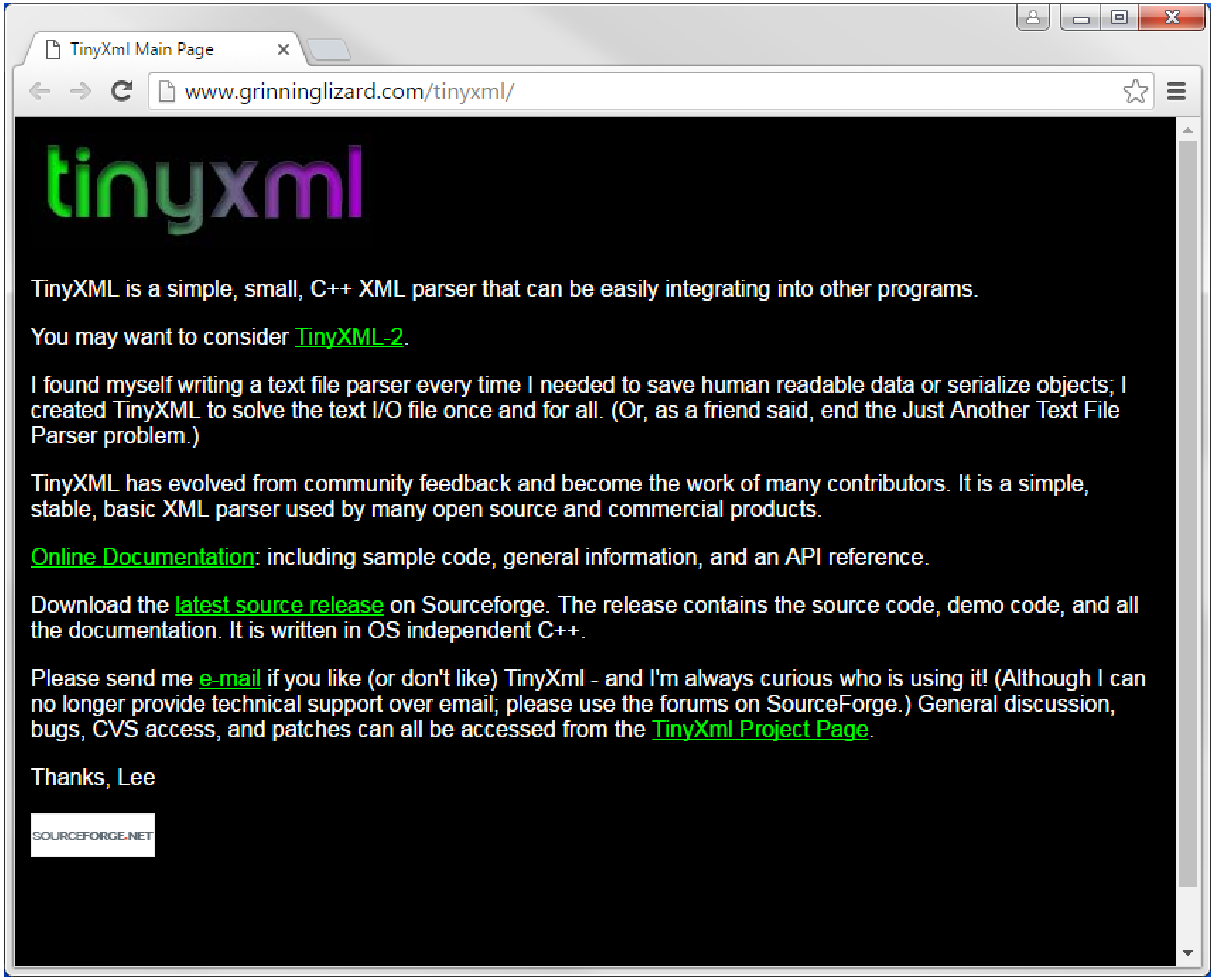
Using an XML Parser

Objective

- You will be able to use a publically available open source parser to convert an XML file into an internal data structure accessible by your C++ code.

tinyxml

http://www.grinninglizard.com/tinyxml/



TinyXML is a simple, small, C++ XML parser that can be easily integrating into other programs.

You may want to consider TinyXML-2.

I found myself writing a text file parser every time I needed to save human readable data or serialize objects; I created TinyXML to solve the text I/O file once and for all. (Or, as a friend said, end the Just Another Text File Parser problem.)

TinyXML has evolved from community feedback and become the work of many contributors. It is a simple, stable, basic XML parser used by many open source and commercial products.

Online Documentation: including sample code, general information, and an API reference.

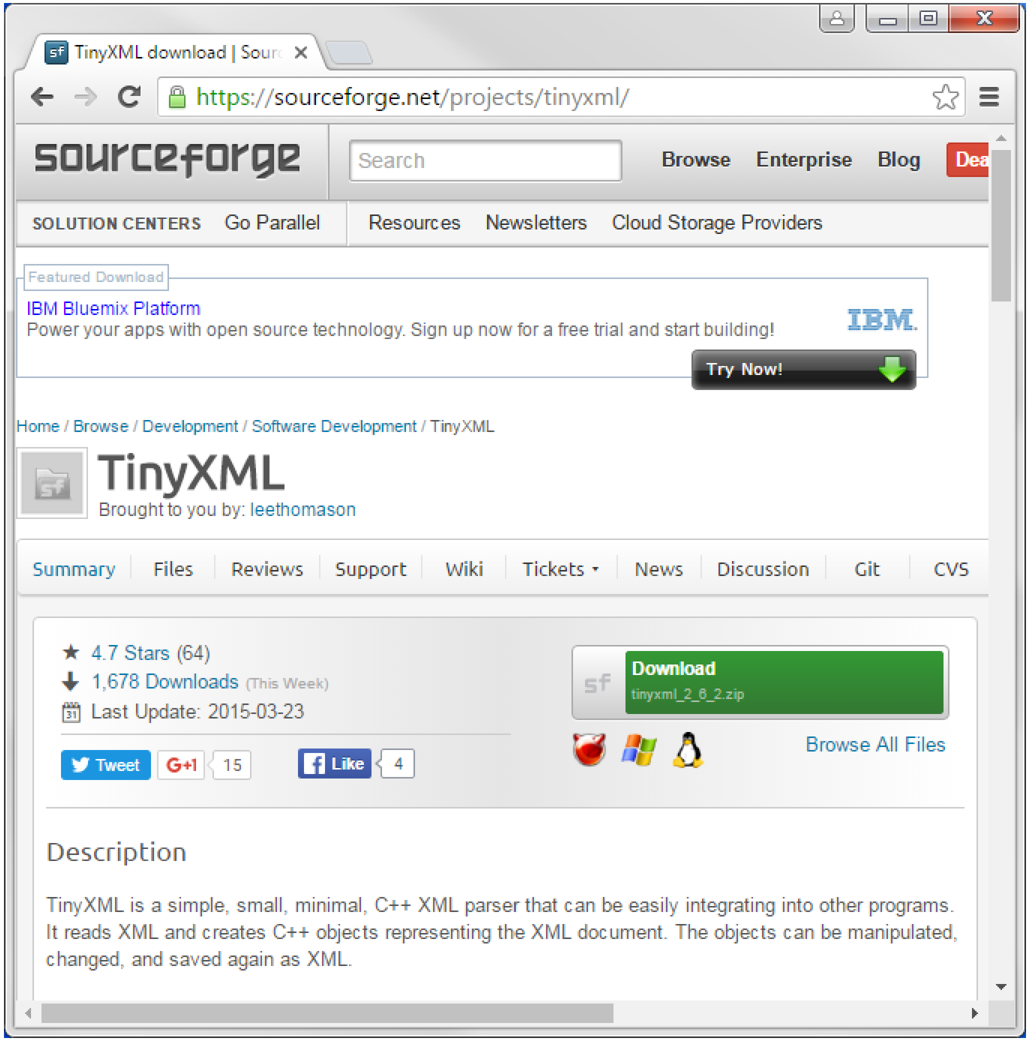
Download the latest source release on Sourceforge. The release contains the source code, demo code, and all the documentation. It is written in OS independent C++.

Please send me e-mail if you like (or don't like) TinyXml - and I'm always curious who is using it! (Although I can no longer provide technical support over email; please use the forums on SourceForge.) General discussion, bugs, CVS access, and patches can all be accessed from the TinyXml Project Page.

Thanks, Lee

Download from sourceforge

http://sourceforge.net/projects/tinyxml/



IBM Bluemix Platform
Power your apps with open source technology. Sign up now for a free trial and start building!

Home / Browse / Development / Software Development / TinyXML

TinyXML

Brought to you by: leethomason

★ 4.7 Stars (64)
1,678 Downloads (This Week)
Last Update: 2015-03-23

Description

TinyXML is a simple, small, minimal, C++ XML parser that can be easily integrating into other programs. It reads XML and creates C++ objects representing the XML document. The objects can be manipulated, changed, and saved again as XML.

Getting Started

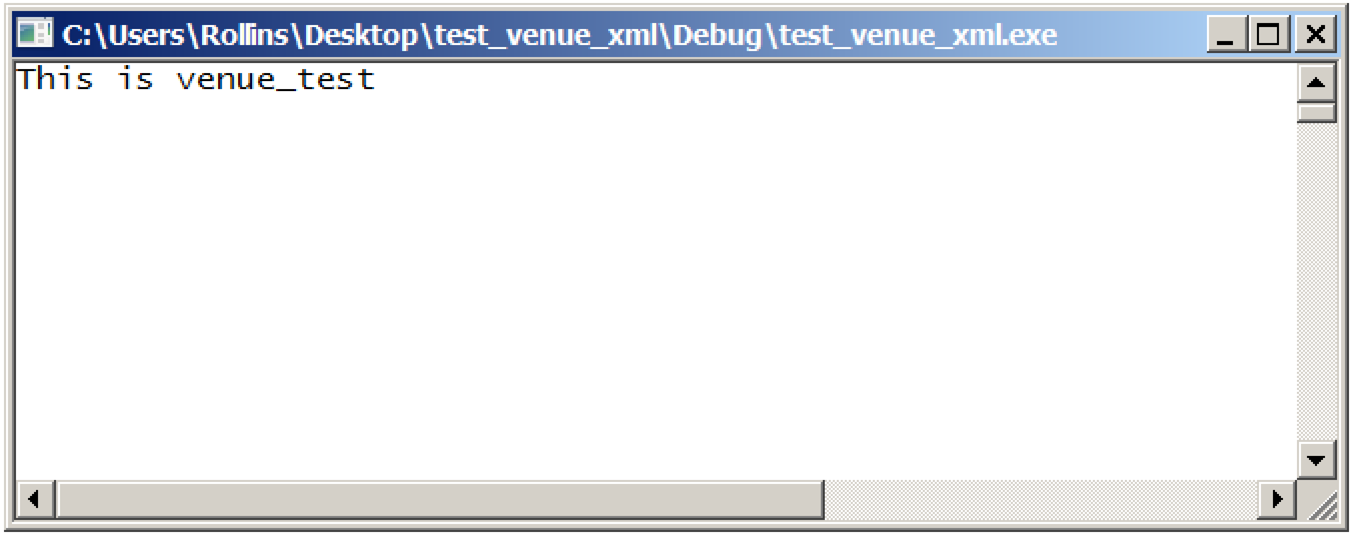
- Create a new empty C++ console application.
  - test_venue_xml

- Add new item test_venue.cpp
- Start with Hello World.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "This is venue_test\n";

    cin.get();
    cin.get();
    return 0;
}
```

Build and Run

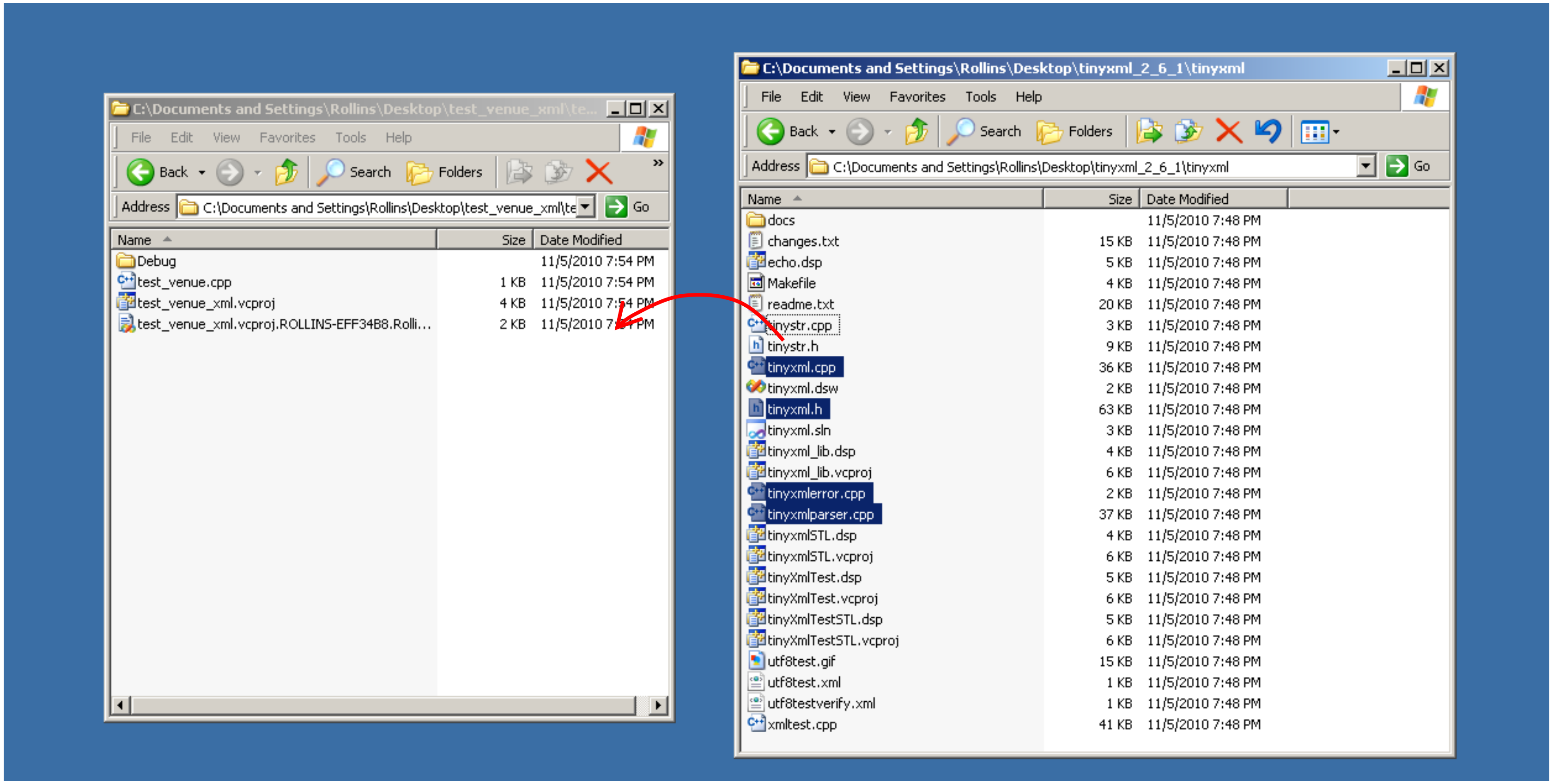
test_venue_xml


C:\Users\Rollins\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
```

Copy Downloaded Source files

- Copy tinyxml source files into the project directory and add to project.
  - tinyxml.cpp, tinyxml.h, tinyxmlerror.cpp, tinyxmlparser.cpp



C:\Documents and Settings\Rollins\Desktop\test_venue_xml\te...

Address C:\Documents and Settings\Rollins\Desktop\test_venue_xml\te

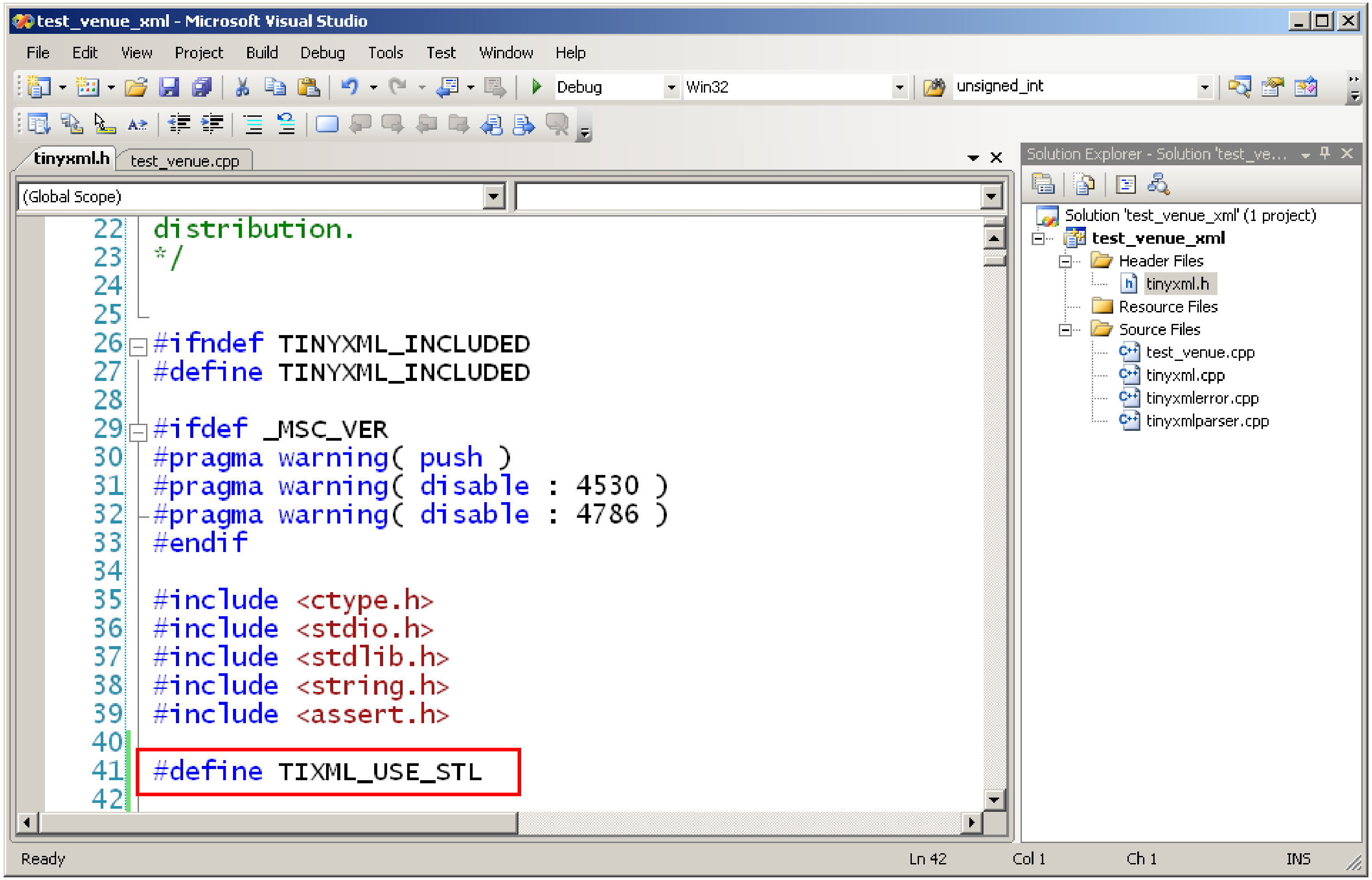
Name	Size	Date Modified
Debug		11/5/2010 7:54 PM
test_venue.cpp	1 KB	11/5/2010 7:54 PM
test_venue_xml.vcproj	4 KB	11/5/2010 7:54 PM
test_venue_xml.vcproj.ROLLINS-EFF34B8.Rolli...	2 KB	11/5/2010 7:54 PM

C:\Documents and Settings\Rollins\Desktop\tinyxml_2_6_1\tinyxml

Address C:\Documents and Settings\Rollins\Desktop\tinyxml_2_6_1\tinyxml

Name	Size	Date Modified
docs		11/5/2010 7:48 PM
changes.txt	15 KB	11/5/2010 7:48 PM
echo.dsp	5 KB	11/5/2010 7:48 PM
Makefile	4 KB	11/5/2010 7:48 PM
readme.txt	20 KB	11/5/2010 7:48 PM
tinystr.cpp	3 KB	11/5/2010 7:48 PM
tinystr.h	9 KB	11/5/2010 7:48 PM
tinyxml.cpp	36 KB	11/5/2010 7:48 PM
tinyxml.dsw	2 KB	11/5/2010 7:48 PM
tinyxml.h	63 KB	11/5/2010 7:48 PM
tinyxml.sln	3 KB	11/5/2010 7:48 PM
tinyxml_lib.dsp	4 KB	11/5/2010 7:48 PM
tinyxml_lib.vcproj	6 KB	11/5/2010 7:48 PM
tinyxmlerror.cpp	2 KB	11/5/2010 7:48 PM
tinyxmlparser.cpp	37 KB	11/5/2010 7:48 PM
tinyxmlSTL.dsp	4 KB	11/5/2010 7:48 PM
tinyxmlSTL.vcproj	6 KB	11/5/2010 7:48 PM
tinyXmlTest.dsp	5 KB	11/5/2010 7:48 PM
tinyXmlTest.vcproj	6 KB	11/5/2010 7:48 PM
tinyXmlTestSTL.dsp	5 KB	11/5/2010 7:48 PM
tinyXmlTestSTL.vcproj	6 KB	11/5/2010 7:48 PM
utf8test.gif	15 KB	11/5/2010 7:48 PM
utf8test.xml	1 KB	11/5/2010 7:48 PM
utf8testverify.xml	1 KB	11/5/2010 7:48 PM
xmltest.cpp	41 KB	11/5/2010 7:48 PM

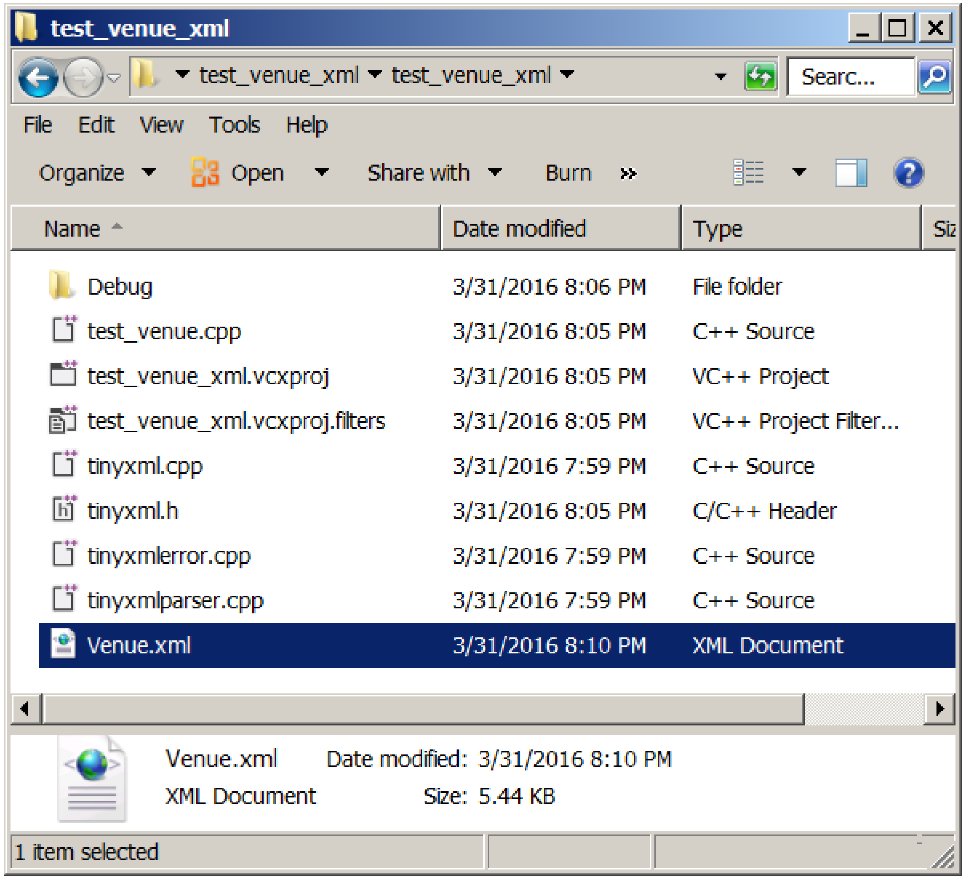
Add a #define to tinyxml.h



```
distribution.
*/


#ifndef TINYXML_INCLUDED
#define TINYXML_INCLUDED

#ifdef _MSC_VER
#pragma warning( push )
#pragma warning( disable : 4530 )
#pragma warning( disable : 4786 )
#endif

#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>

#define TIXML_USE_STL

```

venue.xml

- Download venue.xml
- Copy venue.xml into the project folder


test_venue_xml
test_venue_xml ▾ test_venue_xml ▾
Searc...
File Edit View Tools Help
Organize ▾ Open ▾ Share with ▾ Burn »

Name	Date modified	Type	Siz
Debug	3/31/2016 8:06 PM	File folder	
test_venue.cpp	3/31/2016 8:05 PM	C++ Source	
test_venue_xml.vcxproj	3/31/2016 8:05 PM	VC++ Project	
test_venue_xml.vcxproj.filters	3/31/2016 8:05 PM	VC++ Project Filter...	
tinyxml.cpp	3/31/2016 7:59 PM	C++ Source	
tinyxml.h	3/31/2016 8:05 PM	C/C++ Header	
tinyxmlerror.cpp	3/31/2016 7:59 PM	C++ Source	
tinyxmlparser.cpp	3/31/2016 7:59 PM	C++ Source	
Venue.xml	3/31/2016 8:10 PM	XML Document	

Venue.xml Date modified: 3/31/2016 8:10 PM
XML Document Size: 5.44 KB

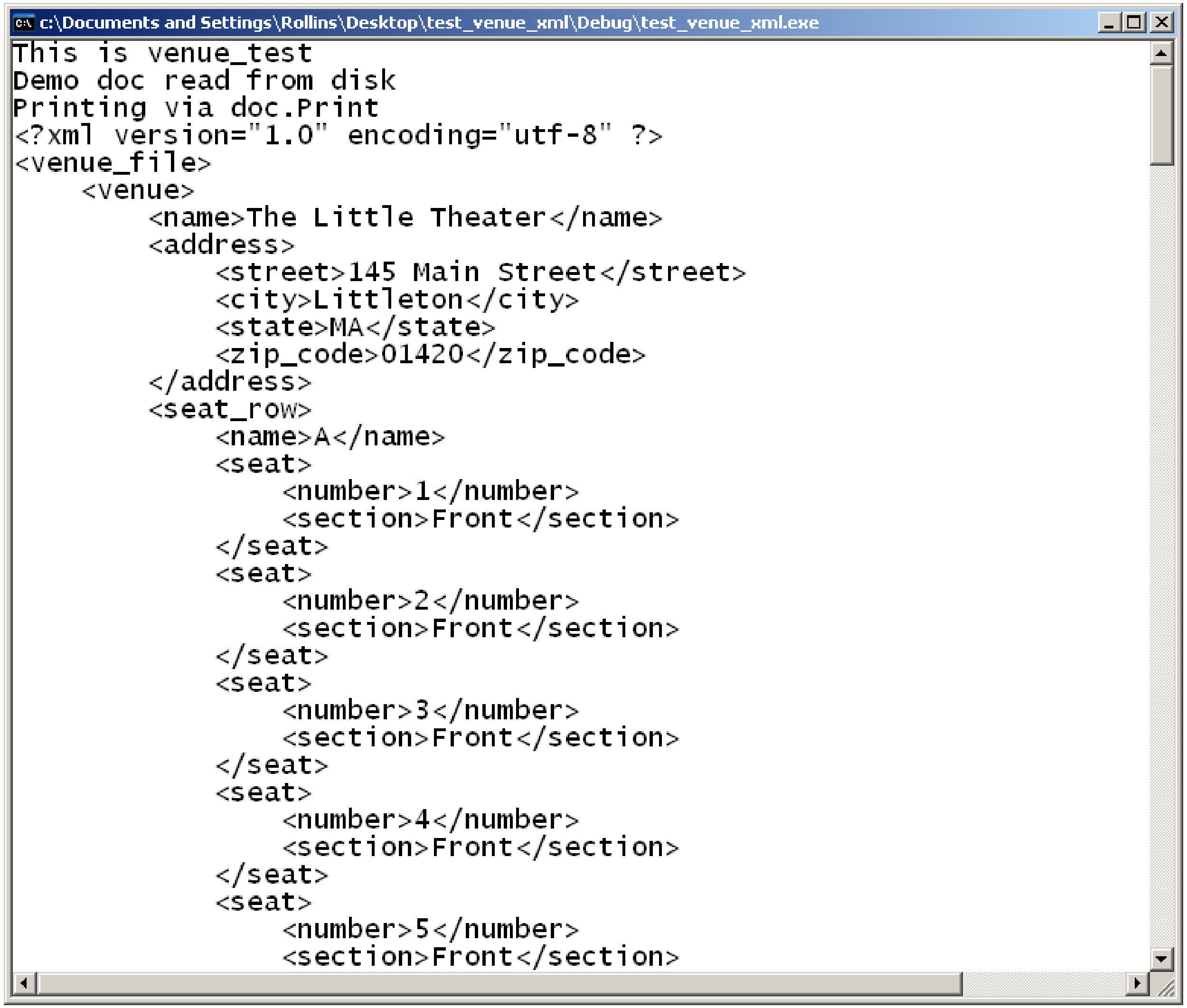
1 item selected

test_venue.cpp

```
#include <iostream>
#include <string>
#include <cassert>
#include "tinyxml.h"

using namespace std;
int main()
{
    cout << "This is venue_test\n";
    string filename = "Venue.xml";
    TiXmlDocument doc(filename);

    bool loadOkay = doc.LoadFile();
    if (!loadOkay)
    {
        cout << "Could not load file " << filename << endl;
        cout << "Error='" << doc.ErrorDesc() <<"'. Exiting.\n";
        cin.get();
        exit( 1 );
    }
    cout << filename << " read from disk " << endl;
    cout << "Printing via doc.Print \n";
    doc.Print( stdout );
    cin.get();
    return 0;
}
```

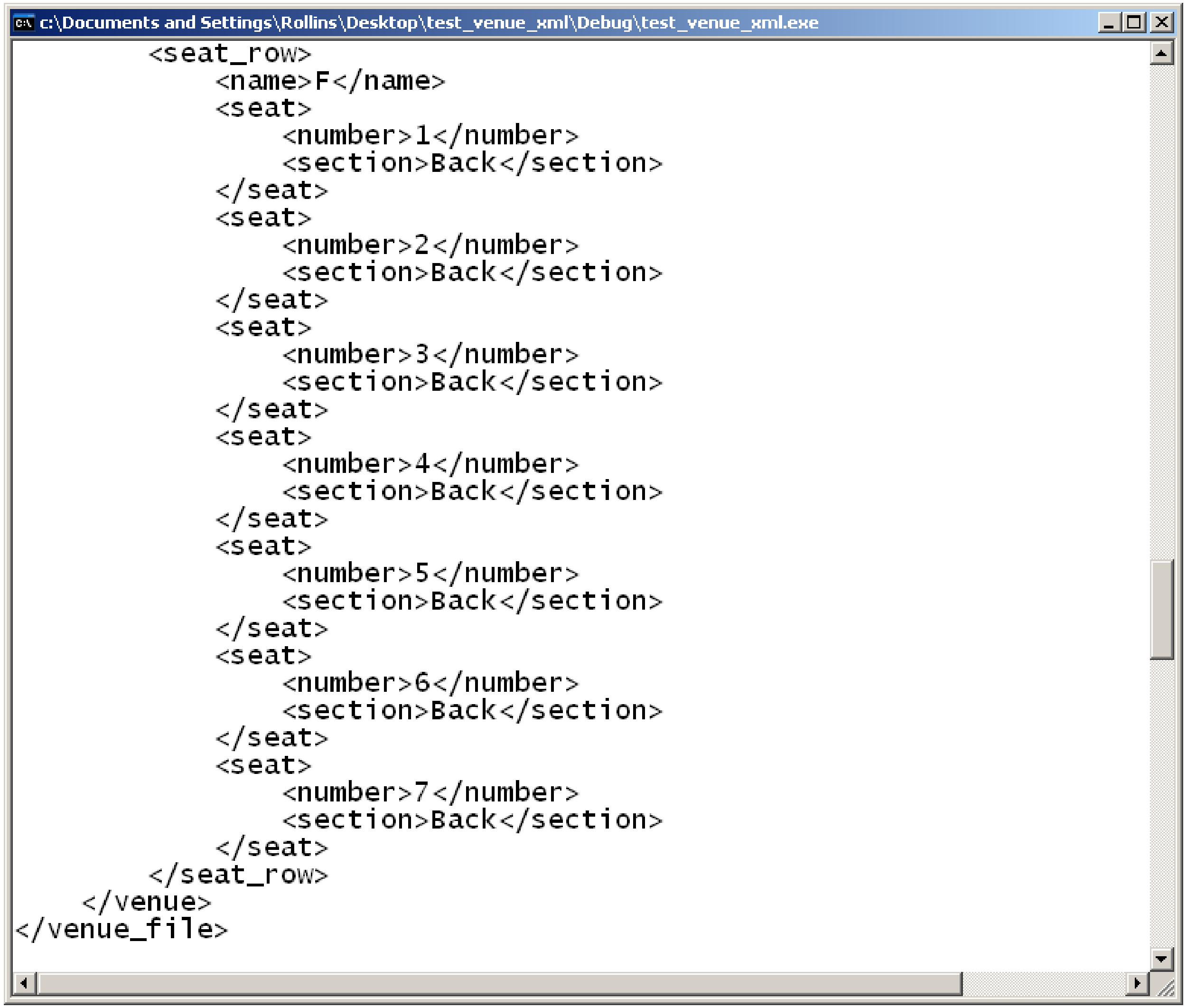
Build and Run

Initial Test Run


c:\Documents and Settings\Rollins\Desktop\test_venue_xml\Debug\test_venue_xml.exe

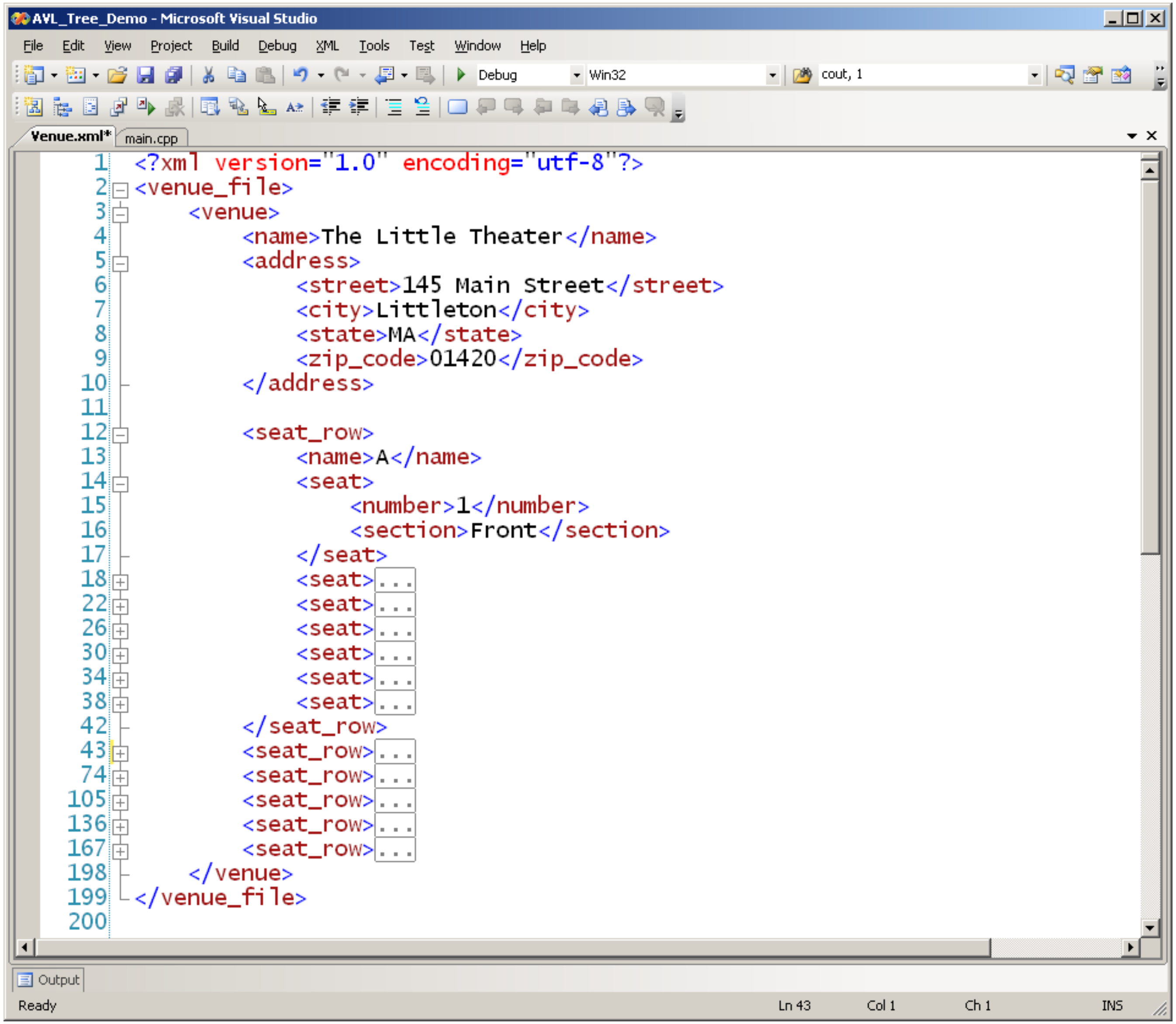
```
This is venue_test
Demo doc read from disk
Printing via doc.Print
<?xml version="1.0" encoding="utf-8" ?>
<venue_file>
    <venue>
        <name>The Little Theater</name>
        <address>
            <street>145 Main Street</street>
            <city>Littleton</city>
            <state>MA</state>
            <zip_code>01420</zip_code>
        </address>
        <seat_row>
            <name>A</name>
            <seat>
                <number>1</number>
                <section>Front</section>
            </seat>
            <seat>
                <number>2</number>
                <section>Front</section>
            </seat>
            <seat>
                <number>3</number>
                <section>Front</section>
            </seat>
            <seat>
                <number>4</number>
                <section>Front</section>
            </seat>
            <seat>
                <number>5</number>
                <section>Front</section>
```

Initial Test Run (continued)


c:\Documents and Settings\Rollins\Desktop\test_venue_xml\Debug\test_venue_xml.exe

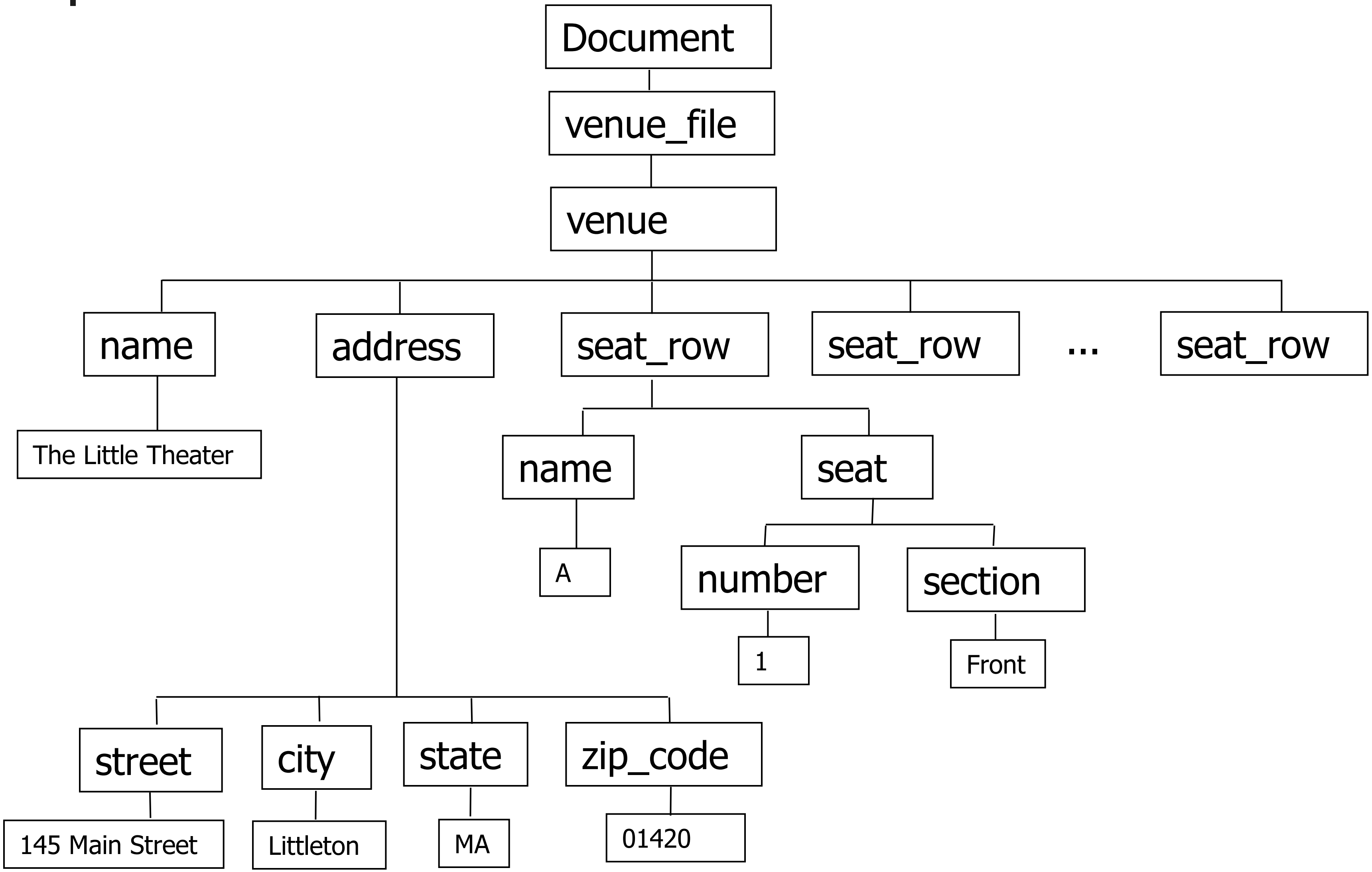
```
        <seat_row>
            <name>F</name>
            <seat>
                <number>1</number>
                <section>Back</section>
            </seat>
            <seat>
                <number>2</number>
                <section>Back</section>
            </seat>
            <seat>
                <number>3</number>
                <section>Back</section>
            </seat>
            <seat>
                <number>4</number>
                <section>Back</section>
            </seat>
            <seat>
                <number>5</number>
                <section>Back</section>
            </seat>
            <seat>
                <number>6</number>
                <section>Back</section>
            </seat>
            <seat>
                <number>7</number>
                <section>Back</section>
            </seat>
        </seat_row>
    </venue>
</venue_file>
```

Venue.xml


AVL_Tree_Demo - Microsoft Visual Studio
File Edit View Project Build Debug XML Tools Test Window Help
Debug
Win32
cout, 1
Venue.xml*
main.cpp
Output
Ready
Ln 43
Col 1
Ch 1
INS

```
<?xml version="1.0" encoding="utf-8"?>
<venue_file>
    <venue>
        <name>The Little Theater</name>
        <address>
            <street>145 Main Street</street>
            <city>Littleton</city>
            <state>MA</state>
            <zip_code>01420</zip_code>
        </address>

        <seat_row>
            <name>A</name>
            <seat>
                <number>1</number>
                <section>Front</section>
            </seat>
            <seat>...
            <seat>...
            <seat>...
            <seat>...
            <seat>...
            <seat>...
        </seat_row>
        <seat_row>...
        <seat_row>...
        <seat_row>...
        <seat_row>...
        <seat_row>...
    </venue>
</venue_file>
```

Venue.xml DOM



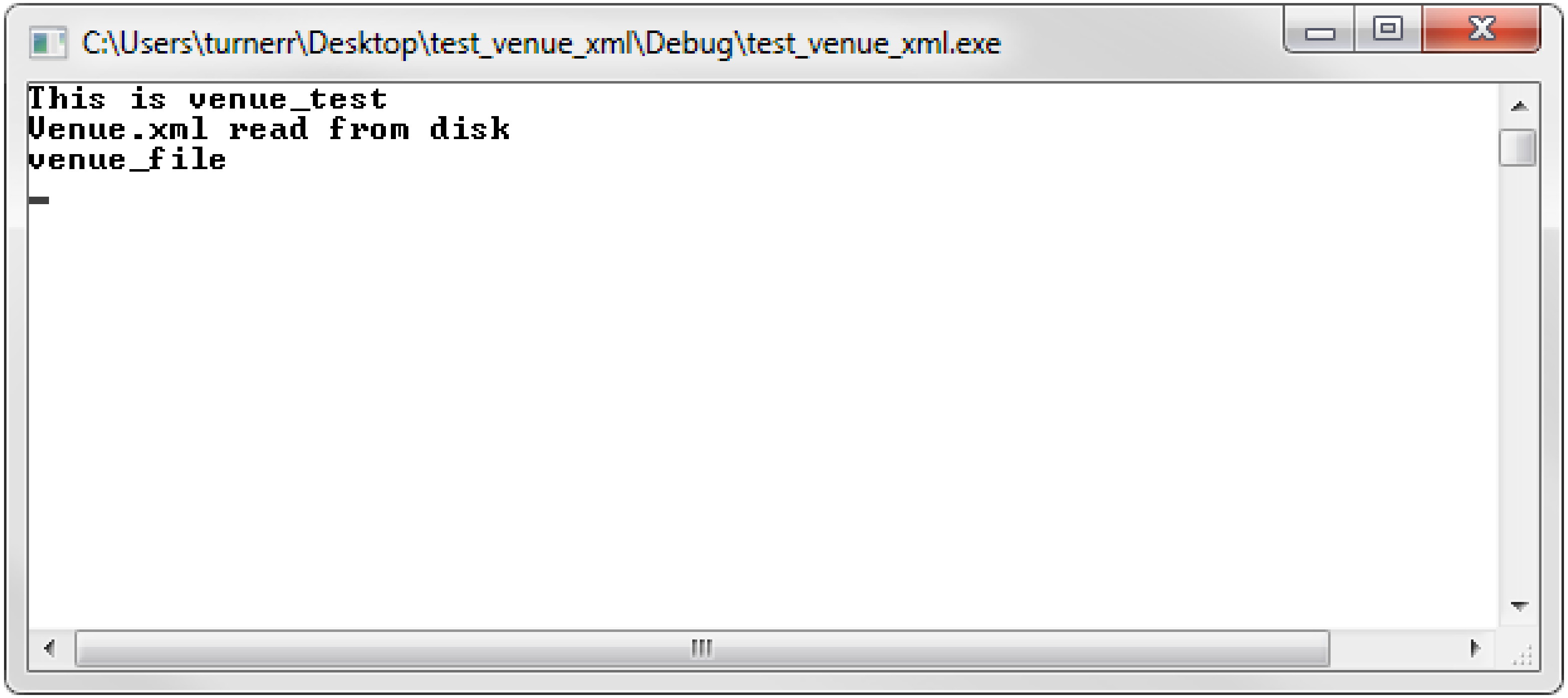
Reading Nodes from the DOM

- Add at end of main():

```
    TiXmlNode* venue_file_node = doc.FirstChild("venue_file");
    assert(venue_file_node != 0);
    cout << venue_file_node->Value() << endl;

    cin.get();
    return 0;
}
```

- Comment out doc.Print

The First Node

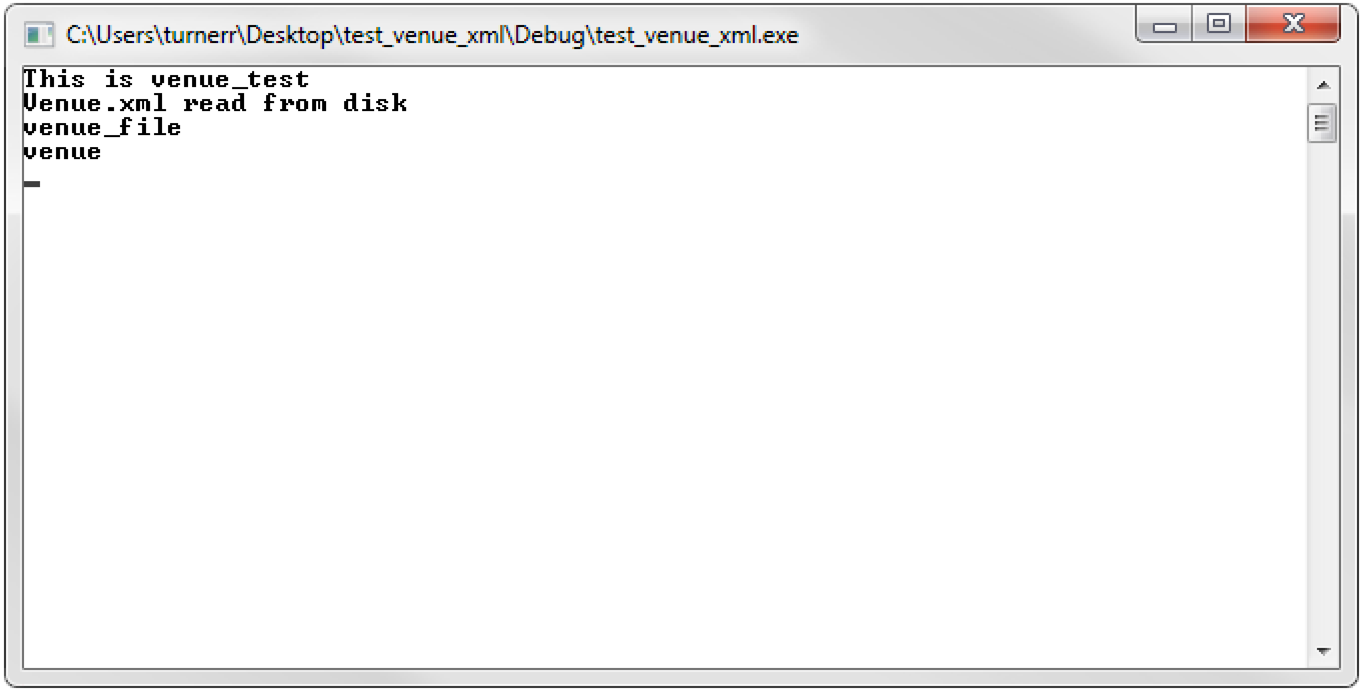

C:\Users\turnerr\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
Venue.xml read from disk
venue_file

_
```

Navigating the DOM

```
    TiXmlNode* venue_file_node = doc.FirstChild("venue_file");
    assert(venue_file_node != 0);
    cout << venue_file_node->Value() << endl;

    TiXmlNode* venue_node = venue_file_node->FirstChild();
    assert(venue_node != 0);
    cout << venue_node->Value() << endl;

    cin.get();
    return 0;
}
```

Second Node



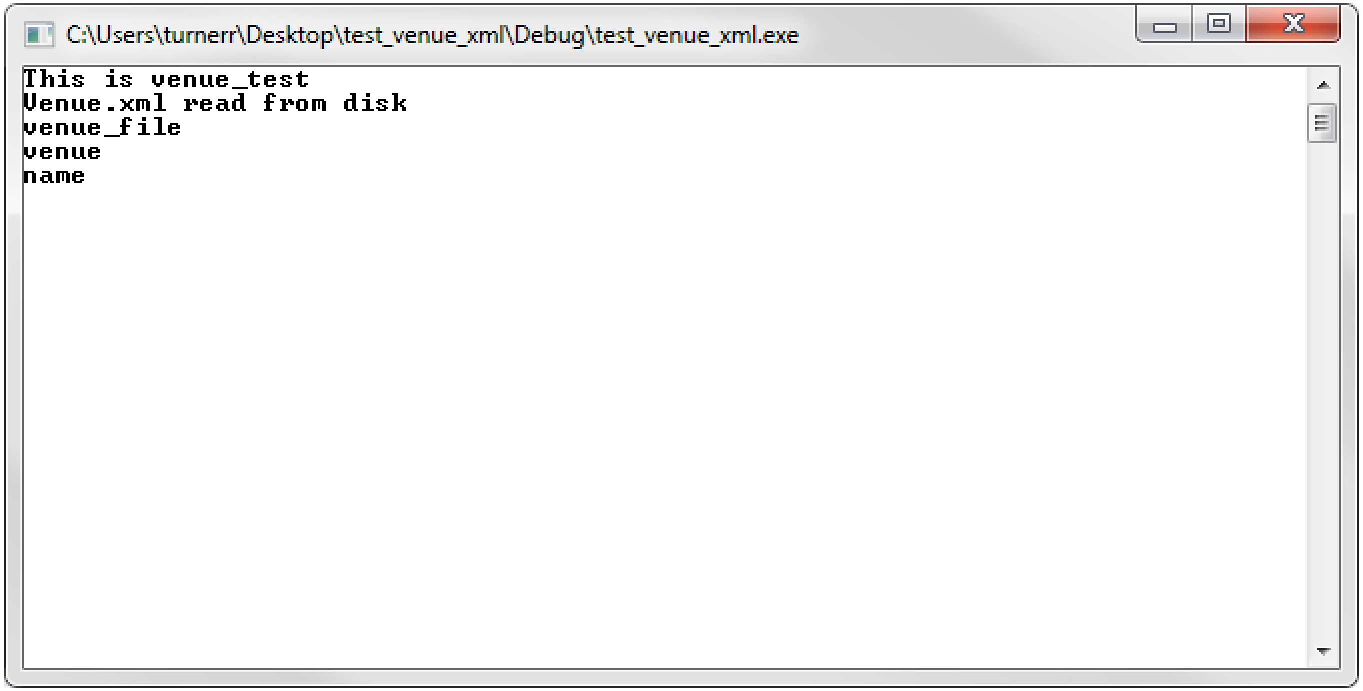
C:\Users\turnerr\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
Venue.xml read from disk
venue_file
venue
```

More Nodes

```
    TiXmlNode* venue_file_node = doc.FirstChild("venue_file");
    assert(venue_file_node != 0);
    cout << venue_file_node->Value() << endl;

    TiXmlNode* venue_node = venue_file_node->FirstChild();
    assert(venue_node != 0);
    cout << venue_node->Value() << endl;

    TiXmlNode* name_node = venue_node->FirstChild();
    assert(name_node != 0);
    cout << name_node->Value() << endl;

    cin.get();
    return 0;
}
```

More Nodes

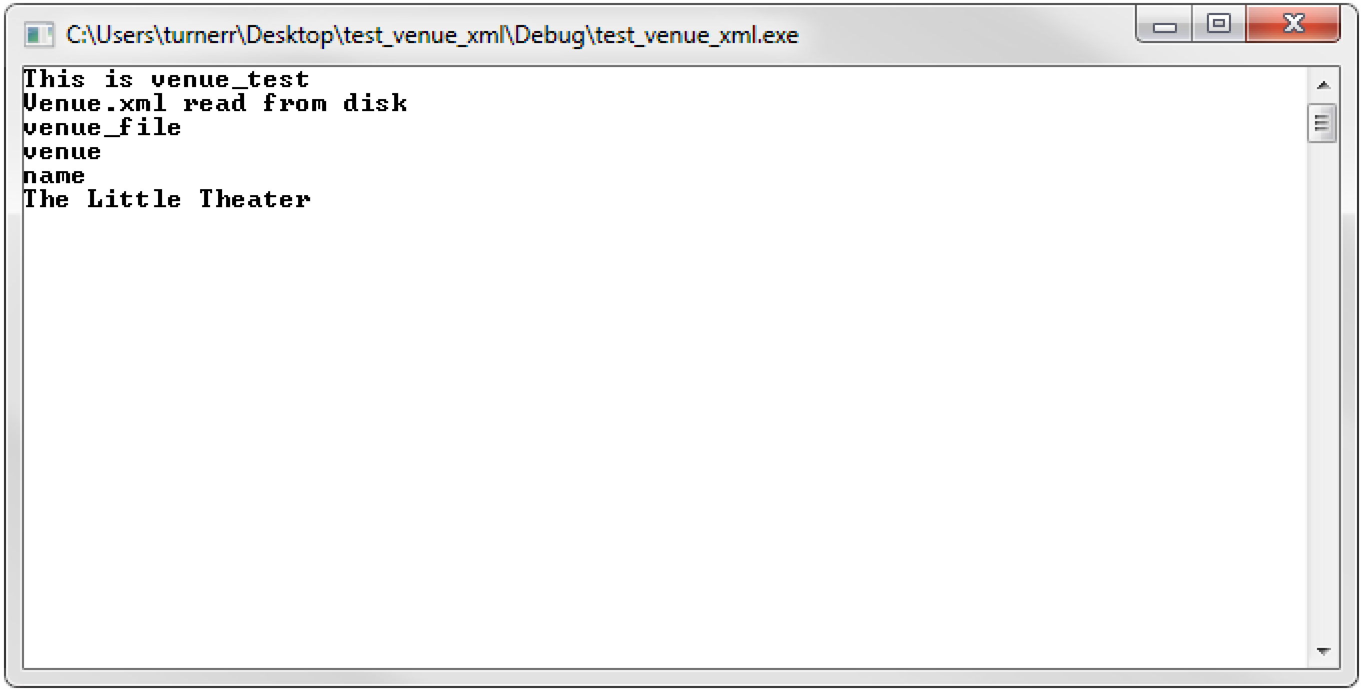

C:\Users\turnerr\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
Venue.xml read from disk
venue_file
venue
name
```

A Text Value

```
    TiXmlNode* venue_file_node = doc.FirstChild("venue_file");
    assert(venue_file_node != 0);
    cout << venue_file_node->Value() << endl;

    TiXmlNode* venue_node = venue_file_node->FirstChild();
    assert(venue_node != 0);
    cout << venue_node->Value() << endl;

    TiXmlNode* name_node = venue_node->FirstChild();
    assert(name_node != 0);
    cout << name_node->Value() << endl;

    TiXmlNode* name_text_node = name_node->FirstChild();
    assert(name_text_node != 0);
    cout << name_text_node->Value() << endl;

    cin.get();
    return 0;
}
```

A Text Value



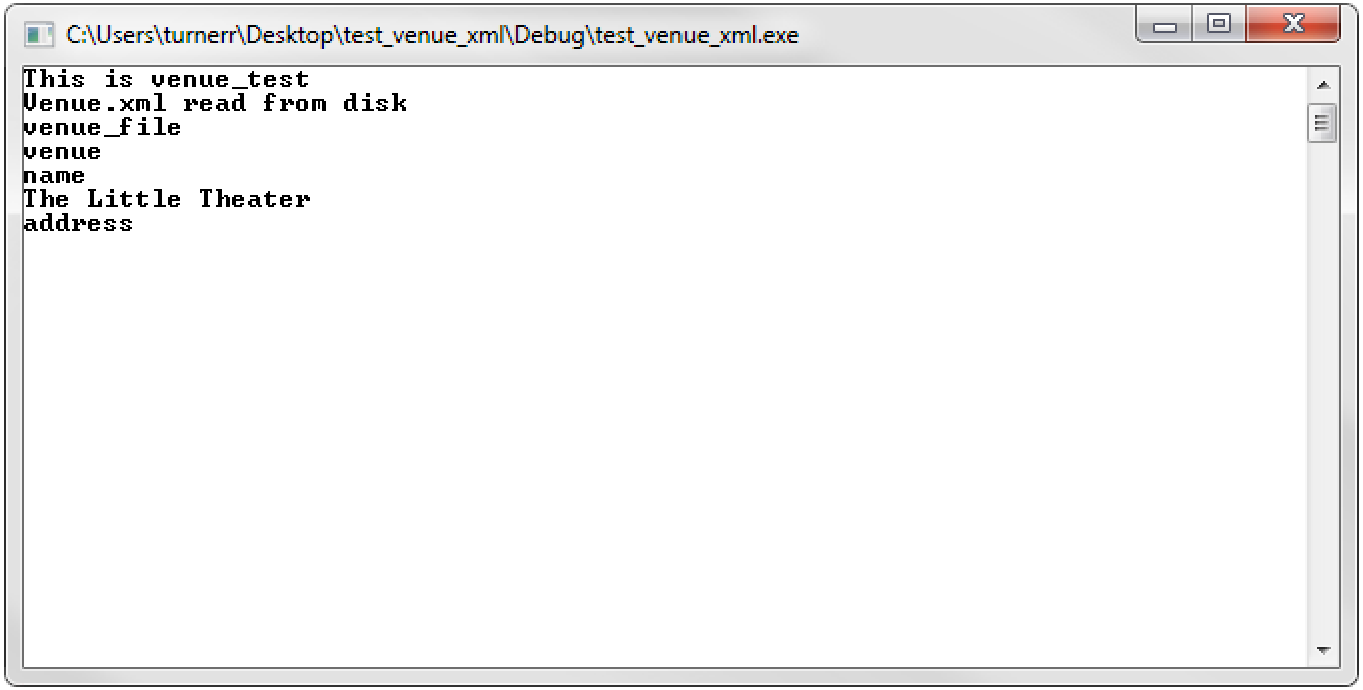
C:\Users\turnerr\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
Venue.xml read from disk
venue_file
venue
name
The Little Theater
```

The Address Node

```
    TiXmlNode* address_node = name_node->NextSibling();
    assert(address_node != 0);
    cout << address_node->Value() << endl;

    cin.get();
    return 0;
}
```

The Address Node

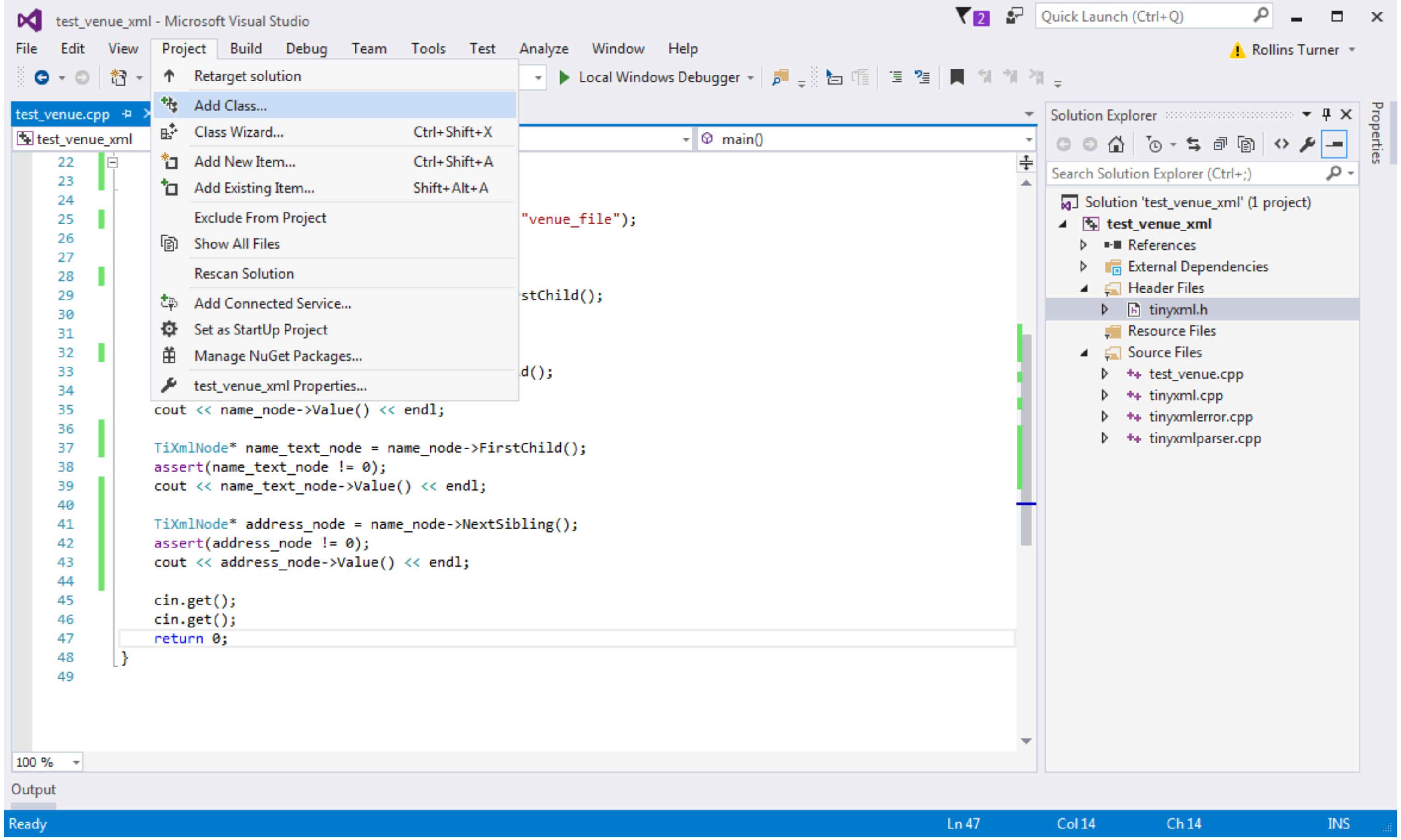

C:\Users\turnerr\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
Venue.xml read from disk
venue_file
venue
name
The Little Theater
address
```

Adding a Boundary Class

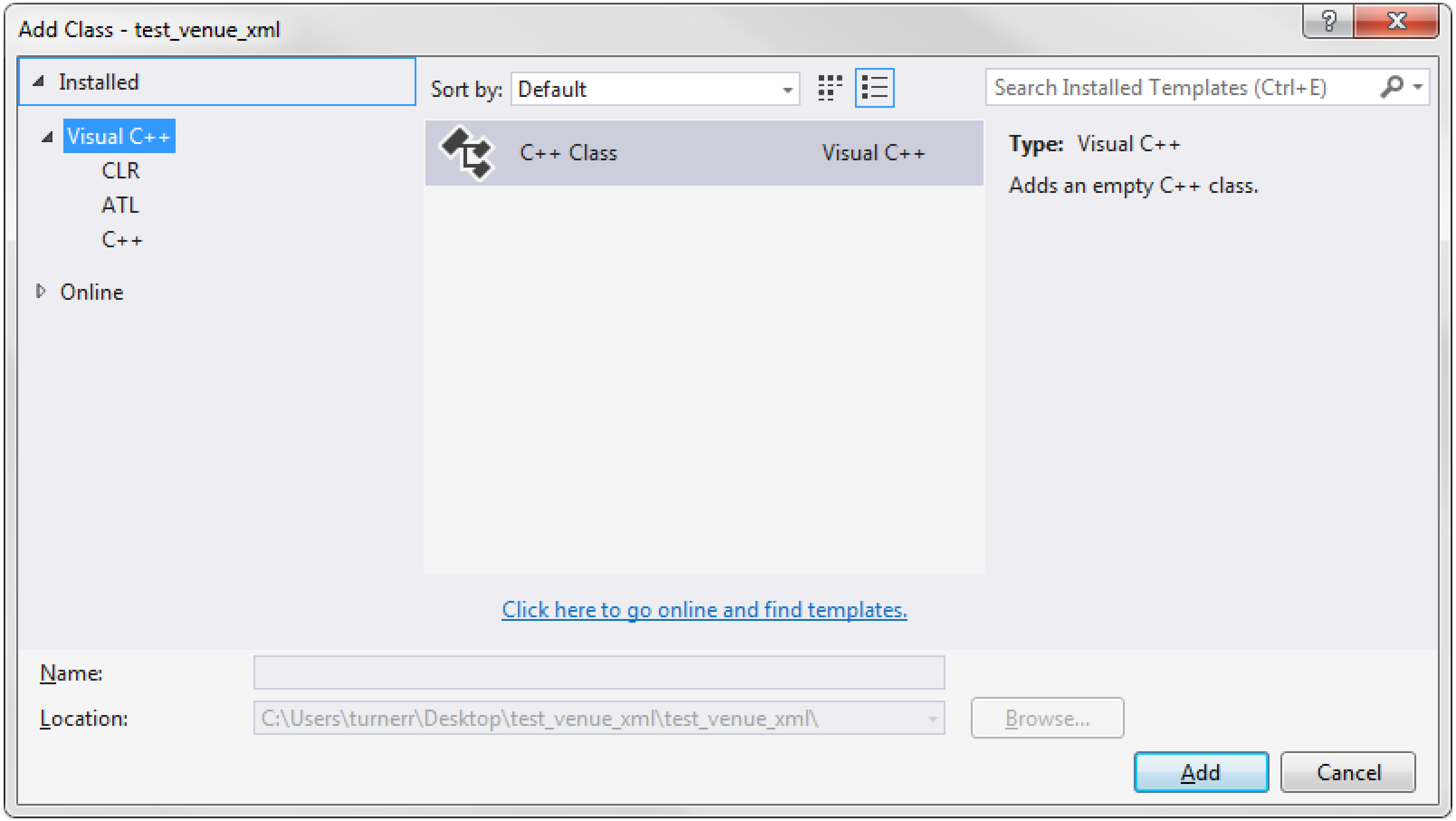
- main() is getter verbose!
- Let's put this code into a boundary class.
- Add class Venue_from_Xml
  - static method Get_Venue(TiXmlNode* venue_node)

Add Class to Project



```
                          "venue_file");

                          stChild();

                          d();

    cout << name_node->Value() << endl;

    TiXmlNode* name_text_node = name_node->FirstChild();
    assert(name_text_node != 0);
    cout << name_text_node->Value() << endl;

    TiXmlNode* address_node = name_node->NextSibling();
    assert(address_node != 0);
    cout << address_node->Value() << endl;

    cin.get();
    cin.get();
    return 0;
}
```

Add Class to Project



Add Class - test_venue_xml

Sort by: Default

Search Installed Templates (Ctrl+E)

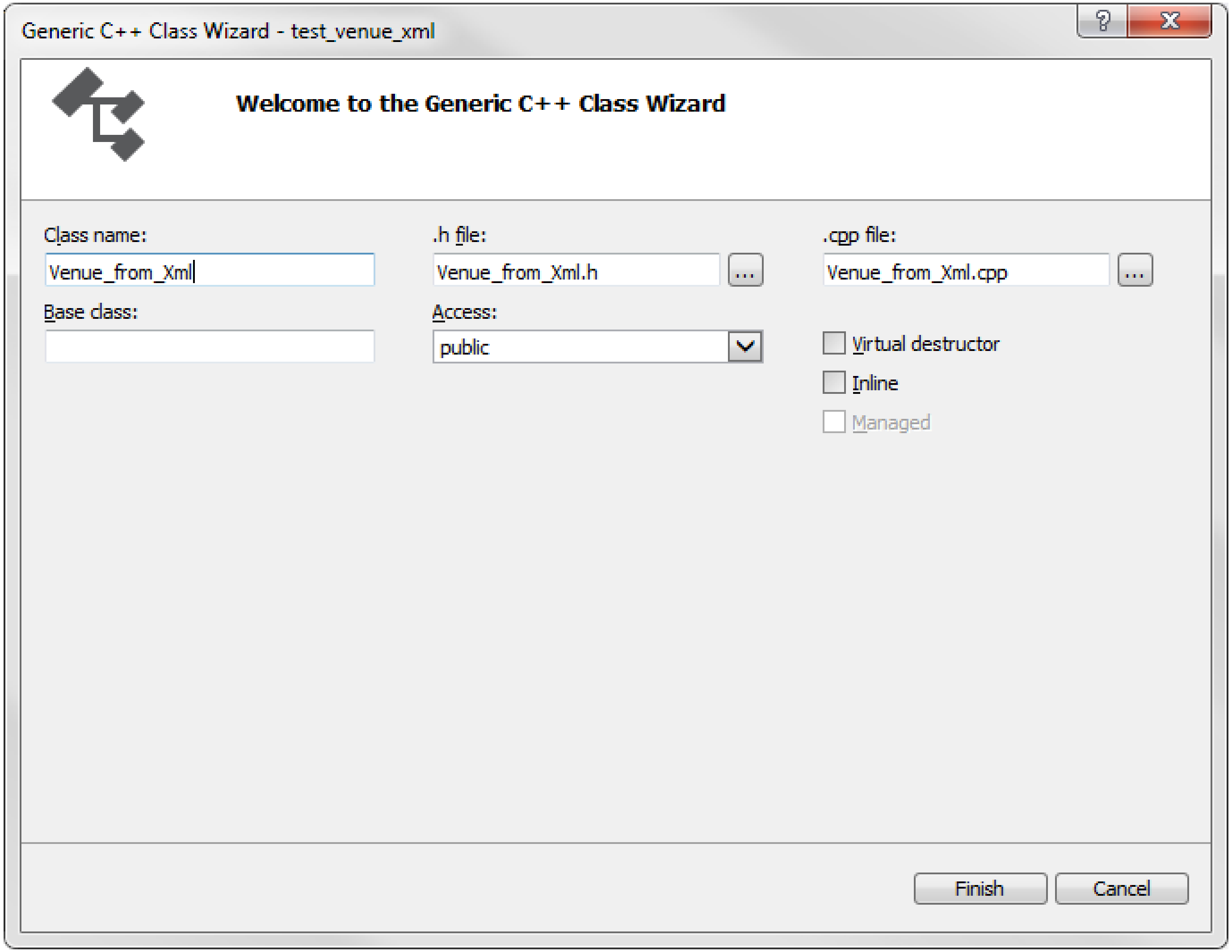
Type: Visual C++

Adds an empty C++ class.

Name:

Location: C:\Users\turnerr\Desktop\test_venue_xml\test_venue_xml\

Add Class to Project



Generic C++ Class Wizard - test_venue_xml

Welcome to the Generic C++ Class Wizard

Class name: Venue_from_Xml

.h file: Venue_from_Xml.h

.cpp file: Venue_from_Xml.cpp

Base class:

Access: public

□ Virtual destructor

□ Inline

□ Managed

Venue_from_Xml.h

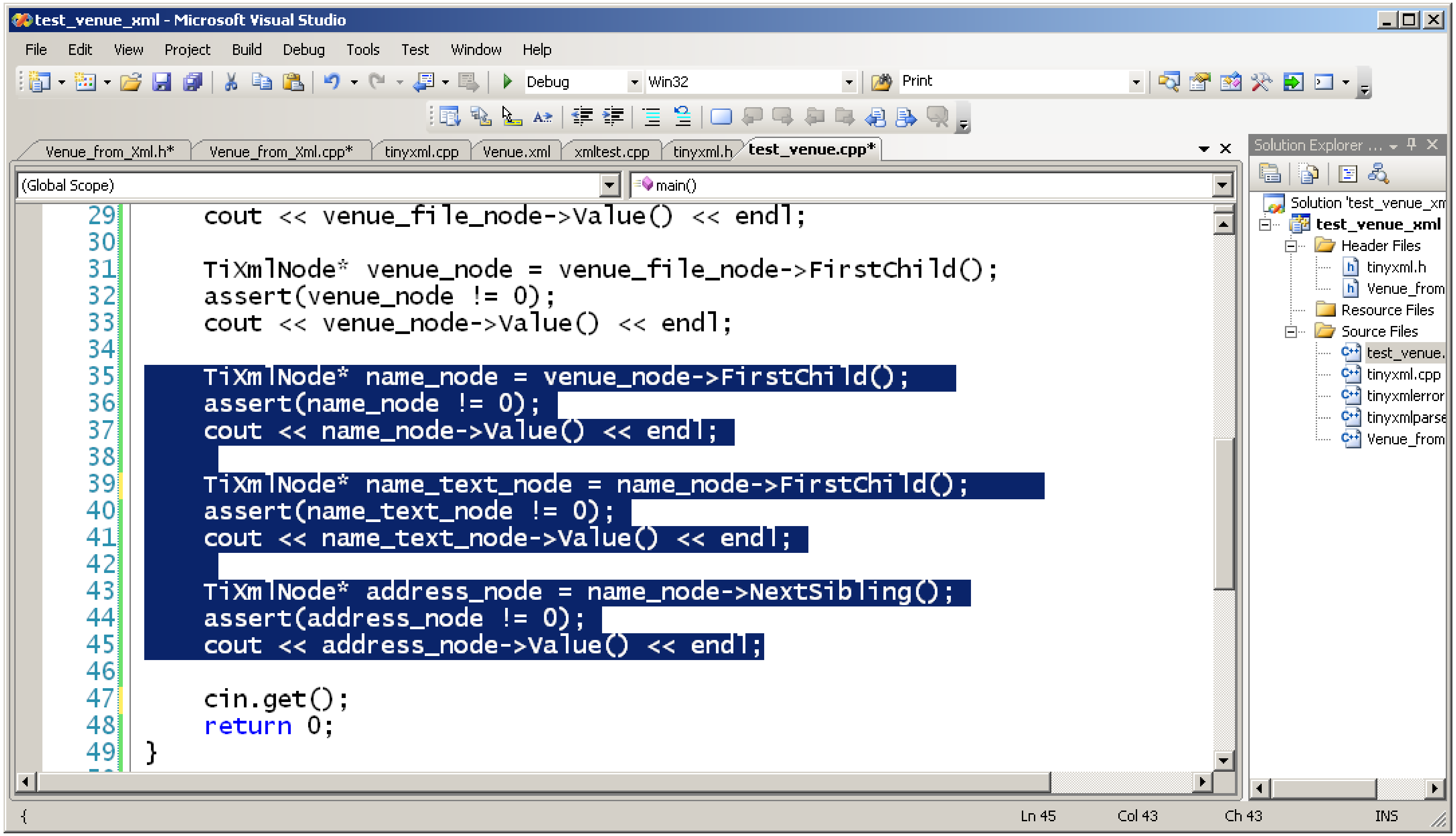
```
#pragma once
#include "tinyxml.h"

class Venue_from_Xml
{

public:
   static void Get_Venue(TiXmlNode* venue_node);
};
```

Venue_from_Xml.cpp

```
#include <iostream>
#include "Venue_from_Xml.h"
#include "tinyxml.h"
using namespace std;

void Venue_from_Xml::Get_Venue(TiXmlNode* venue_node)
{

}
```

Move Code from main()

Add #include "Venue_from_Xml.h"



```
    cout << venue_file_node->Value() << endl;

    TiXmlNode* venue_node = venue_file_node->FirstChild();
    assert(venue_node != 0);
    cout << venue_node->Value() << endl;

    TiXmlNode* name_node = venue_node->FirstChild();
    assert(name_node != 0);
    cout << name_node->Value() << endl;

    TiXmlNode* name_text_node = name_node->FirstChild();
    assert(name_text_node != 0);
    cout << name_text_node->Value() << endl;

    TiXmlNode* address_node = name_node->NextSibling();
    assert(address_node != 0);
    cout << address_node->Value() << endl;

    cin.get();
    return 0;
}
```

Move these statements to Venue_from_Xml.cpp

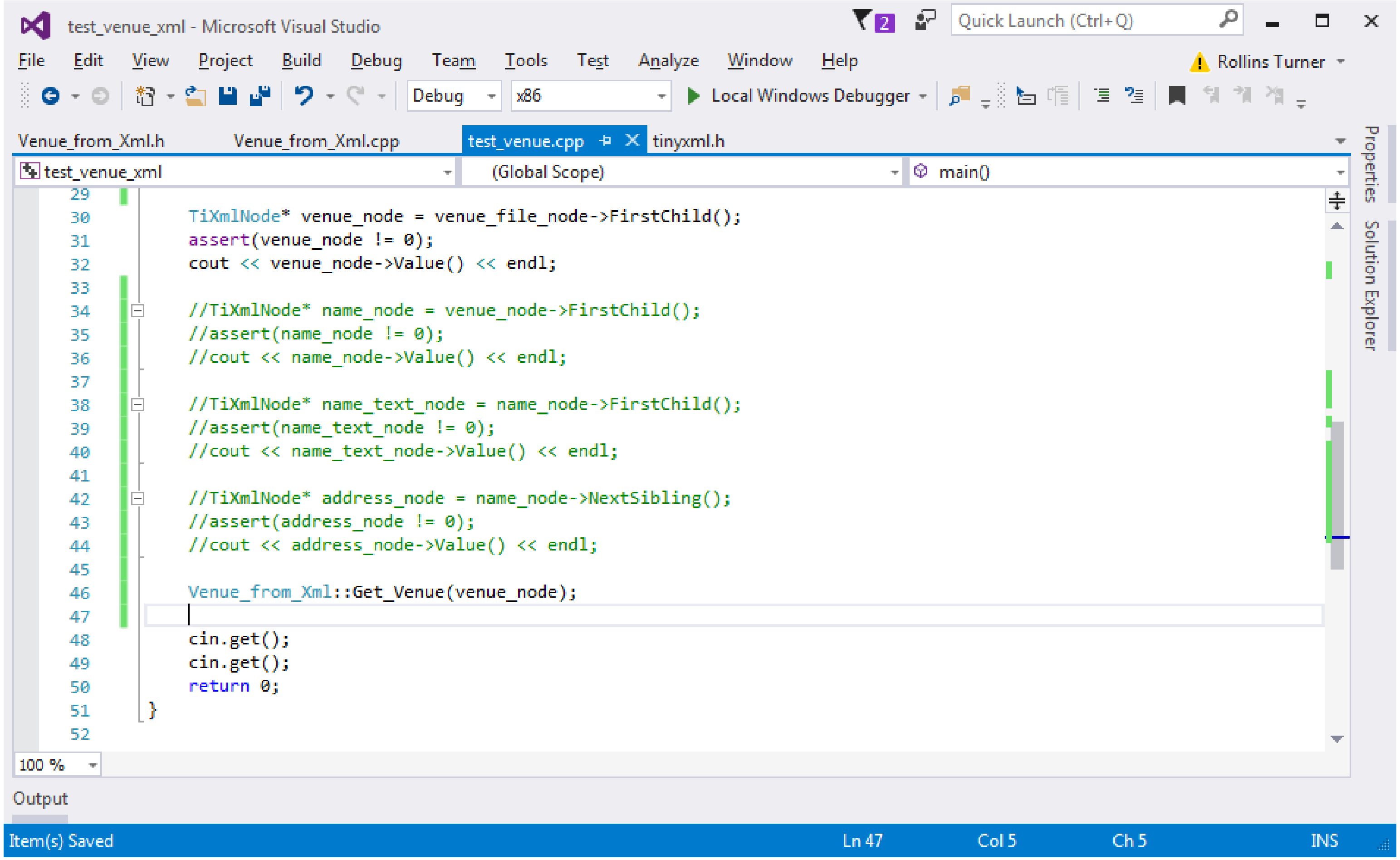
Replace with call to Venue_from_Xml::Get_Venue(venue_node)

Venue_from_Xml.cpp

```
#include <iostream>
#include "Venue_from_Xml.h"
#include "tinyxml.h"
using namespace std;

void Venue_from_Xml::Get_Venue(TiXmlNode* venue_node)
{
    TiXmlNode* name_node = venue_node->FirstChild();
    assert(name_node != 0);
    cout << name_node->Value() << endl;

    TiXmlNode* name_text_node = name_node->FirstChild();
    assert(name_text_node != 0);
    cout << name_text_node->Value() << endl;

    TiXmlNode* address_node = name_node->NextSibling();
    assert(address_node != 0);
    cout << address_node->Value() << endl;
}
```

test_venue.cpp

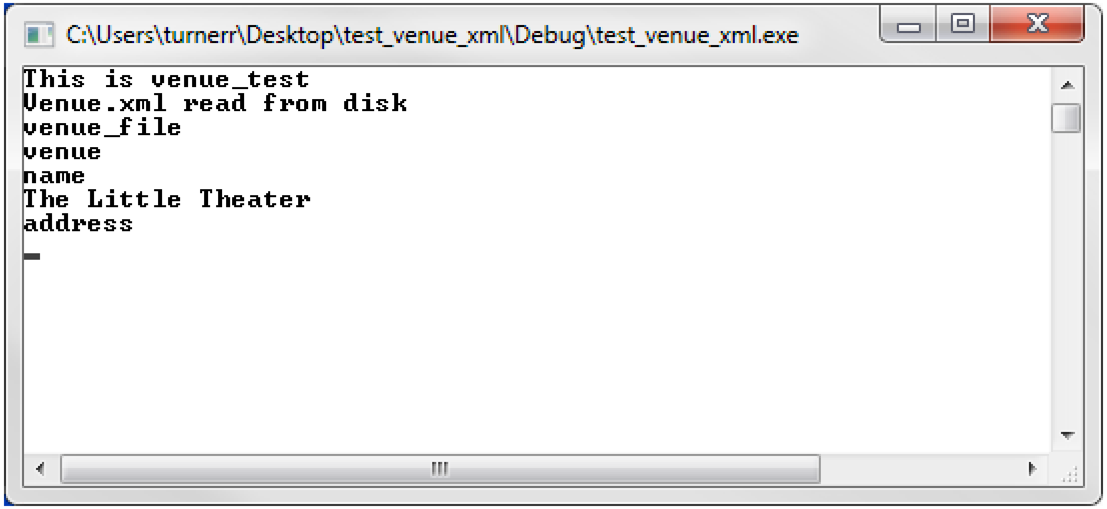
Add `#include "Venue_from_Xml.h"` at top.



```
    TiXmlNode* venue_node = venue_file_node->FirstChild();
    assert(venue_node != 0);
    cout << venue_node->Value() << endl;

    //TiXmlNode* name_node = venue_node->FirstChild();
    //assert(name_node != 0);
    //cout << name_node->Value() << endl;

    //TiXmlNode* name_text_node = name_node->FirstChild();
    //assert(name_text_node != 0);
    //cout << name_text_node->Value() << endl;

    //TiXmlNode* address_node = name_node->NextSibling();
    //assert(address_node != 0);
    //cout << address_node->Value() << endl;

    Venue_from_Xml::Get_Venue(venue_node);

    cin.get();
    cin.get();
    return 0;
}
```

Build and run

Program Running



C:\Users\turnerr\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
Venue.xml read from disk
venue_file
venue
name
The Little Theater
address

```

Works the same!

Add Get_Address

```
void Venue_from_Xml::Get_Address(TiXmlNode* address_node)
{
    TiXmlNode* street_node = address_node->FirstChild();
    assert(street_node != 0);
    cout << street_node->FirstChild()->Value() << endl;

    TiXmlNode* city_node = street_node->NextSibling();
    assert(city_node != 0);
    cout << city_node->FirstChild()->Value() << endl;

    TiXmlNode* state_node = city_node->NextSibling();
    assert(state_node != 0);
    cout << state_node->FirstChild()->Value() << endl;

    TiXmlNode* zip_code = state_node->NextSibling();
    assert(zip_code != 0);
    cout << zip_code->FirstChild()->Value() << endl;
}
```

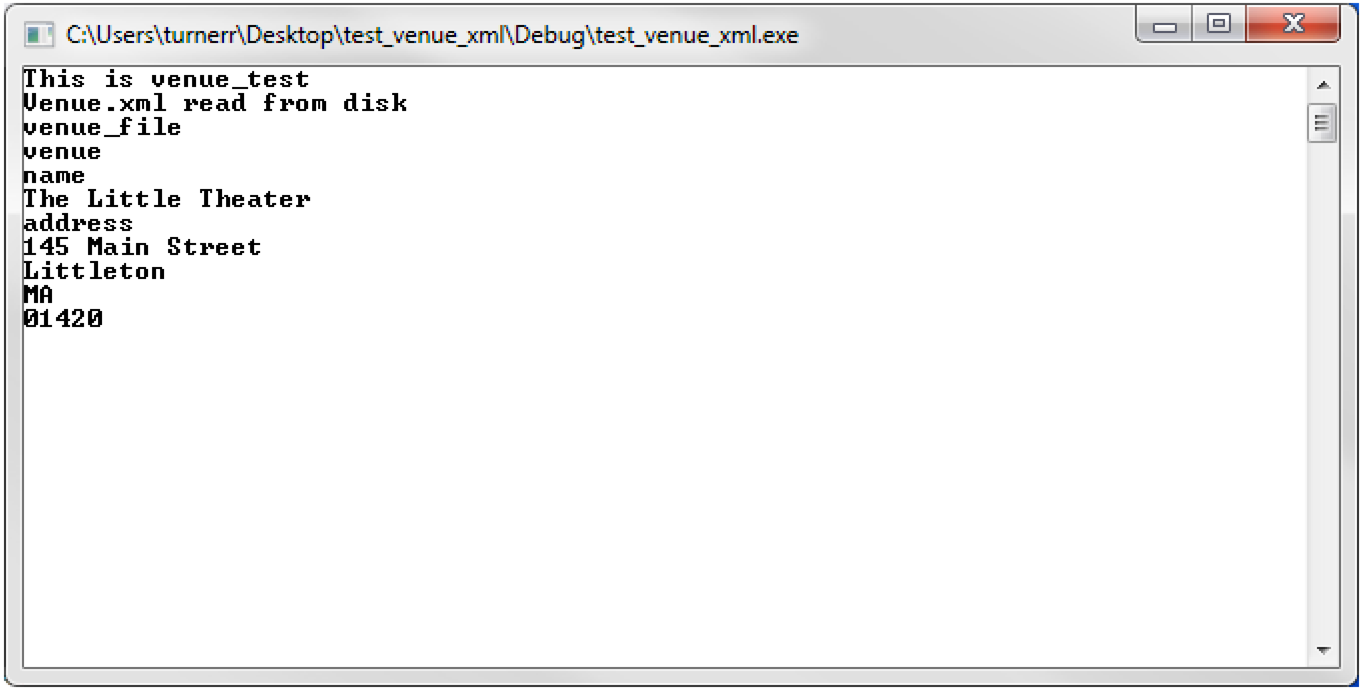
Add Call to Get_Address

```
void Venue_from_Xml::Get_Venue(TiXmlNode* venue_node)
{
    TiXmlNode* name_node = venue_node->FirstChild();
    assert(name_node != 0);
    cout << name_node->Value() << endl;

    TiXmlNode* name_text_node = name_node->FirstChild();
    assert(name_text_node != 0);
    cout << name_text_node->Value() << endl;

    TiXmlNode* address_node = name_node->NextSibling();
    assert(address_node != 0);
    cout << address_node->Value() << endl;

    Get_Address(address_node);
}
```

Add Get_Address to Venue_from_Xml.h

```
#pragma once
#include "tinyxml.h"

class Venue_from_Xml
{
public:
    static void Get_Venue(TiXmlNode* venue_node);

private:
    static void Get_Address(TiXmlNode* address_node);
};
```

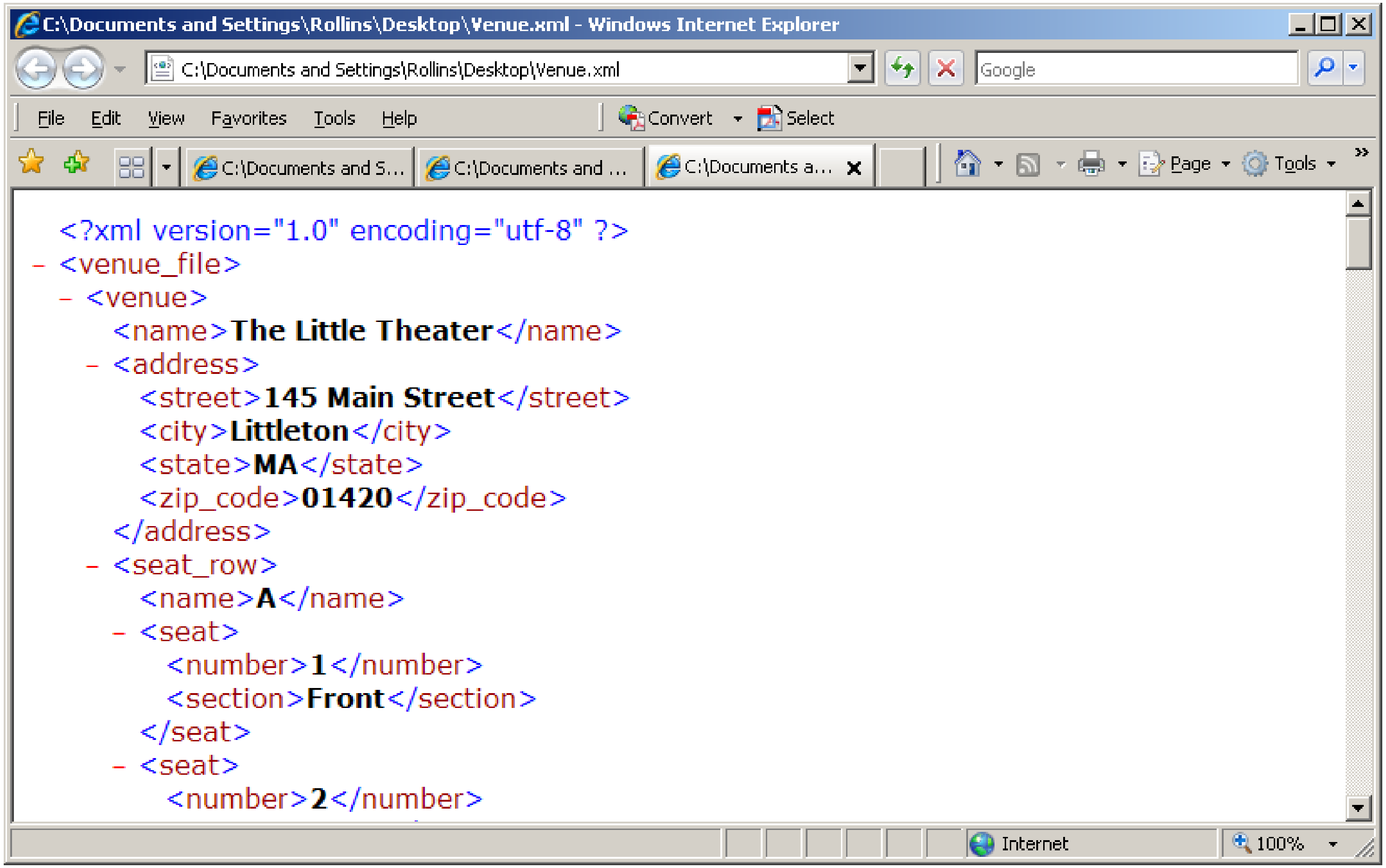
The Address in Output


C:\Users\turnerr\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
Venue.xml read from disk
venue_file
venue
name
The Little Theater
address
145 Main Street
Littleton
MA
01420
```

End of Section

Get Seat Rows



C:\Documents and Settings\Rollins\Desktop\Venue.xml - Windows Internet Explorer
C:\Documents and Settings\Rollins\Desktop\Venue.xml
Google
File Edit View Favorites Tools Help
Convert Select
C:\Documents and S...
C:\Documents and ...
C:\Documents a...
Page
Tools
Internet
100%

```
  <?xml version="1.0" encoding="utf-8" ?>
- <venue_file>
  - <venue>
      <name>The Little Theater</name>
    - <address>
        <street>145 Main Street</street>
        <city>Littleton</city>
        <state>MA</state>
        <zip_code>01420</zip_code>
      </address>
    - <seat_row>
        <name>A</name>
      - <seat>
          <number>1</number>
          <section>Front</section>
        </seat>
      - <seat>
          <number>2</number>
```

Get_Seat_Row()

- Start by getting a single seat row name.

Venue_from_Xml.cpp

```
void Venue_from_Xml::Get_Seat_Row(TiXmlNode* seat_row_node)
{
    cout << seat_row_node->Value() << endl;
    TiXmlNode* name_node = seat_row_node->FirstChild("name");
    assert(name_node != 0);
    cout << name_node->Value() << ": ";
    cout << name_node->FirstChild()->Value() << endl;
}

void Venue_from_Xml::Get_Seats(TiXmlNode* seat_row_node)
{
    Get_Seat_Row(seat_row_node);
}
```

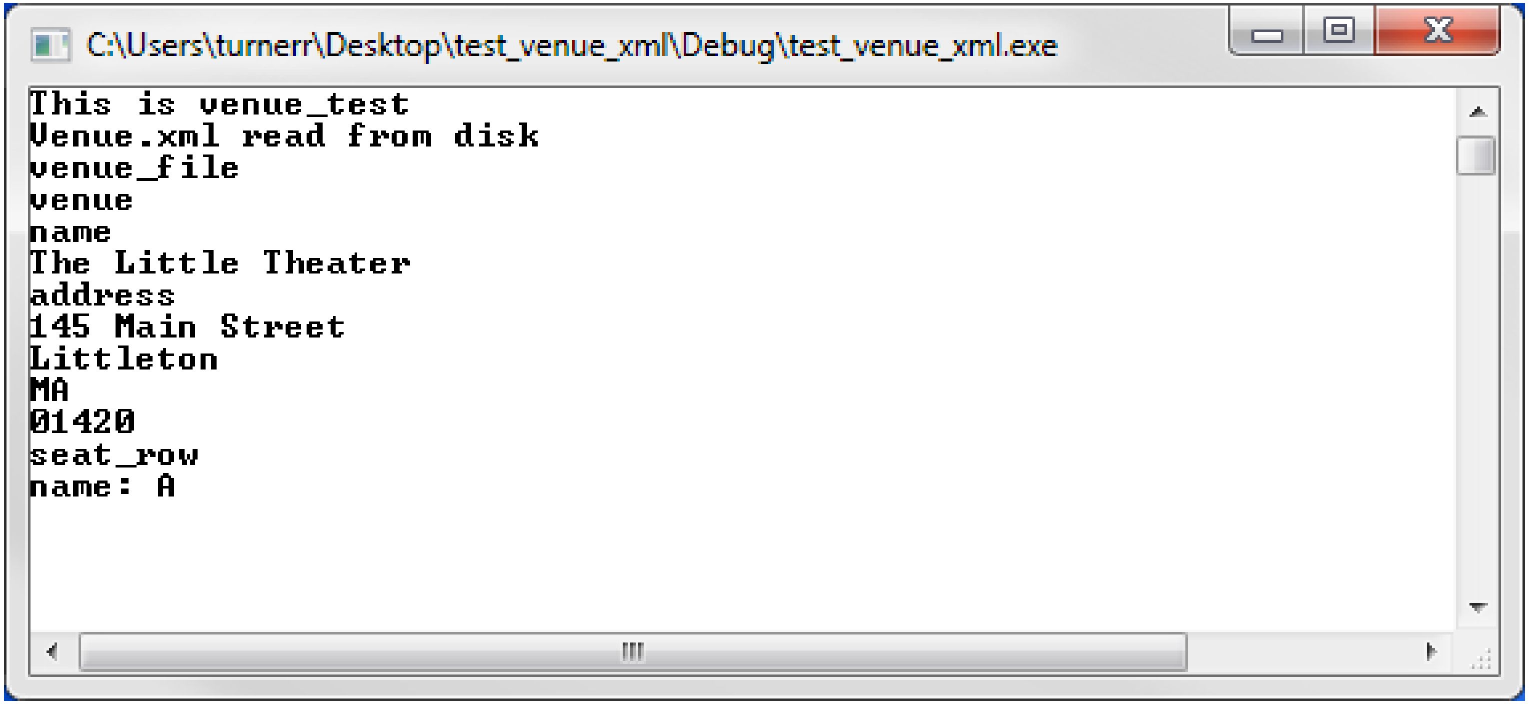
Venue_from_Xml.cpp

```
void Venue_from_Xml::Get_Venue(TiXmlNode* venue_node)
{
    ...
    Get_Address(address_node);

    TiXmlNode* seat_row_node = address_node->NextSibling();
    assert(seat_row_node != 0);
    Get_Seats(seat_row_node);
}
```

Venue_from_Xml.h

```
#pragma once
#include "tinyxml.h"

class Venue_from_Xml
{
public:
    static void Get_Venue(TiXmlNode* venue_node);

private:
    static void Get_Address(TiXmlNode* address_node);

    static void Get_Seat_Row(TiXmlNode* seat_row_node);
    static void Get_Seats(TiXmlNode* seat_row_node);
};
```

First Seat Row



C:\Users\turnerr\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
Venue.xml read from disk
venue_file
venue
name
The Little Theater
address
145 Main Street
Littleton
MA
01420
seat_row
name: A
```

Get All Seats in the Row

```
void Venue_from_Xml::Get_Seat_Row(TiXmlNode* seat_row_node)
{
    cout << seat_row_node->Value() << endl;
    TiXmlNode* name_node = seat_row_node->FirstChild("name");
    assert(name_node != 0);
    cout << name_node->Value() << ": ";
    cout << name_node->FirstChild()->Value() << endl;

    TiXmlNode* seat_node = seat_row_node->FirstChild("seat");
    while (seat_node != 0)
    {
        cout << seat_node->Value() << " ";
        Get_Seat(seat_node);
        seat_node = seat_node->NextSibling();
    }
}
```

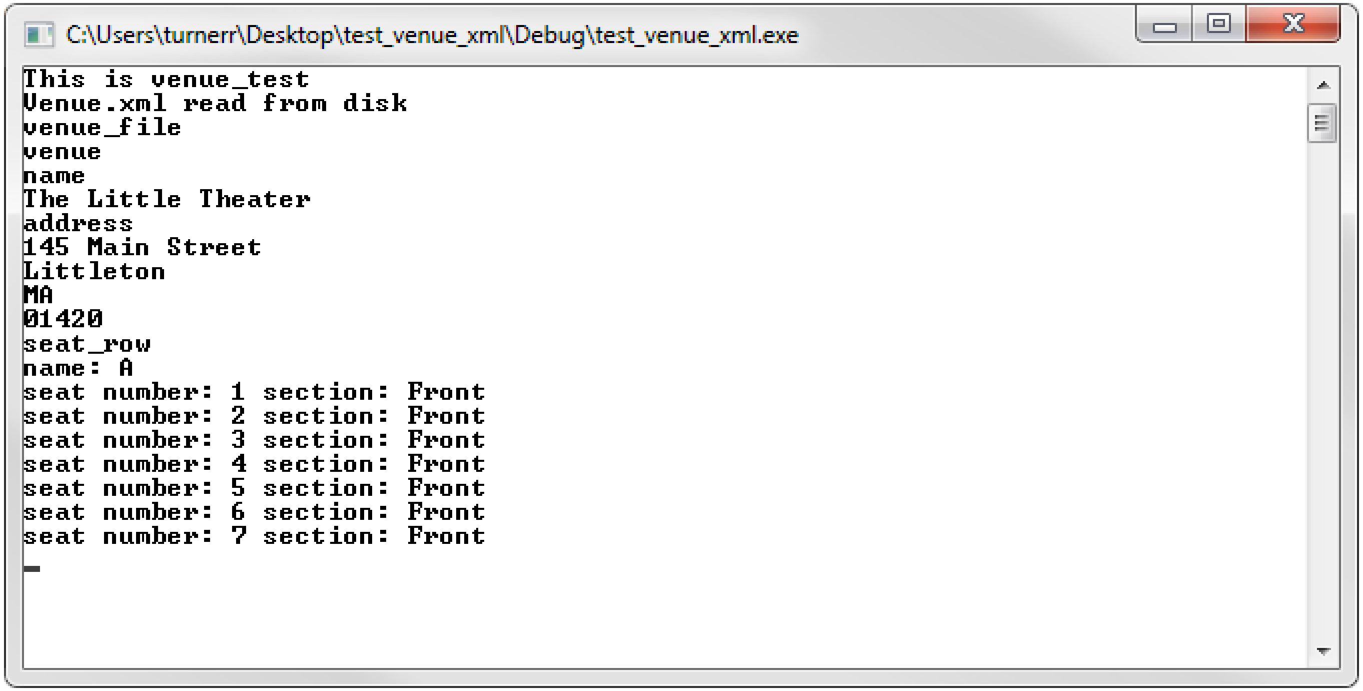
Get Seat

```
void Venue_from_Xml::Get_Seat(TiXmlNode* seat_node)
{
    TiXmlNode* number_node = seat_node->FirstChild("number");
    cout << number_node->Value() << ": ";
    cout << number_node->FirstChild()->Value() << " ";

    TiXmlNode* section_node = seat_node->FirstChild("section");
    cout << section_node->Value() << ": ";
    cout << section_node->FirstChild()->Value() << endl;
}
```

Venue_from_Xml.h

```
#pragma once
#include "tinyxml.h"

class Venue_from_Xml
{
public:
    static void Get_Venue(TiXmlNode* venue_node);

private:
    static void Get_Address(TiXmlNode* address_node);

    static void Get_Seat_Row(TiXmlNode* seat_row_node);
    static void Get_Seats(TiXmlNode* seat_row_node);
    static void Get_Seat(TiXmlNode* seat_node);
};
```

Seat Row A



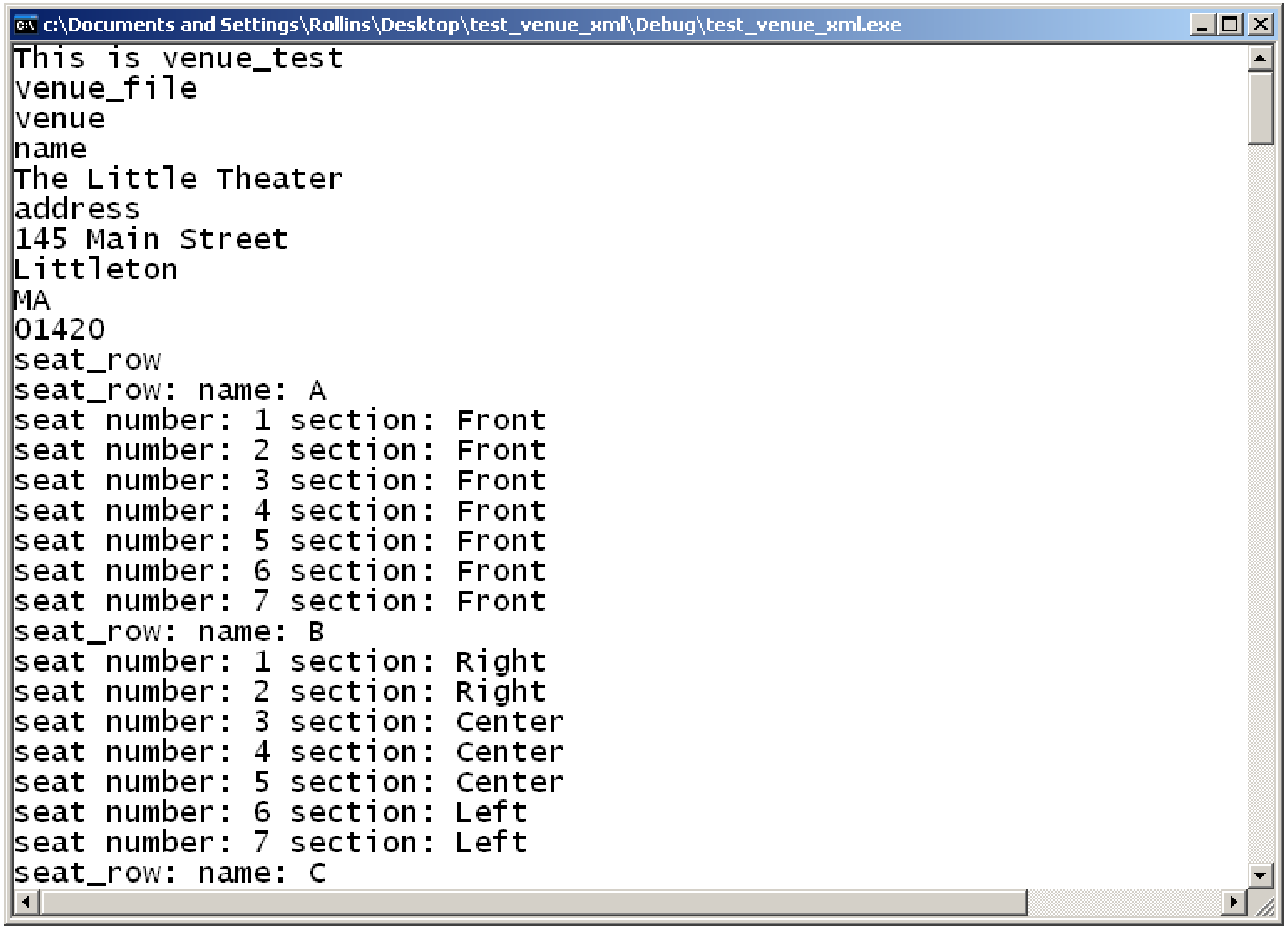
C:\Users\turnerr\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
Venue.xml read from disk
venue_file
venue
name
The Little Theater
address
145 Main Street
Littleton
MA
01420
seat_row
name: A
seat number: 1 section: Front
seat number: 2 section: Front
seat number: 3 section: Front
seat number: 4 section: Front
seat number: 5 section: Front
seat number: 6 section: Front
seat number: 7 section: Front
_
```

Get all seat rows

```
void Venue_from_Xml::Get_Seats(TiXmlNode* seat_row_node)
{
    while (seat_row_node != 0)
    {
        Get_Seat_Row(seat_row_node);
        seat_row_node = seat_row_node->NextSibling();
    }
}
```

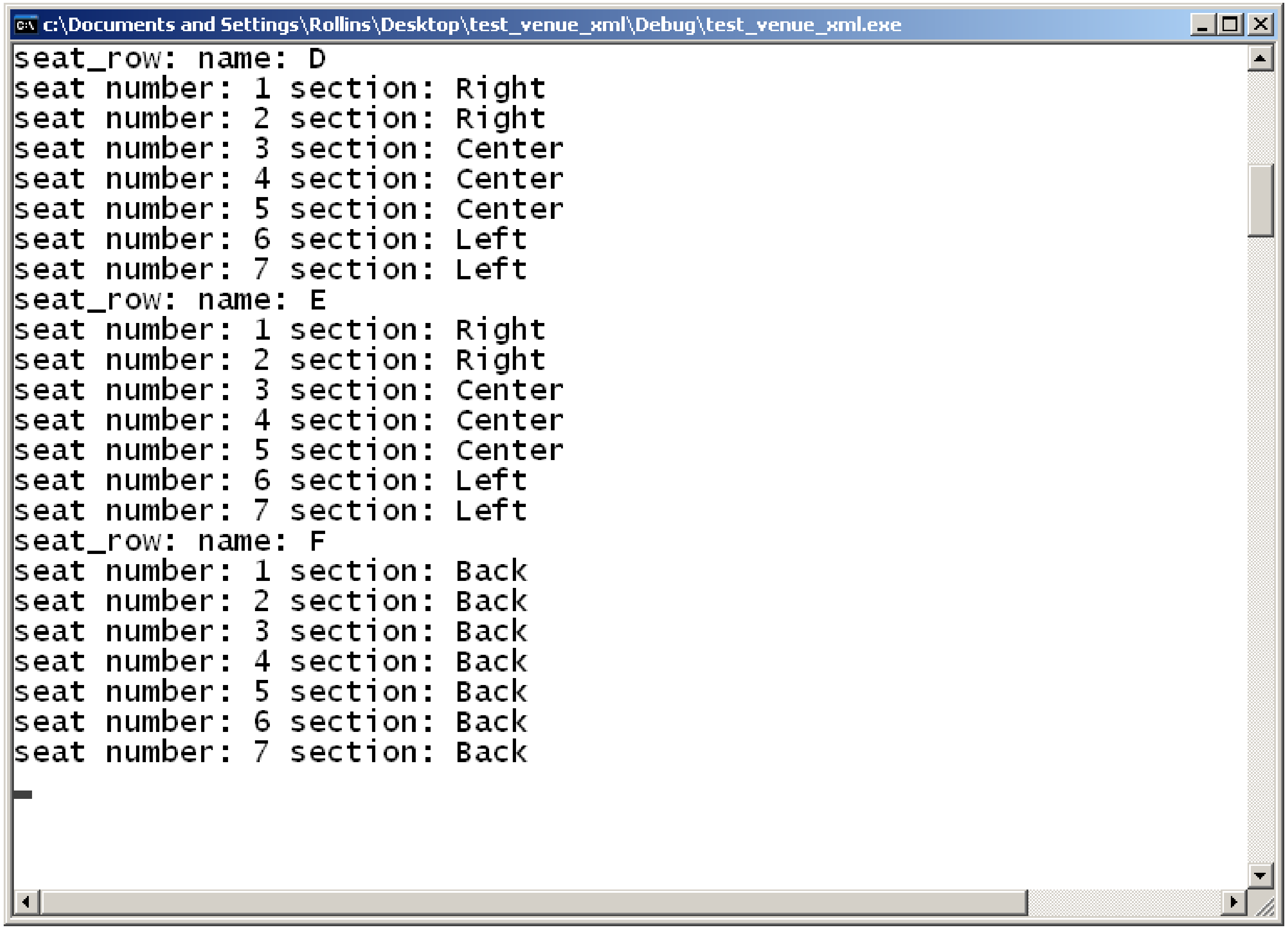
All Seat Rows in Output



c:\Documents and Settings\Rollins\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
This is venue_test
venue_file
venue
name
The Little Theater
address
145 Main Street
Littleton
MA
01420
seat_row
seat_row: name: A
seat number: 1 section: Front
seat number: 2 section: Front
seat number: 3 section: Front
seat number: 4 section: Front
seat number: 5 section: Front
seat number: 6 section: Front
seat number: 7 section: Front
seat_row: name: B
seat number: 1 section: Right
seat number: 2 section: Right
seat number: 3 section: Center
seat number: 4 section: Center
seat number: 5 section: Center
seat number: 6 section: Left
seat number: 7 section: Left
seat_row: name: C
```

All Seat Rows in Output


c:\Documents and Settings\Rollins\Desktop\test_venue_xml\Debug\test_venue_xml.exe

```
seat_row: name: D
seat number: 1 section: Right
seat number: 2 section: Right
seat number: 3 section: Center
seat number: 4 section: Center
seat number: 5 section: Center
seat number: 6 section: Left
seat number: 7 section: Left
seat_row: name: E
seat number: 1 section: Right
seat number: 2 section: Right
seat number: 3 section: Center
seat number: 4 section: Center
seat number: 5 section: Center
seat number: 6 section: Left
seat number: 7 section: Left
seat_row: name: F
seat number: 1 section: Back
seat number: 2 section: Back
seat number: 3 section: Back
seat number: 4 section: Back
seat number: 5 section: Back
seat number: 6 section: Back
seat number: 7 section: Back
```