Reading and writing text files in Open C++

Overview

This code snippet shows how to use Open C++ file I/O classes for text file read and write operations. In this case, a text file is considered to be a file that stores words and sentences in readable plain text. The stream class fstream can be used for both read and write file I/O operations. Ifstream is only capable of file read operations and ofstream is only capable of file write operations. A header file <fstream> must be included in the application before these classes can be used. Reading and writing text files is possible with the extraction (<<) and the insertion (>>) operators and methods like `put()`, `get()`, and `getline()`.

Note: In order to use this code, you need to install the Open C/C++ plug-in.

This snippet can be self-signed.

MMP file

The following libraries are required:

```
STATICLIBRARY libcrt0.lib

LIBRARY libstdcpp.lib
LIBRARY libc.lib
LIBRARY euser.lib
```

Source file

```
#include <fstream>  //fstream, ifstream, ofstream
#include <string>   //string, getline()
#include <iostream> //cout

using namespace std;

void simple_write_example()
{
    ofstream file;

    char line [] = "Text To File";
    string str  = "Hello";

    //default open mode is a text mode,
    //a new file is created if a file does not exist
    //if a path is not given the applications's \private folder is used
    file.open("simple.txt");  //ios::out | ios::trunc

    file << line << endl;

    file.put('A') ;

    file <<  1;

    file << str;

    //close the file
    file.close();
}

void simple_read_example()
{
    ifstream file;

    char line [20];
    char chr = ' ';
    int num = 0;
    string str = "";

    file.open("simple.txt"); //ios::in

    //getline has an optional third argument (character)
    //that will end getline's input, the default value is '\n'
    file.getline(line, 20);

    file.get(chr); // or file >> chr;

    file >> num;

    file >> str;

    cout << "line:"<< line << endl;
    cout << "chr:" << chr << endl;
    cout << "num:" << num << endl;
    cout << "str:"<< str << endl;

    file.close();
}


void simple_read_and_write_example()
{
```


(C) Copyright Nokia 2013. All rights reserved.

```
    fstream file;
    string line;

    //fstream provides attributes which define
    //how a file should be opened:
    //app    = 0x01 - append to the end of a file
    //ate    = 0x02 - place the file marker at the end of the file
    //binary = 0x04 - open as a binary file
    //in     = 0x08 - open file for reading
    //out    = 0x10 - open file for writing
    //trunc  = 0x20 - truncate an existing file and overwrite (default)
    file.open("simple.txt", ios::in | ios::out);

    //write to text file
    file << "This is a line 1" << endl;
    file << "This is a line 2" << endl;

    //fstream works with read and write file pointers,
    //by moving these pointers it is possible to access any part of
    //the file at random - seekg() moves the read pointer and
    //seekp() moves the write pointer

    //seek attributes:
    //ios::beg - beginning of the file
    //ios::end - end of the file
    //ios::cur - current location of the file
    file.seekg(0,ios::beg);

    //read text from file
    if (file.is_open())
        {
        while (!file.eof())
            {
            getline(file, line);
            cout << line << endl;
            }
        }

    file.close();
}

int main()
{
    //using class ofstream
    simple_write_example();

    //using class ifstream
    simple_read_example();

    //using class fstream
    simple_read_and_write_example();

    return 0;
}
```

Postconditions

The Open C++ I/O classes fstream, ifstream, and ofstream are used to read and write text from/to the created file simple.txt. Content of the file is displayed as standard output.

See also

- Reading and writing binary files in Open C++
- Checking the file I/O status in Open C++

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.