#pragma once

#include <twml/defines.h>

#include <cstdlib>

#include <cstdio>

#include <unistd.h>

#include <cinttypes>

#include <cstdint>

#ifndef PATH\_MAX

#define PATH\_MAX (8096)

#endif

#ifdef \_\_cplusplus

extern "C" {

#endif

struct block\_format\_writer\_\_;

typedef block\_format\_writer\_\_ \* block\_format\_writer;

#ifdef \_\_cplusplus

}

#endif

#ifdef \_\_cplusplus

namespace twml {

class BlockFormatWriter {

private:

const char \*file\_name\_;

FILE \*outputfile\_;

char temp\_file\_name\_[PATH\_MAX];

int record\_index\_;

int records\_per\_block\_;

int pack\_tag\_and\_wiretype(FILE \*file, uint32\_t tag, uint32\_t wiretype);

int pack\_varint\_i32(FILE \*file, int value);

int pack\_string(FILE \*file, const char \*in, size\_t in\_len);

int write\_int(FILE \*file, int value);

public:

BlockFormatWriter(const char \*file\_name, int record\_per\_block);

~BlockFormatWriter();

int write(const char \*class\_name, const char \*record, int record\_len) ;

int flush();

block\_format\_writer getHandle();

};

BlockFormatWriter \*getBlockFormatWriter(block\_format\_writer w);

} //twml namespace

#endif

#ifdef \_\_cplusplus

extern "C" {

#endif

twml\_err block\_format\_writer\_create(block\_format\_writer \*w, const char \*file\_name, int records\_per\_block);

twml\_err block\_format\_write(block\_format\_writer w, const char \*class\_name, const char \*record, int record\_len);

twml\_err block\_format\_flush(block\_format\_writer w);

twml\_err block\_format\_writer\_delete(const block\_format\_writer w);

#ifdef \_\_cplusplus

}

#endif