#pragma once

#include <twml/Error.h>

namespace twml {

namespace io {

class IOError : public twml::Error {

public:

enum Status {

OUT\_OF\_RANGE = 1,

WRONG\_MAGIC = 2,

WRONG\_HEADER = 3,

ERROR\_HEADER\_CHECKSUM = 4,

INVALID\_METHOD = 5,

USING\_RESERVED = 6,

ERROR\_HEADER\_EXTRA\_FIELD\_CHECKSUM = 7,

CANT\_FIT\_OUTPUT = 8,

SPLIT\_FILE = 9,

BLOCK\_SIZE\_TOO\_LARGE = 10,

SOURCE\_LARGER\_THAN\_DESTINATION = 11,

DESTINATION\_LARGER\_THAN\_CAPACITY = 12,

HEADER\_FLAG\_MISMATCH = 13,

NOT\_ENOUGH\_INPUT = 14,

ERROR\_SOURCE\_BLOCK\_CHECKSUM = 15,

COMPRESSED\_DATA\_VIOLATION = 16,

ERROR\_DESTINATION\_BLOCK\_CHECKSUM = 17,

EMPTY\_RECORD = 18,

MALFORMED\_MEMORY\_RECORD = 19,

UNSUPPORTED\_OUTPUT\_TYPE = 20,

OTHER\_ERROR

};

IOError(Status status);

Status status() const {

return m\_status;

}

private:

Status m\_status;

};

}

}