// Copyright 2013 The Flutter Authors. All rights reserved.

// Use of this source code is governed by a BSD-style license that can be

// found in the LICENSE file.

#ifndef FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_BYTE\_STREAMS\_H\_

#define FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_BYTE\_STREAMS\_H\_

// Interfaces for interacting with a stream of bytes, for use in codecs.

namespace flutter {

// An interface for a class that reads from a byte stream.

class ByteStreamReader {

public:

explicit ByteStreamReader() = default;

virtual ~ByteStreamReader() = default;

// Reads and returns the next byte from the stream.

virtual uint8\_t ReadByte() = 0;

// Reads the next |length| bytes from the stream into |buffer|. The caller

// is responsible for ensuring that |buffer| is large enough.

virtual void ReadBytes(uint8\_t\* buffer, size\_t length) = 0;

// Advances the read cursor to the next multiple of |alignment| relative to

// the start of the stream, unless it is already aligned.

virtual void ReadAlignment(uint8\_t alignment) = 0;

// Reads and returns the next 32-bit integer from the stream.

int32\_t ReadInt32() {

int32\_t value = 0;

ReadBytes(reinterpret\_cast<uint8\_t\*>(&value), 4);

return value;

}

// Reads and returns the next 64-bit integer from the stream.

int64\_t ReadInt64() {

int64\_t value = 0;

ReadBytes(reinterpret\_cast<uint8\_t\*>(&value), 8);

return value;

}

// Reads and returns the next 64-bit floating point number from the stream.

double ReadDouble() {

double value = 0;

ReadBytes(reinterpret\_cast<uint8\_t\*>(&value), 8);

return value;

}

};

// An interface for a class that writes to a byte stream.

class ByteStreamWriter {

public:

explicit ByteStreamWriter() = default;

virtual ~ByteStreamWriter() = default;

// Writes |byte| to the stream.

virtual void WriteByte(uint8\_t byte) = 0;

// Writes the next |length| bytes from |bytes| to the stream

virtual void WriteBytes(const uint8\_t\* bytes, size\_t length) = 0;

// Writes 0s until the next multiple of |alignment| relative to the start

// of the stream, unless the write positition is already aligned.

virtual void WriteAlignment(uint8\_t alignment) = 0;

// Writes the given 32-bit int to the stream.

void WriteInt32(int32\_t value) {

WriteBytes(reinterpret\_cast<const uint8\_t\*>(&value), 4);

}

// Writes the given 64-bit int to the stream.

void WriteInt64(int64\_t value) {

WriteBytes(reinterpret\_cast<const uint8\_t\*>(&value), 8);

}

// Writes the given 36-bit double to the stream.

void WriteDouble(double value) {

WriteBytes(reinterpret\_cast<const uint8\_t\*>(&value), 8);

}

};

} // namespace flutter

#endif // FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_BYTE\_STREAMS\_H\_