// 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\_STANDARD\_CODEC\_SERIALIZER\_H\_

#define FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_STANDARD\_CODEC\_SERIALIZER\_H\_

#include "byte\_streams.h"

#include "encodable\_value.h"

namespace flutter {

// Encapsulates the logic for encoding/decoding EncodableValues to/from the

// standard codec binary representation.

//

// This can be subclassed to extend the standard codec with support for new

// types.

class StandardCodecSerializer {

public:

virtual ~StandardCodecSerializer();

// Returns the shared serializer instance.

static const StandardCodecSerializer& GetInstance();

// Prevent copying.

StandardCodecSerializer(StandardCodecSerializer const&) = delete;

StandardCodecSerializer& operator=(StandardCodecSerializer const&) = delete;

// Reads and returns the next value from |stream|.

EncodableValue ReadValue(ByteStreamReader\* stream) const;

// Writes the encoding of |value| to |stream|, including the initial type

// discrimination byte.

//

// Can be overridden by a subclass to extend the codec.

virtual void WriteValue(const EncodableValue& value,

ByteStreamWriter\* stream) const;

protected:

// Codecs require long-lived serializers, so clients should always use

// GetInstance().

StandardCodecSerializer();

// Reads and returns the next value from |stream|, whose discrimination byte

// was |type|.

//

// The discrimination byte will already have been read from the stream when

// this is called.

//

// Can be overridden by a subclass to extend the codec.

virtual EncodableValue ReadValueOfType(uint8\_t type,

ByteStreamReader\* stream) const;

// Reads the variable-length size from the current position in |stream|.

size\_t ReadSize(ByteStreamReader\* stream) const;

// Writes the variable-length size encoding to |stream|.

void WriteSize(size\_t size, ByteStreamWriter\* stream) const;

private:

// Reads a fixed-type list whose values are of type T from the current

// position in |stream|, and returns it as the corresponding EncodableValue.

// |T| must correspond to one of the supported list value types of

// EncodableValue.

template <typename T>

EncodableValue ReadVector(ByteStreamReader\* stream) const;

// Writes |vector| to |stream| as a fixed-type list. |T| must correspond to

// one of the supported list value types of EncodableValue.

template <typename T>

void WriteVector(const std::vector<T> vector, ByteStreamWriter\* stream) const;

};

} // namespace flutter

#endif // FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_STANDARD\_CODEC\_SERIALIZER\_H\_