// 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\_BINARY\_MESSENGER\_IMPL\_H\_

#define FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_BINARY\_MESSENGER\_IMPL\_H\_

#include <flutter\_messenger.h>

#include <map>

#include <string>

#include "include/flutter/binary\_messenger.h"

namespace flutter {

// Wrapper around a FlutterDesktopMessengerRef that implements the

// BinaryMessenger API.

class BinaryMessengerImpl : public BinaryMessenger {

public:

explicit BinaryMessengerImpl(FlutterDesktopMessengerRef core\_messenger);

virtual ~BinaryMessengerImpl();

// Prevent copying.

BinaryMessengerImpl(BinaryMessengerImpl const&) = delete;

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

// |flutter::BinaryMessenger|

void Send(const std::string& channel,

const uint8\_t\* message,

size\_t message\_size,

BinaryReply reply) const override;

// |flutter::BinaryMessenger|

void SetMessageHandler(const std::string& channel,

BinaryMessageHandler handler) override;

private:

// Handle for interacting with the C API.

FlutterDesktopMessengerRef messenger\_;

// A map from channel names to the BinaryMessageHandler that should be called

// for incoming messages on that channel.

std::map<std::string, BinaryMessageHandler> handlers\_;

};

} // namespace flutter

#endif // FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_BINARY\_MESSENGER\_IMPL\_H\_