// 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\_WINDOWS\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_FLUTTER\_VIEW\_H\_

#define FLUTTER\_SHELL\_PLATFORM\_WINDOWS\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_FLUTTER\_VIEW\_H\_

#include <flutter\_windows.h>

namespace flutter {

// The unique identifier for a view.

typedef int64\_t FlutterViewId;

// A view displaying Flutter content.

class FlutterView {

public:

explicit FlutterView(FlutterDesktopViewRef view) : view\_(view) {}

// Destroys this reference to the view. The underlying view is not destroyed.

virtual ~FlutterView() = default;

// Prevent copying.

FlutterView(FlutterView const&) = delete;

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

// Returns the backing HWND for the view.

HWND GetNativeWindow() { return FlutterDesktopViewGetHWND(view\_); }

// Returns the DXGI adapter used for rendering or nullptr in case of error.

IDXGIAdapter\* GetGraphicsAdapter() {

return FlutterDesktopViewGetGraphicsAdapter(view\_);

}

private:

// Handle for interacting with the C API's view.

FlutterDesktopViewRef view\_ = nullptr;

};

} // namespace flutter

#endif // FLUTTER\_SHELL\_PLATFORM\_WINDOWS\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_FLUTTER\_VIEW\_H\_