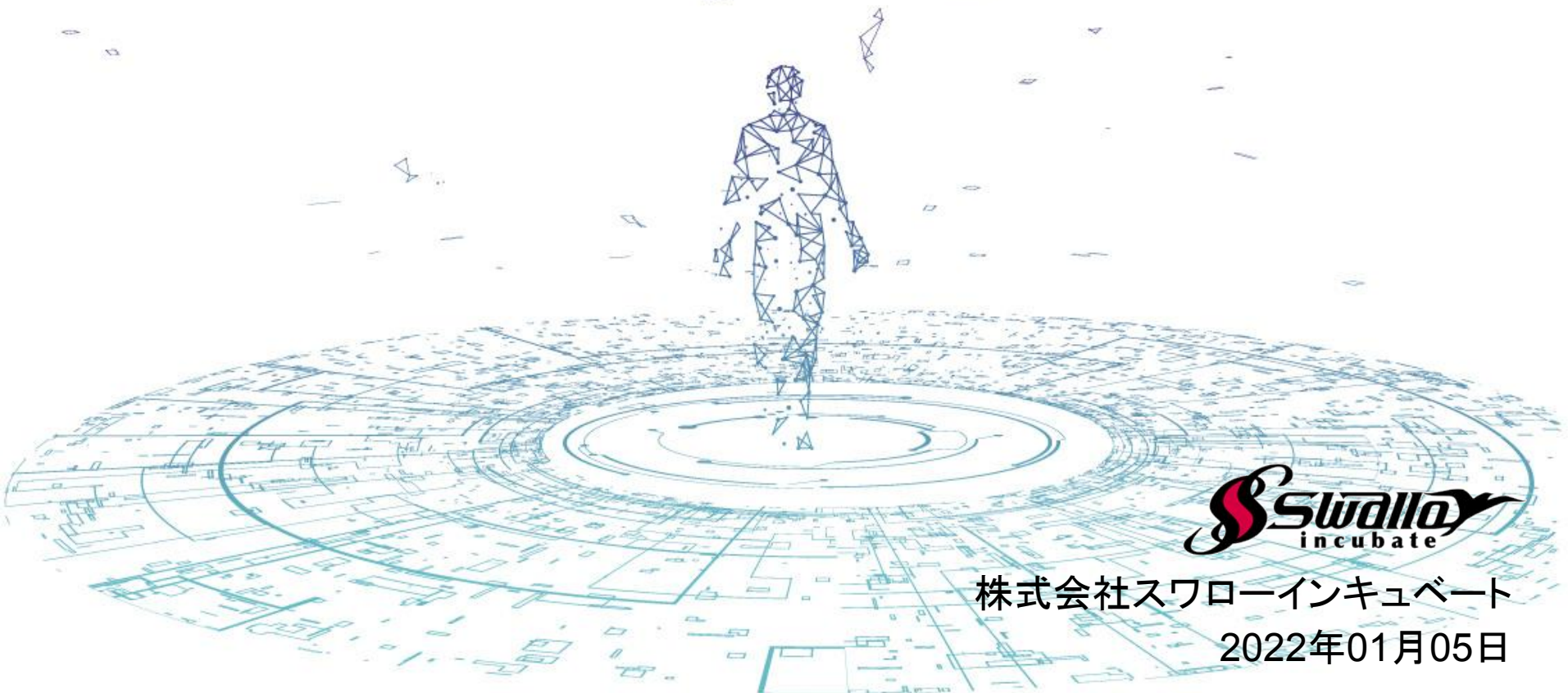


視線検出技術 .NETアプリ

SDK組み込みマニュアル



株式会社スワローインキュベート

2022年01月05日

■はじめに

- ・視線検出技術.NETアプリは、株式会社スワローインキュベートが提供しています。
- ・本書は、予告なしに変更されることがあります。
- ・本書を無断で、複製、転用、公衆送信、貸与等を行わないようお願いします。
- ・アプリをご利用いただくには、あらかじめ当社利用規約に同意いただく必要があります。
詳しくは営業担当までお問い合わせください。

■ご注意事項

- ・SDKをご購入いただいていない場合は、本アプリのソースコードはすべて当社に帰属します。
- ・当社は本プログラムに関する保証は一切行っておりません。お客様の責任にてご使用ください。
本プログラム使用による問題や損害が生じた場合にも当社は何ら責任を負いませんので、あらかじめ
ご了承ください。

お問い合わせ

株式会社スワローインキュベート

視線検出技術 テクニカルサポート窓口

TEL: 029-886-9912 MAIL: support@swallow-incubate.com

同梱内容

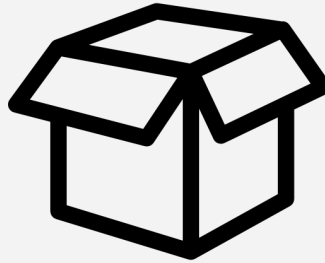
■同梱内容

以下のファイルが提供時の主要なファイルとなります。



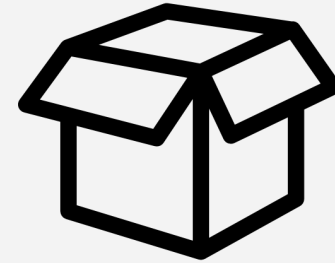
EyeTrackDotNETSample.sln

視線検出.NETアプリ



動的リンクライブラリ
EyeTrack.dll / lib / hpp

視線検出ライブラリ



動的リンクライブラリ
opencv_world453.dll ほか

OpenCVライブラリ



期限付きトークン
(試用版)

アクティベーションツール



USB dongle
(購入版)



ご利用マニュアル
(本書)

ライブラリ仕様

■ご利用環境

現在のバージョンでは、以下のご利用環境に対応しています。

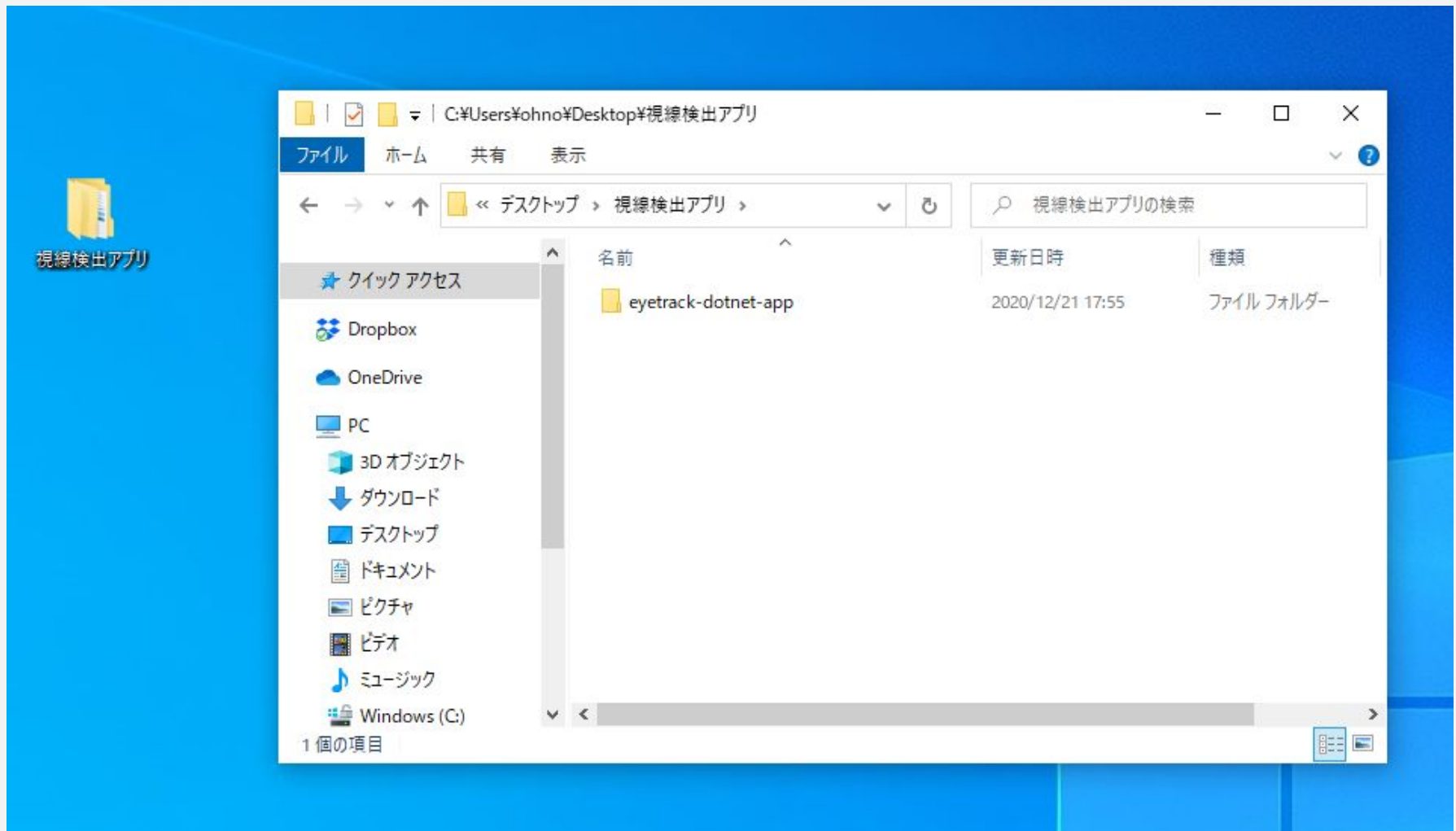
項目	内容
プログラム言語	Visual C++言語 / C++11
開発環境	Visual Studio 16 2019
ビルド環境	MicroSoft Visual C++コンパイラ
アーキテクチャ	Release x64 / Debug x64
占有メモリ	実行中 約250MB
依存ライブラリ	EyeTrack ライブラリ OpenCV 4.5.3 (一部OpenCV_Contribモジュール入り)
接続カメラ	USBカメラ (UVCタイプ) / ビルドインカメラ

※その他の環境でのご利用を希望される場合は、お問い合わせください。

SDK組み込み手順

■ご利用手順①

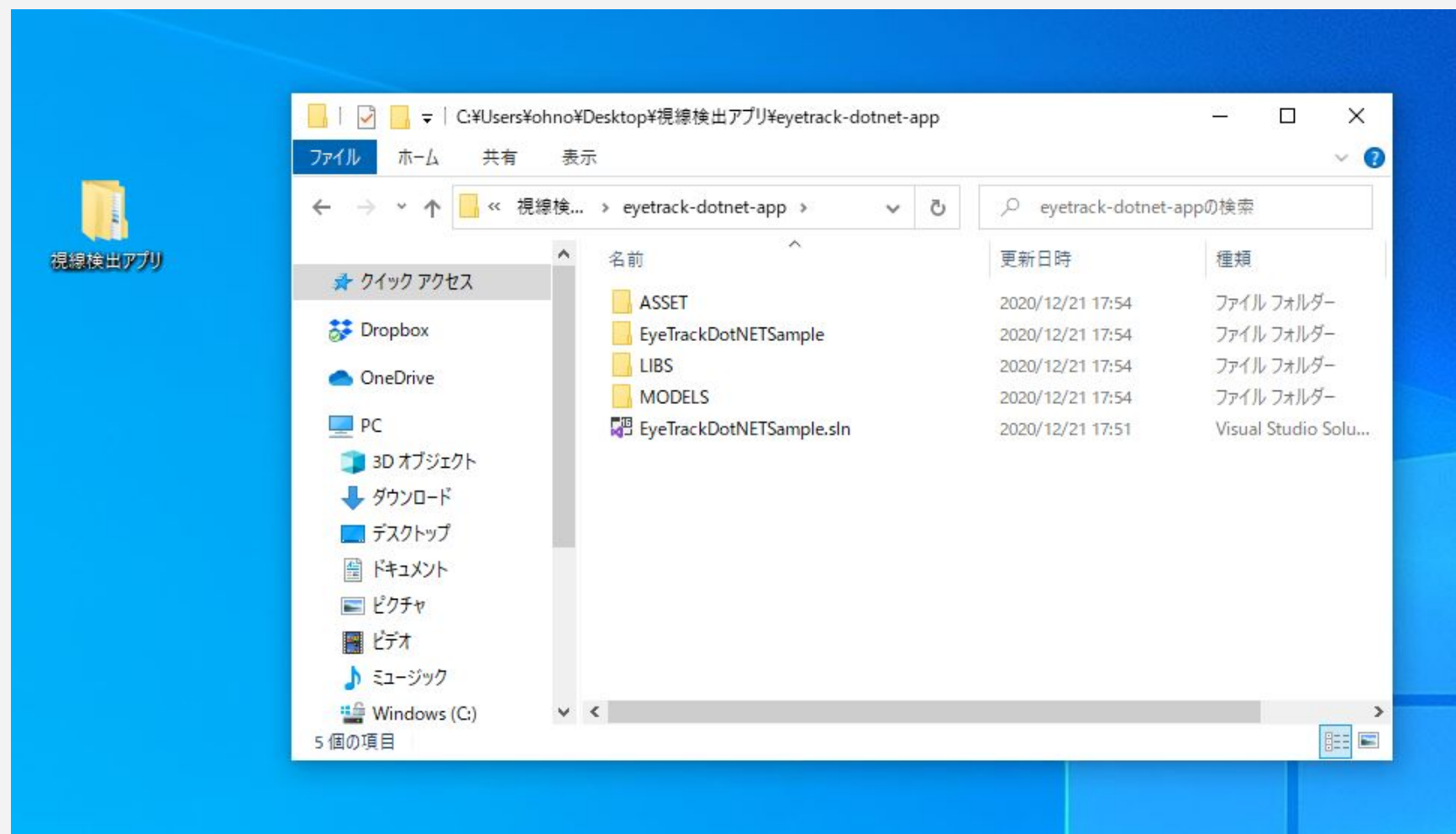
ご提供した.NETアプリフォルダを任意の場所に設置してください。
本マニュアルでは、デスクトップに置いたものとして説明をします。



■ご利用手順②

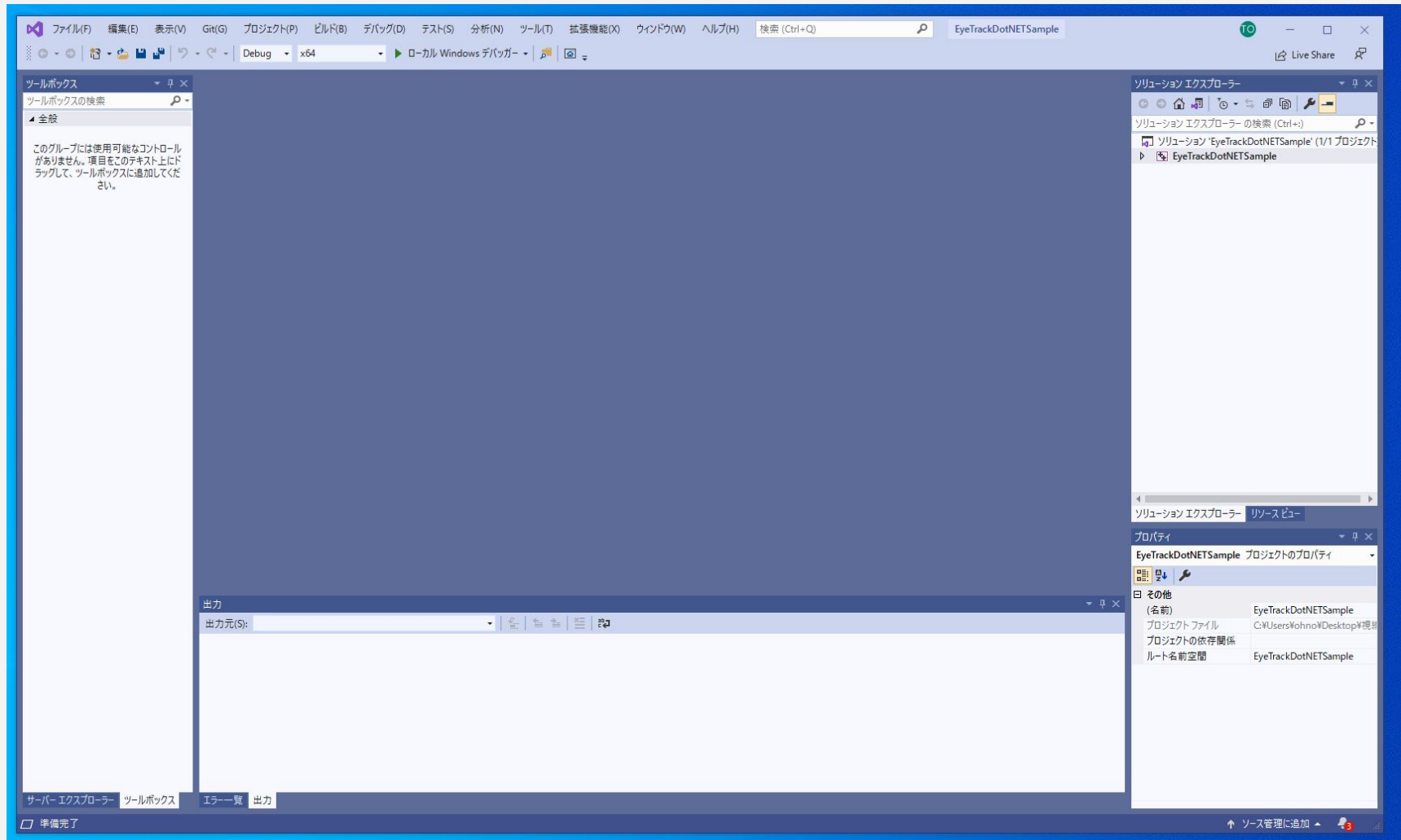
視線検出アプリ.NETアプリのフォルダを開くと、Visual Studio向けソリューションファイル(.sln)とソースコード、モデルファイル、SDKのライブラリなどが同梱されています。

お使いのマシンでVisual Studioが使える状態になっていることを確認の上、slnファイルを開いてください。



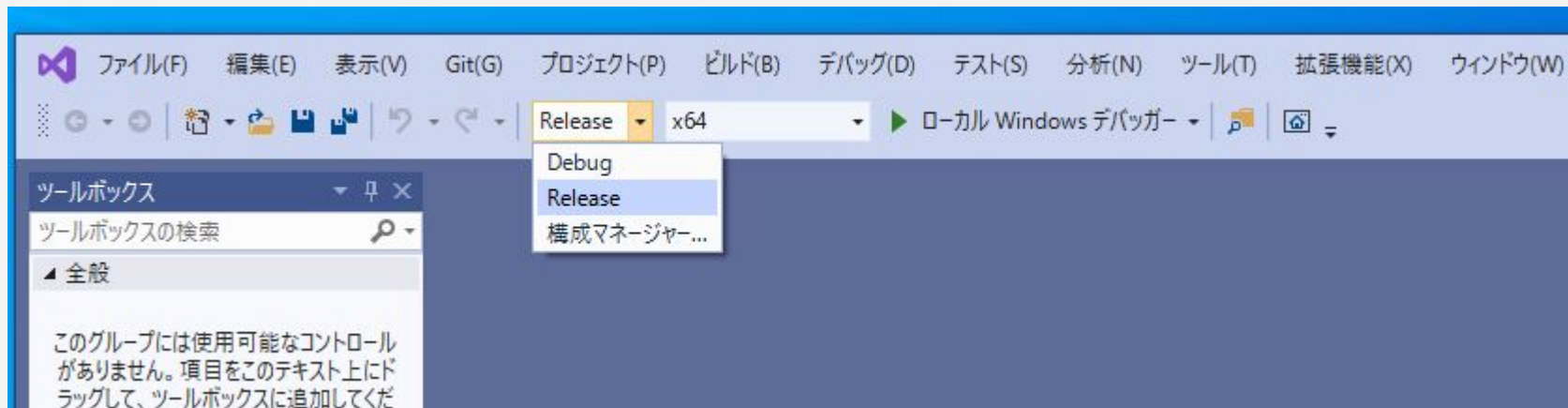
■ご利用手順③

視線検出アプリ.NETアプリをVisual Studioで開くと、
以下のようなウィンドウが立ち上がります。



■ご利用手順④

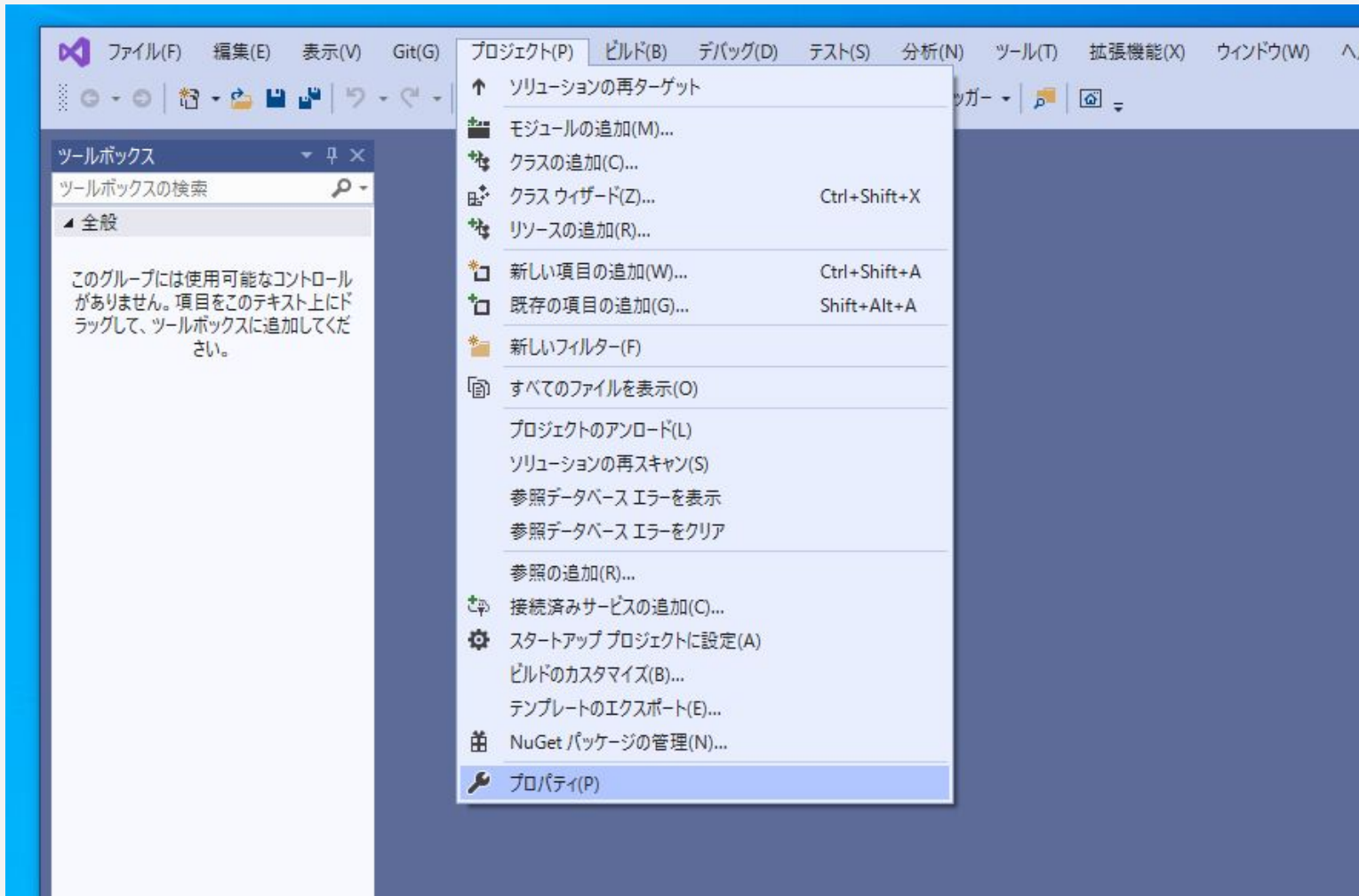
今回は、x64アーキテクチャのReleaseビルドを用いるものとします。
Visual Studioのヘッダー部分で、Release / x64に変更しておいてください。



■ご利用手順⑤

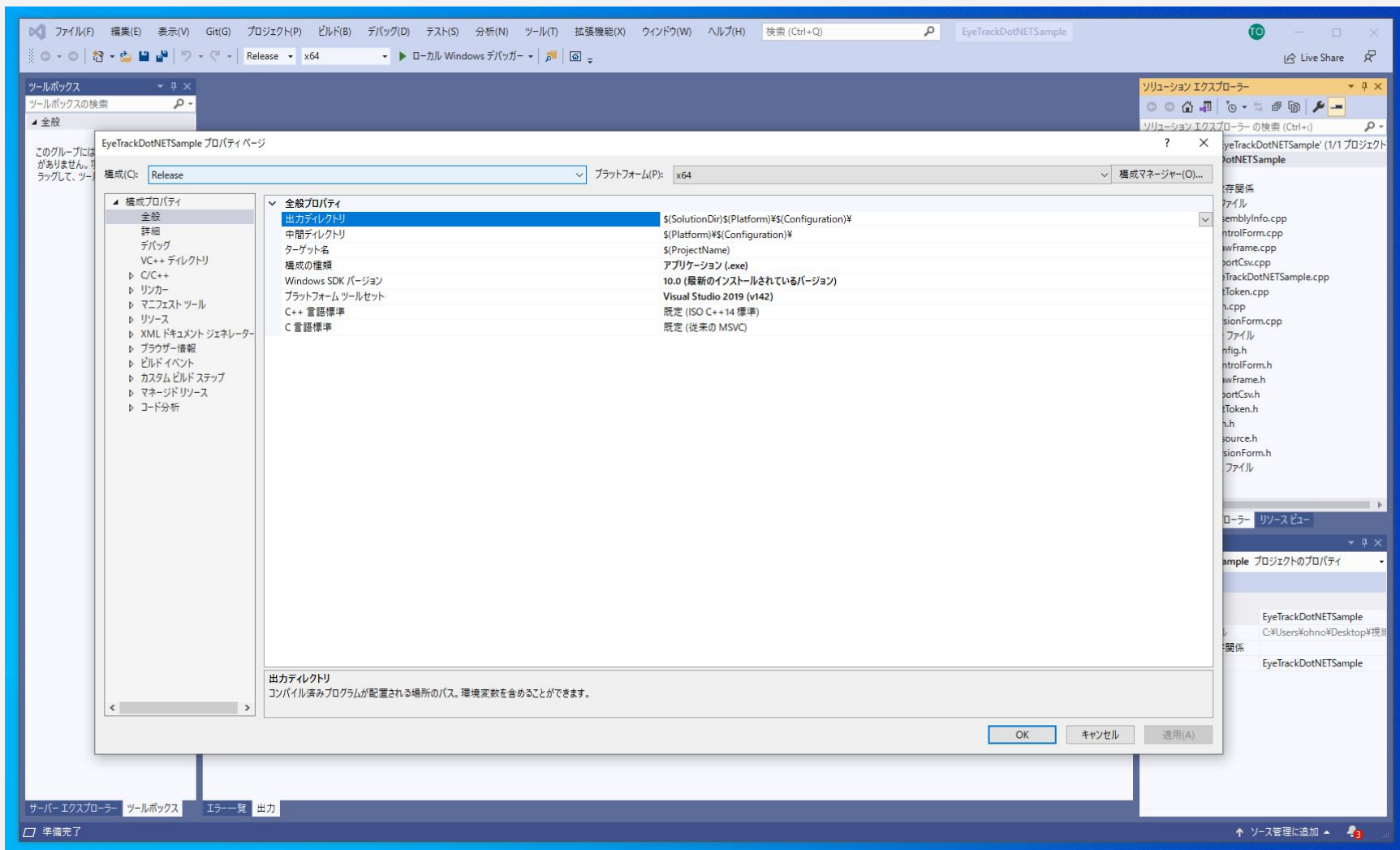
SDKのライブラリを読み込めるように、プロパティを設定します。

Visual Studioヘッダーから、「プロジェクト(P)」→「プロパティ(P)」をクリックして、プロパティページを開いてください。



■ご利用手順⑥

以下のプロパティページが立ち上がったことを確認してください。



■ご利用手順⑦

プロパティページの、「構成プロパティ」→「C/C++」→「全般」を選択し、
「追加のインクルードディレクトリ」欄より、「<編集..>」をクリックします。

EyeTrackDotNETSample プロパティ ページ

構成(C): Release ▼ プラットフォーム(P): x64

▲ 構成プロパティ	追加のインクルードディレクトリ	%(AdditionalIncludeDirectories)
全般	追加の #using ディレクトリ	<編集...>
詳細	追加の BMI ディレクトリ	<親またはプロジェクトの既定値から継承>
デバッグ	追加のモジュールの依存関係	
VC++ ディレクトリ	追加のヘッダー ユニットの依存関係	
▲ C/C++	デバッグ情報の形式	プログラム データベース (/ZI)
全般	[マイ コードのみ] のデバッグをサポートする	いいえ
最適化	共通言語ランタイム サポート	共通言語ランタイム サポート (/clr)
プリプロセッサ	Windows ランタイム拡張機能の使用	
コード生成	著作権情報の非表示	はい (/nologo)
言語	警告レベル	レベル 3 (/W3)
プリコンパイル済みヘッダー	警告をエラーとして扱う	いいえ (/WX-)
出力ファイル		

■ご利用手順⑧

ご提供した.NETアプリのLIBSフォルダ下の場所をインクルード先として以下を参考に設定をしてください。設定が終わったら「OK」ボタンで閉じます。
※本設定がうまくいっていない場合、ビルドエラーになります。

追加のインクルードディレクトリ

c:\Users\ohno\OneDrive\デスクトップ\dotnetapp\eyetrack-dotnet-app-camerainput\LIBS\OpenCV453\Release
c:\Users\ohno\OneDrive\デスクトップ\dotnetapp\eyetrack-dotnet-app-camerainput\LIBS\EyeTrack\Release

評価された値:

c:\Users\ohno\OneDrive\デスクトップ\dotnetapp\eyetrack-dotnet-app-camerainput\LIBS\OpenCV453\Release
c:\Users\ohno\OneDrive\デスクトップ\dotnetapp\eyetrack-dotnet-app-camerainput\LIBS\EyeTrack\Release
%(AdditionalIncludeDirectories)

継承の値:

☒ 親またはプロジェクトの既定値から継承(I)

マクロ(M) >>

OK キャンセル

■ご利用手順⑨

続いてプロパティページの「構成プロパティ」→「リンカー」→「すべてのオプション」を選択し、追加のライブラリディレクトリ欄から「<編集...>」を選択します。

EyeTrackDotNETSample プロパティ ページ

構成(C): Release プラットフォーム(P): x64

構成プロパティ

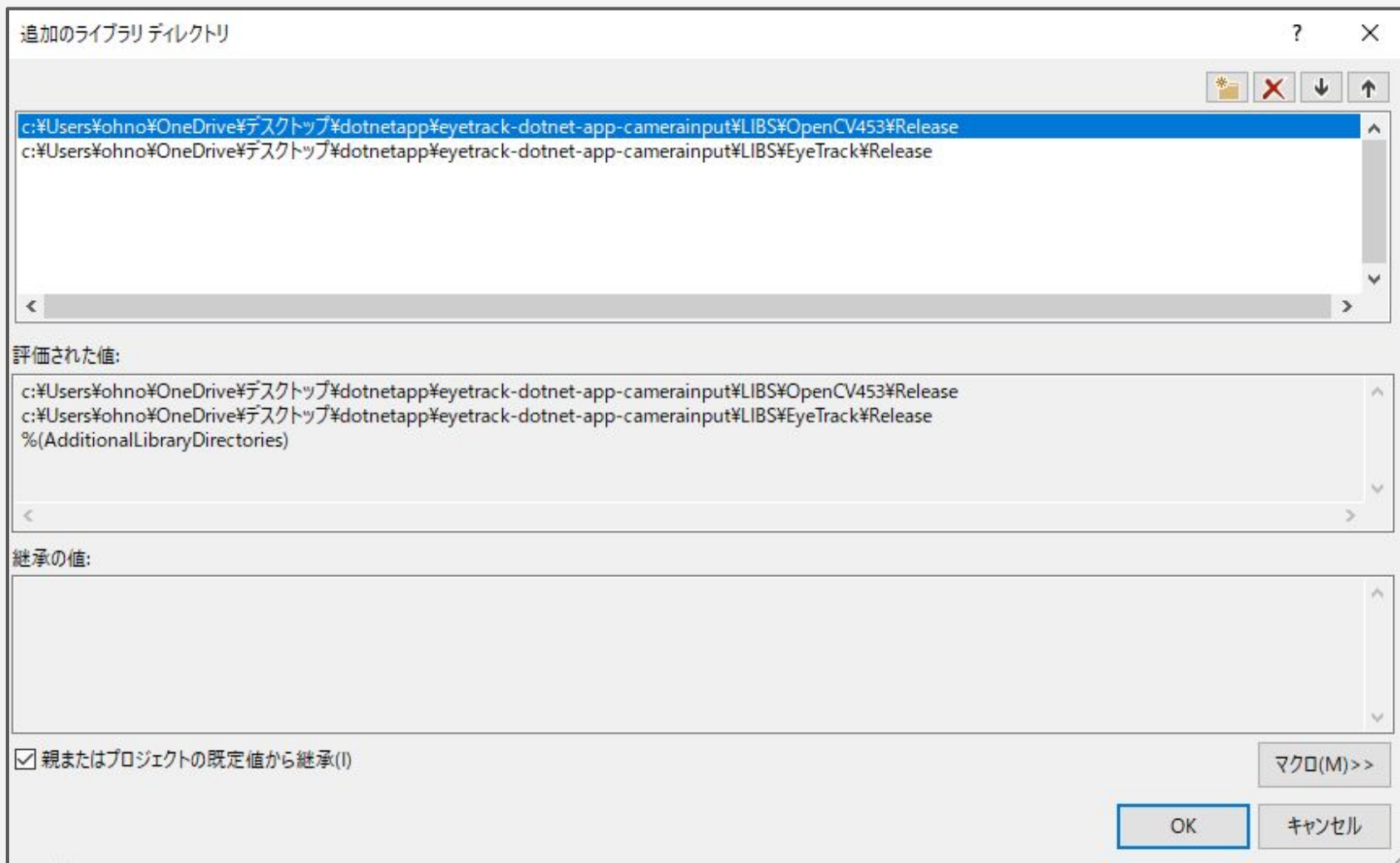
- 全般
- 詳細
- デバッグ
- VC++ ディレクトリ
- C/C++
 - 全般
 - 最適化
 - プリプロセッサ
 - コード生成
 - 言語
 - プリコンパイル済みヘッ
 - 出力ファイル
 - ブラウザー情報
 - 詳細設定
 - すべてのオプション
 - コマンドライン
- リンカー
 - 全般
 - 入力
 - マニフェスト ファイル
 - デバッグ
 - システム
 - 最適化
 - 埋め込み IDL
 - Windows メタデータ
 - 詳細設定
 - すべてのオプション
 - コマンドライン
- マニフェスト ツール
- リソース
- XML ドキュメント ジェネレー
- ブラウザー情報

オプションまたはスイッチの検索:

ライブラリ依存関係のリンク	はい
ライブラリ依存関係の入力の使用	いいえ
ランダム化されたベース アドレス	はい (/DYNAMICBASE)
リンカー警告をエラーとして扱う	
リンク ステータス	
リンク時のコード生成	
安全な例外ハンドラーを含むイメージ	
関数の順序	
固定ベース アドレス	いいえ (/FIXED:NO)
最低限必要なバージョン	
参照	
出力の登録	いいえ
出力ファイル	\$(OutDir)\$(TargetName)\$(TargetExt)
進行状況の表示	設定なし
対象コンピューター	MachineX64 (/MACHINE:X64)
大きいサイズのアドレス	
遅延署名	
遅延読み込みされた DLL のアンロード	
遅延読み込みされた DLL のバインドなし	
著作権情報の非表示	はい (/NOLOGO)
追加のマニフェスト依存関係	
追加のライブラリ ディレクトリ	%{(AdditionalLibraryDirectories)}
追加の依存ファイル	<編集...>
特定の既定のライブラリを無視	<親またはプロジェクトの既定値から継承>
分離の許可	
埋め込み IDL の無視	いいえ

■ご利用手順⑩

ご提供した.NETアプリのLIBSフォルダ下の場所をライブラリリンク先として以下を参考に設定をしてください。設定が終わったら「OK」ボタンで閉じます。
※本設定がうまくいっていない場合、ビルドエラーになります。



■ご利用手順⑪

続いてプロパティページの「構成プロパティ」→「リンカー」→「すべてのオプション」を選択し、追加の依存ファイル欄から「<編集...>」を選択します。

EyeTrackDotNETSample プロパティページ

構成(C): Release プラットフォーム(P): x64

オプションまたはスイッチの検索:

ライブラリ依存関係のリンク	はい
ライブラリ依存関係の入力の使用	いいえ
ランダム化されたベース アドレス	はい (/DYNAMICBASE)
リンカー 警告をエラーとして扱う	
リンク ステータス	
リンク時のコード生成	
安全な例外ハンドラーを含むイメージ	
関数の順序	
固定ベース アドレス	いいえ (/FIXED:NO)
最低限必要なバージョン	
参照	
出力の登録	いいえ
出力ファイル	\$(OutDir)\$(TargetName)\$(TargetExt)
進行状況の表示	設定なし
対象コンピューター	MachineX64 (/MACHINE:X64)
大きいサイズのアドレス	
遅延署名	
遅延読み込みされた DLL のアンロード	
遅延読み込みされた DLL のバインドなし	
著作権情報の非表示	はい (/NOLOGO)
追加のマニフェスト依存関係	
追加のライブラリ ディレクトリ	C:\Users\ohno\Desktop\視線検出アプリ\eyetrack-dotnet-app\Libs\Op
追加の依存ファイル	<編集...>
特定の既定のライブラリを無視	<親またはプロジェクトの既定値から継承>
分離の許可	
埋め込み IDL の無視	いいえ

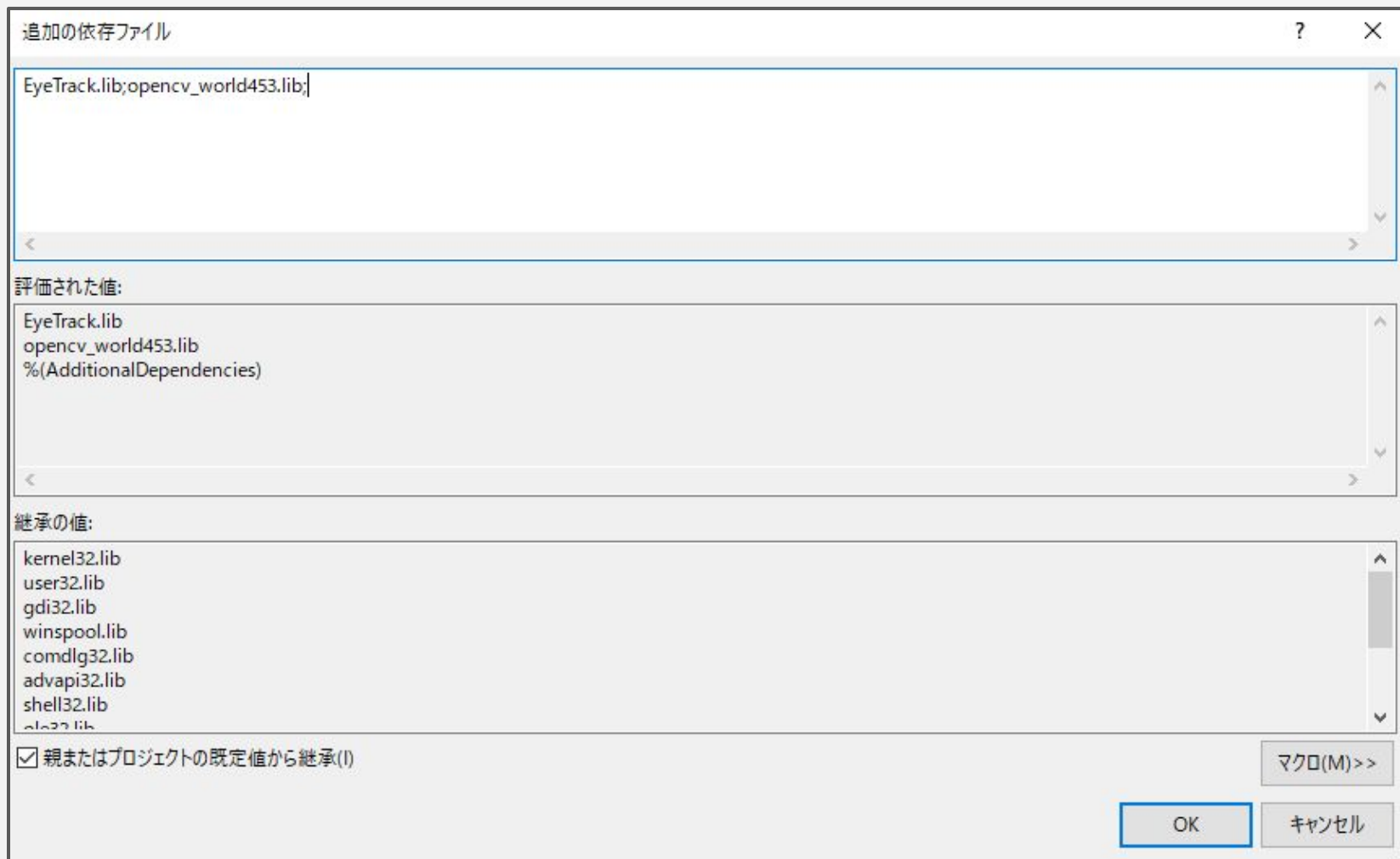
追加の依存ファイル

リンク コマンドラインに追加する項目を指定します。(例: kernel32.lib)

■ご利用手順⑫

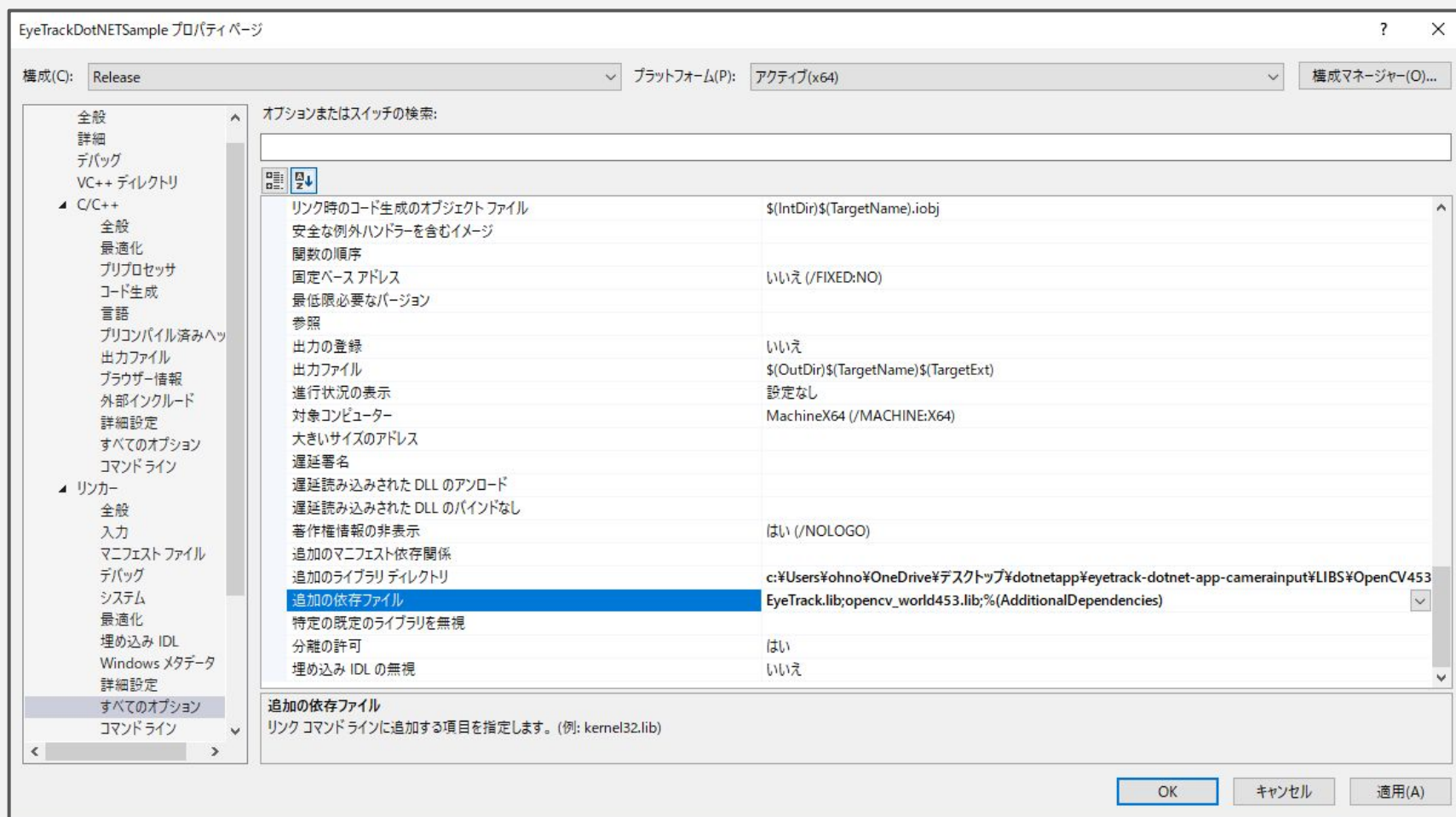
リンクするライブラリとして、OpenCVおよび視線検出ライブラリのlibファイル名をそれぞれ「;」区切りで指定します。

※本設定がうまくいっていない場合、ビルドエラーになります。



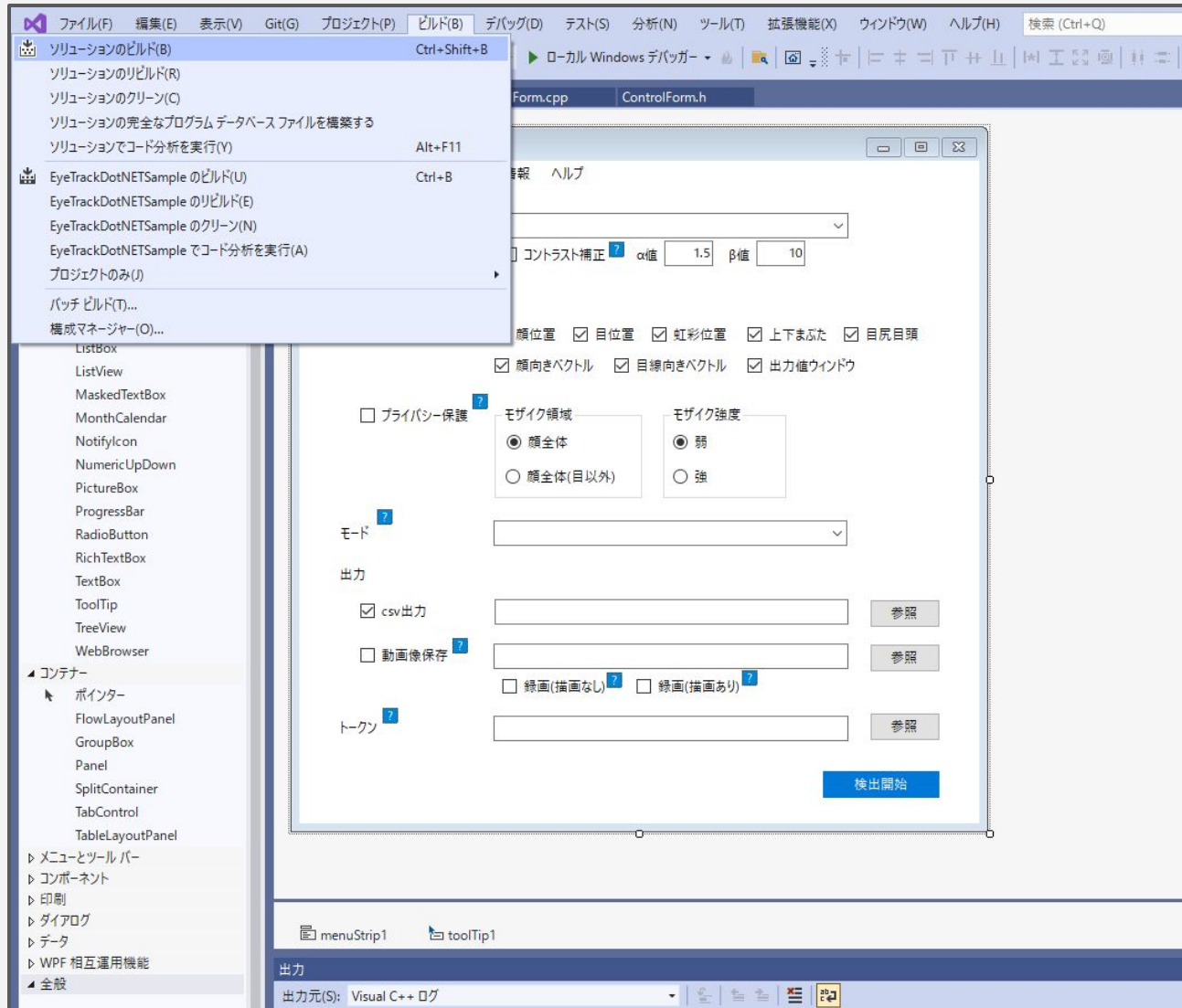
■ご利用手順⑬

すべてのプロパティの設定が終了したら、「適用」ボタンを押して、「OK」ボタンでプロパティページウィンドウを閉じてください。



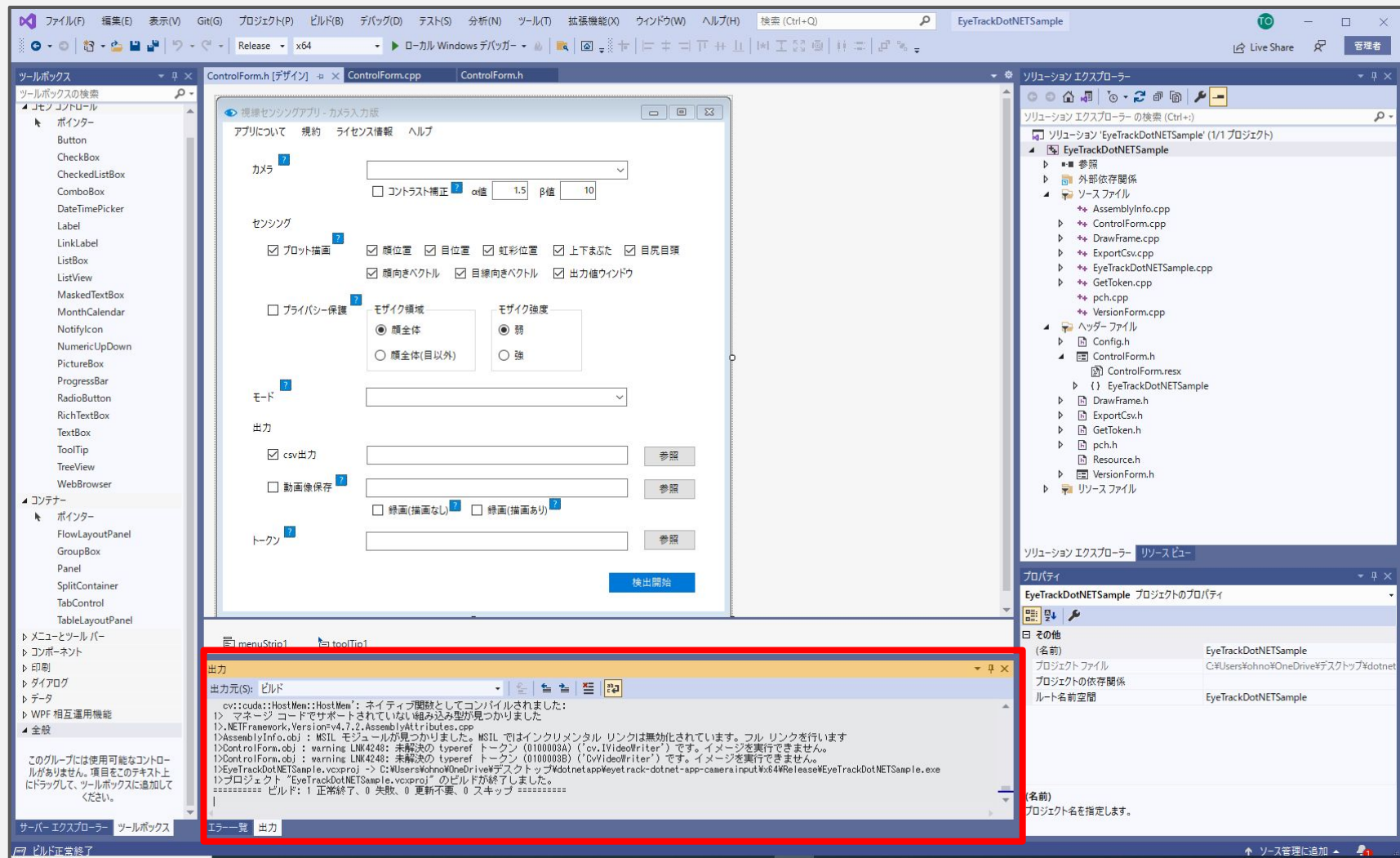
■ご利用手順⑭

ビルド準備ができましたので、Visual Studio ヘッダーより、「ビルド(B)」→「ソリューションのビルド(B)」をクリックして、ビルドを開始します。



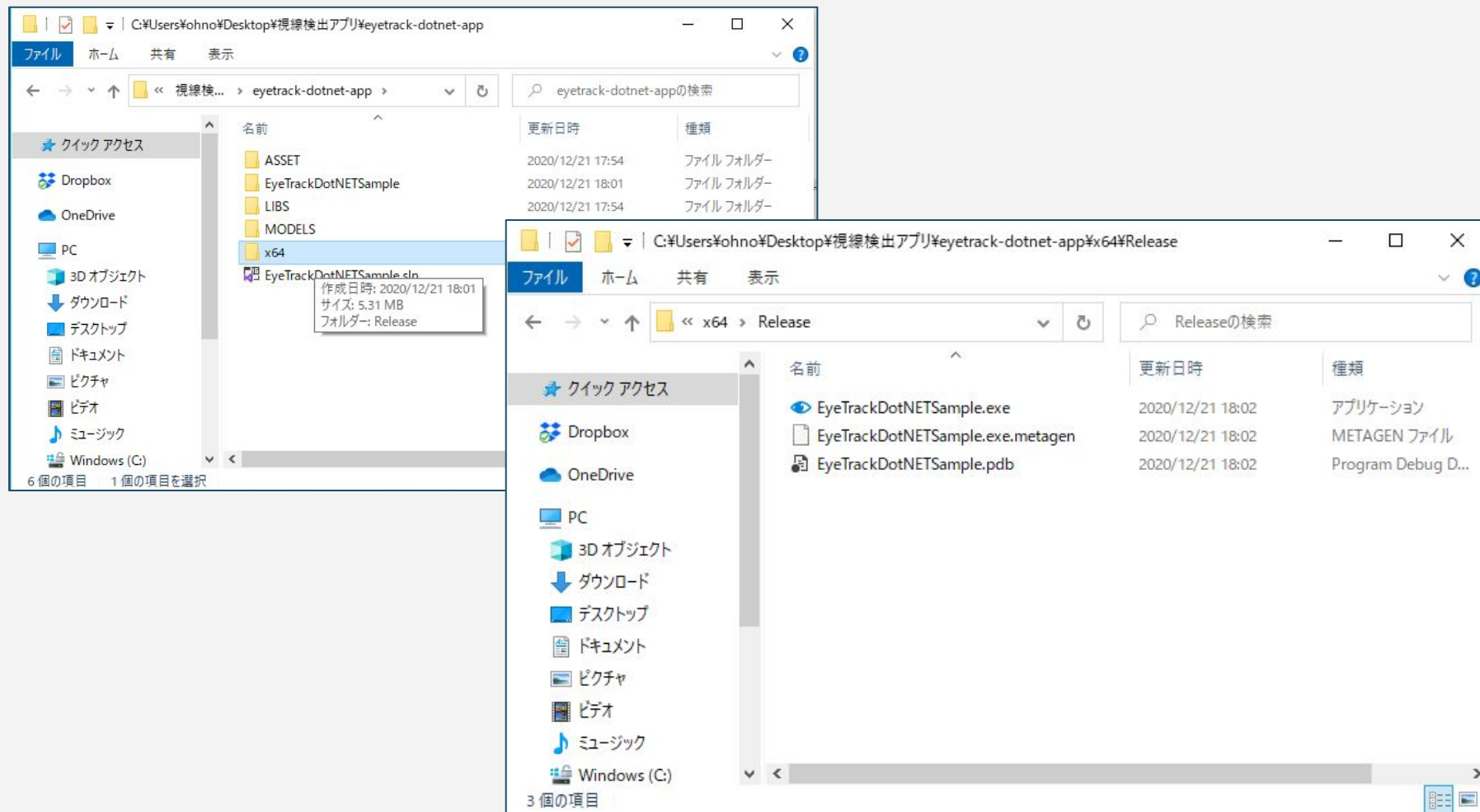
■ご利用手順⑮

ビルドが問題なく完了すると、正常終了が1、失敗が0という表示が出力ウィンドウに出て、無事にビルドが成功していることがわかります。



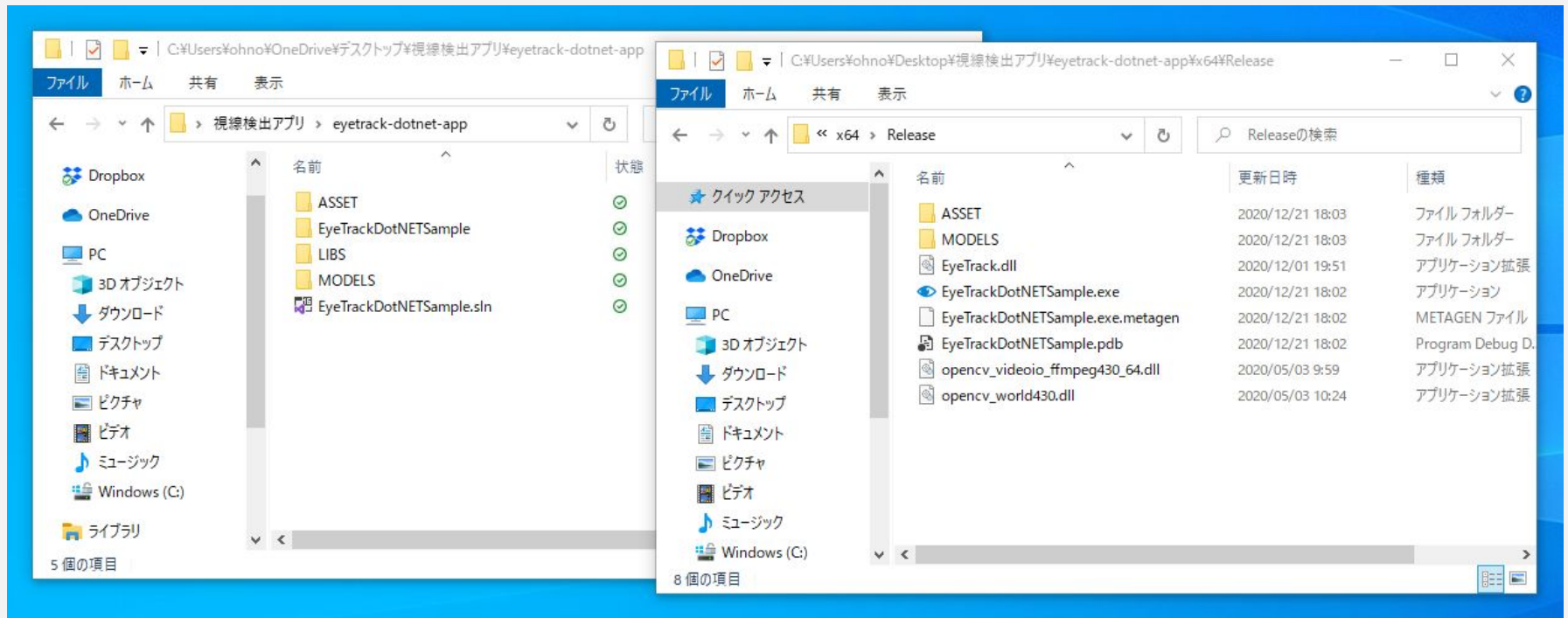
■ご利用手順①⑥

ビルドを行うと、.NETアプリフォルダの直下に、「x64」フォルダが生成されています。
「x64」→「Release」とフォルダを開き、ビルドが成功していれば
以下の通りexeファイルが生成されています。



■ご利用手順①⑦

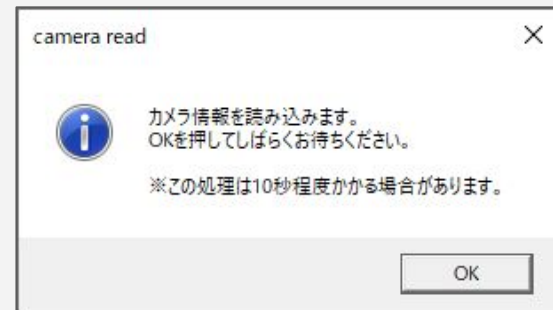
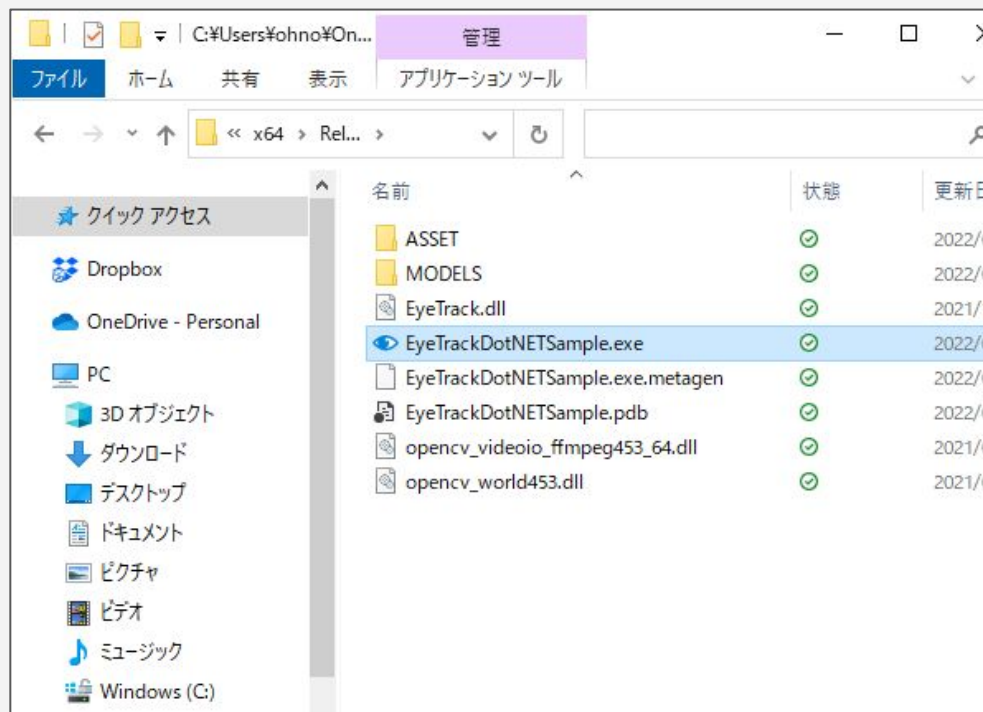
exeファイルのある実行フォルダに、アプリの実行に必要なファイルを移動します。
ご提供した.NETアプリフォルダ直下の「ASSET」フォルダ、「MODELS」フォルダを、
実行フォルダへコピーし、「EyeTrack.dll」、「opencv_world453.dll」、
「opencv_videoio_ffmpeg453_64.dll」も実行フォルダへコピーしておきます。



■ご利用手順⑱

実行準備はこれで完了です。

exeファイルをクリックすると、USBカメラの読み込みを開始します。



■ご利用手順①⑨

以下のウィンドウが立ち上がれば、SDKの組み込みは完了です。
ソースコードを編集し、お好みに応じてカスタマイズを行ってください。

視線センシングアプリ - カメラ入力版

アプリについて 規約 ライセンス情報 ヘルプ

カメラ ?

Camera 0 - (1920 x 1080) ▼

☐ コントラスト補正 ? α値 1.5 β値 10

センシング ?

☒ プロット描画 ? ☒ 顔位置 ☒ 目位置 ☒ 虹彩位置 ☒ 上下まぶた ☒ 目尻目頭

☒ 顔向きベクトル ☒ 視線向きベクトル ☒ 出力値ウィンドウ

☐ プライバシー保護 ?

モザイク領域

☒ 顔全体 ☐ 顔全体(目以外)

モザイク強度

☒ 弱 ☐ 強

モード ?

α: プロットモード ▼

出力

☒ csv出力 参照

☐ 動画像保存 ? 参照

☐ 録画(描画なし) ? ☐ 録画(描画あり) ?

検出開始

ファイル項目

■ファイル項目（弊社ソースのみ）

ファイル名	内容
ControlForm.h / .cpp	GUIウィンドウに関するプログラムです
DrawFrame.h / .cpp	検出ウィンドウの描画に関するプログラムです
ExportCsv.h / .cpp	検出ログのcsv出力に関するプログラムです
EyeTrackDotNETSample.cpp	アプリの起動に関する main関数のあるプログラムです
GetToken.h / .cpp	アクティベーショントークンを読み込みプログラムです
VersionForm.h / .cpp	バージョン情報を表示する GUIウィンドウに関するプログラムです
Config.h	設定ファイルである xmlファイルを生成するためのプログラムです