

IJCAD 2025 アプリケーション開発での特記事項

2025 年 2 月

本書では IJCAD 2025 のアプリケーション開発に関する特記事項をまとめています。

サポートコンパイラの変更

IJCAD 2020 以降は GRX と .NET API で使用するコンパイラが変更されました。

変更前：Visual Studio 2010

変更後：Visual Studio 2017（バージョン 15.9.17 以降）

開発環境の変更点

IJCAD 2024 以降で下記の開発環境が変更されました。

IJCAD 2024 では 32bit 版の提供がなくなり、64bit 版のみの提供となりました。

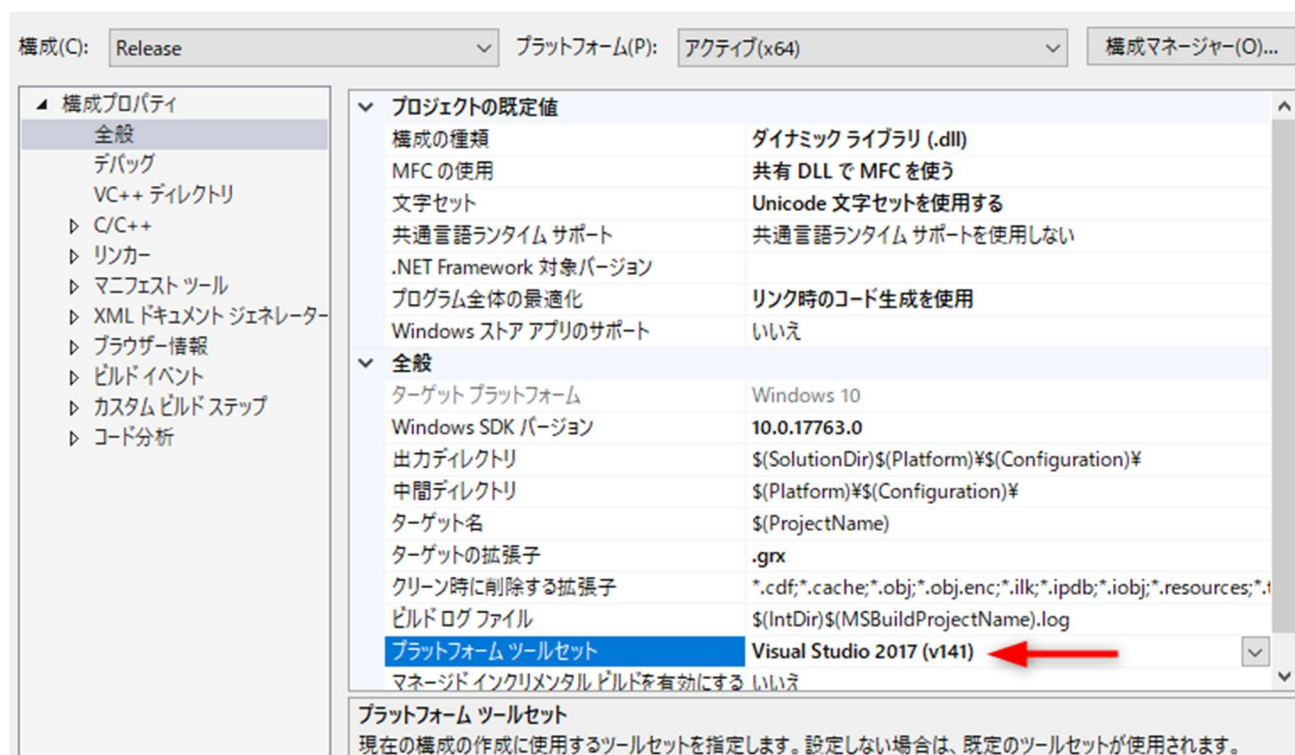
そのため、32bit 環境依存のソースコードを含む場合は、該当部分の変更が必要になります。

各詳細は開発ヘルプをご確認ください。

API	変更点
.NET API	IJCAD2020 から .Net Framework のバージョンが 4.0 から 4.8 に変更されました。 また、IJCAD2024 から下記の変更があります。 ① モジュール構成の変更 ② 名前空間の変更 ③ 一部 API の廃止/変更(AutoCAD 互換性の向上)
AutoLISP	Unicode タイプの LISP ファイルに対応しました。 本変更に伴い、LISP の文字コードを設定する機能(LISPSYS システム変数)が実装されました。
DIESEL	開発環境に変更はありませんが、バージョンアップ時はテストの実施を推奨します。
VBA	Microsoft Visual C++ 2010 再頒布可能パッケージがインストールされている環境が必要です。 該当の環境は Microsoft のサイトにて提供されています。
GRX	IJCAD2024 から下記の変更があります。 ① モジュール構成の変更 ② SDK のフォルダ構成の変更 ③ 一部 API の廃止/変更(AutoCAD 互換性の向上)

Visual Studio 2017 を用いた GRX モジュールの作成方法

Visual Studio 2017 で GRX モジュールを生成する場合、プロジェクトに設定する「プラットフォーム ツールセット」に「Visual Studio 2017 (v141)」を指定します。



「プラットフォーム ツールセット」に「Visual Studio 2017 (v141)」が表示されない場合は、Visual Studio 2017 で追加インストールが必要です。


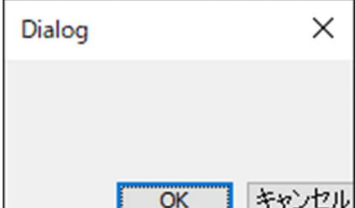
詳細は Visual Studio のヘルプをご覧ください。

GRX Visual Studio のスケルトンで stdafx.h が生成されない問題について

Visual Studio の最新版で GRX モジュールのスケルトンを生成した場合、プリコンパイル済みヘッダーのファイルが stdafx.h ではなく、pch.h と framework.h が作成されるケースがあります。この場合、stdafx.h に対する変更を framework.h に対して修正することで解決します。

4K ディスプレイ対応による変更点

IJCAD 2020 から 4K ディスプレイへ正式に対応しました。プログラムでダイアログサイズやコントロールの位置を変更している場合、以下のようなずれが発生するケースがあります。

IJCAD 2019 以前	IJCAD 2020 以降
	

原因の多くはウィンドウの枠線の幅を固定値で計算しているためです。対策としてはウィンドウ枠線の幅を OS (例、Win32 API の `GetSystemMetrics()` 関数) から取得するようにソースコードを変更する必要があります。

AutoCAD .NET API アプリケーションから移植する際の注意事項

AutoCAD で .NET API を利用したアプリケーションを IJCAD へ移植する際は以下にご注意ください (サンプルコードは C# 用です)。

using の使用

インスタンスを生成するコードでは以下のように using 構文を使用します。

変更前：

```
var mapping = new IdMapping();
```

変更後：

```
using( var mapping = new IdMapping())  
{  
    ...  
}
```

ResultBuffer の使用

ResultBuffer クラスでは、UnmanagedObject の代わりに ResbufObject を利用します。
コンストラクタで ResultBuffer(IntPtr, bool)を利用したい場合は、ResultBuffer.Create(IntPtr, bool)で代用してください。

トランザクション内での SetCurrentView の使用

トランザクション内で SetCurrentView メソッドを使用しても現在のビューは更新されません。
トランザクション終了後に更新するようにしてください。(＃1925)

GetOrthoProjectedCurve を用いた作図の違い

GetOrthoProjectedCurve メソッドで得られた情報を元に作図をする場合、AppendEntity ではなく、AddNewlyCreatedDBObject を呼び出してください。(＃4078)

AutoCAD と IJCAD の既定のロックモードの違い

Application.DocumentManager.MdiActiveDocument.LockMode()で返される結果が AutoCAD と IJCAD とでは異なります。(＃1750)

- ① コマンドを CommandFlags.Modal で登録した場合
 - ・ AutoCAD : Write が返されます。
 - ・ IJCAD : NotLocked が返されます。
- ② コマンドを CommandFlags.Session で登録した場合
 - ・ AutoCAD : NotLocked が返されます。
 - ・ IJCAD : ProtectedAutoWrite が返されます。

DrawJig クラスで描画したオブジェクトがちらつく問題

DrawJig クラスを使ってラバーバンドなどを描画した場合、AutoCAD と比較すると、IJCAD ではちらついて見えるケースがあります。こちらは制限となります。(＃5636)

CloseAndSave を実行すると例外が発生する問題

CloseAndSave を実行すると例外が発生する場合があります。この問題を解決するにはコマンド登録時に CommandFlags.Modal ではなく、CommandFlags.Session を指定してください。(＃6968)

ダイアログのボタンを押したタイミングでブレイクポイントを設定しても停止しない問題

.NET アプリケーションでダイアログのボタンをクリックしたイベント内でブレイクポイントを設定しても、Visual Studio のデバッガではプログラムを停止することはできません。(＃1312)

AutoCAD から移植する際のその他の注意事項

.NET API 以外で AutoCAD から IJCAD へ移植する際の注意事項をまとめました。

複数図面を開いた状態でアクティブの図面を変更した際の挙動の違い

複数図面を開いている状態で、コマンド実行中にアクティブ図面を変更して再び元の図面をアクティブに戻した場合、IJCAD ではコマンドが強制終了されずに継続されます。AutoCAD (2015) 以降では実行中のコマンドが強制的にキャンセルされます。

DCL でスライドファイルが表示されない問題

AutoLISP などで、DCL を使ってスライドファイルを表示する際、スライドファイルが存在するフォルダ名によって問題が発生することがあります。(＃1806)

① フォルダ名に「(」と「)」(半角括弧)の両方がある場合、スライドファイルのイメージが表示されません。フォルダ名に半角括弧を使用しなければ解決します。

(例) C:¥(sample)

※この現象は AutoCAD でも発生します。

② フォルダ名に「)」(半角閉じ括弧)のみがある場合、IJCAD では異常終了します。フォルダ名に半角閉じ括弧を使用しなければ解決します。

(例) C:¥sample)

VBA の GcadDocument.EndCommand イベント内で他のコマンドを呼び出したときの制限について

VBA の GcadDocument.EndCommand イベント内で、SendCommand 文を使って他のコマンドを呼び出した場合、無限ループに陥ることがあります。これは IJCAD がコマンドキャンセル時に、CancelCommand ではなく EndCommand を呼び出すためです。EndCommand イベント内では他のコマンドを呼び出さないでください。(＃2121)