

IJCAD 2023 アプリケーション開発での特記事項

2023 年 2 月

本書では IJCAD 2023 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 でお使いいただく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 2020 以降は .NET API と GRX の開発環境が変更されました。

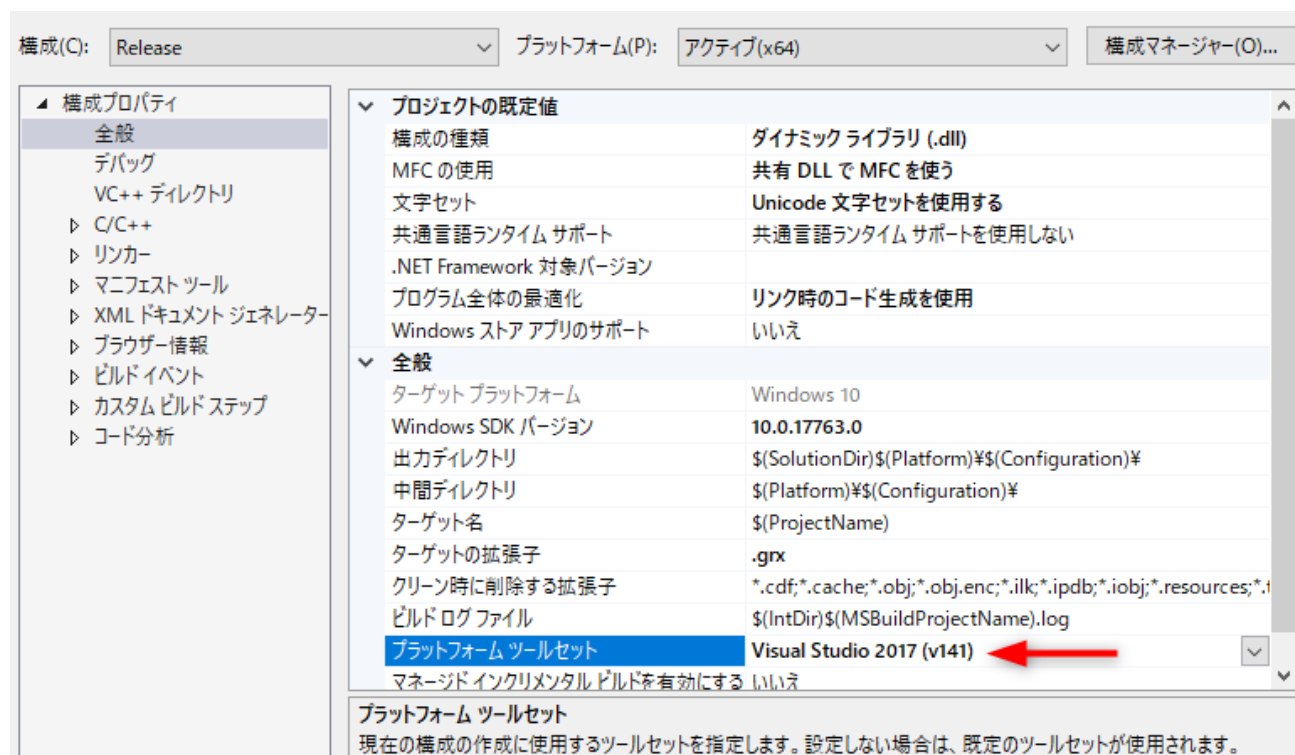
API	変更点
DIESEL	変更はありません。
AutoLISP	変更はありません。
VBA	変更はありません。
.NET API	サポートされる .Net Framework が「4.0」から「4.8」に変更されました。 それに伴い、IJCAD 2023 アセンブリファイルを参照し直し、.Net Framework を「4.8」で再ビルドする必要があります。 一部のクラスやメソッド、プロパティが変更されているケースがありますので、その場合はソースコードに適切な修正を加えてください。
GRX	前バージョンの IJCAD とは非バイナリ互換となりました。 過去のバージョンの IJCAD（GRXSDK）で作成された GRX モジュールはそのままではロードできません。 新しい開発環境である Visual Studio 2017 を使い、IJCAD 2023 用 GRXSDK に参照を変更して、再ビルドする必要があります。

IJCAD 2023 の開発環境は IJCAD 2020～IJCAD 2022 と同じです。ただし、以下の点にご注意ください。

API	変更点
.NET API	サポートされる .Net Framework は「4.8」です。IJCAD 2020 以降に作成したアプリケーションは再ビルドせずに動作するはずですが、テストの実施を推奨します。
GRX	Visual Studio を使い、IJCAD 2023 用 GRXSDK に参照を変更して、再ビルドしてください。

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。



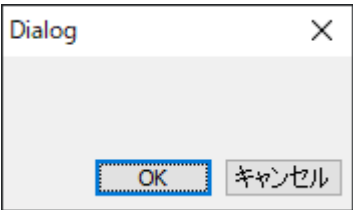
「プラットフォーム ツールセット」に「Visual Studio 2017 (v141)」が表示されない場合は、Visual Studio 2017 で追加インストールが必要です。詳細は Visual Studio のヘルプをご覧ください。

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 が作成される場合があります。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場合があります。

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ケースがあります。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)

図形選択解除時の SelectionAdded と SelectionRemoved の違い

コマンドアイドル状態（入力待ち）の際、図形を選択して ESC キーを押して選択を解除したときに通知する挙動が IJCAD と AutoCAD と異なっています。(＃3845)

AutoCAD の動き：SelectionAdded が通知されます。

IJCAD の動き：SelectionRemoved が通知されます。

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、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)

LISP の文字コード対応の違いについて

AutoCAD 2021 にて追加された Unicode タイプの LISP ファイルは、IJCAD には対応していません。IJCAD は、Shift_JIS などのマルチバイト文字セット(MBCS)タイプのファイルに対応しています。AutoCAD 2021 以降の Unicode 対応の LISP アプリケーションを IJCAD への移植する際は、文字コードに対応させるための修正が必要になる場合があります。(＃8099)