AUTODESK
DevCon

Inventor SDKを用いたAPIカスタマイズ開発 再入門

加藤 丈博

免責事項

本イベントでのプレゼンテーションには、当社の見通し、将来の実績および関連する仮定、獲得可能な最大市場規模、買収、製品および製品能力、戦略に関する将来の見通しに関する記述が含まれる場合があります。これらの記述は、現在判明している要因に基づく当社の最善の判断を反映したものです。実際の出来事や実績は大きく異なる可能性があります。当社の実績が将来の見通しに関する記述と異なる原因となりうる重要なリスクおよびその他の要因については、 www.sec.gov で入手可能な最新の Form 10-K および Form 10-Q を含む当社の SEC 提出書類をご参照ください。

これらのプレゼンテーションにおける将来の見通しに関する記述は、プレゼンテーション実施日当日時点でのものです。 これらのプレゼンテーションが実施日当日時点以降に見直される場合、たとえその後当社が当社の Web サイトその他で利用可能にしたとしても、それらのプレゼンテーションには最新または正確な情報が含まれていない可能性があります。当社は、将来の見通しに関する記述を更新または修正する義務を一切負いません。

当社の製品およびサービスに関する計画済みまたは将来的な開発努力に関する記述は、製品、サービス、または機能が将来利用可能になることを約束または保証することを意図したものではなく、単に当社の現在の計画を反映したものであり、現在当社が把握している要素に基づくものです。これらの記述に依存して購入の意思決定を行うべきではありません。

アンケート：InventorのAPIを使ったことはありますか？

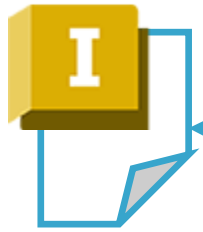
- InventorのVBA環境
- iLogic
- .net アドイン
- アペレンテス
- エクセルなどのマクロとの連携

アジェンダ

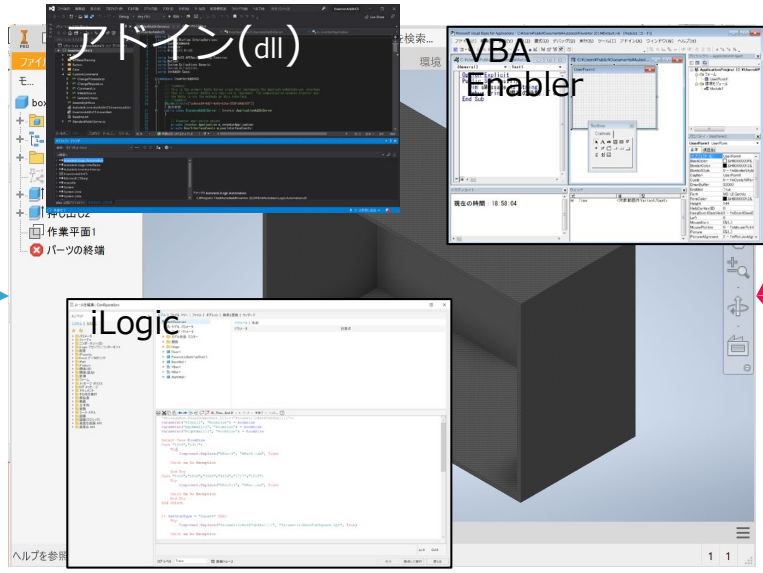
1. Inventor APIを用いたカスタム開発概要
2. Inventor オブジェクトモデル概要
3. Inventor カスタム開発のTips
4. Inventor 2024 API アップデート

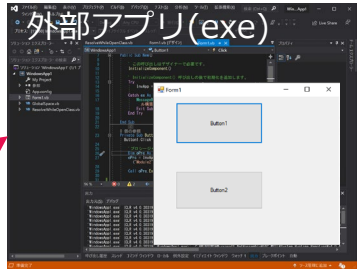
Inventor APIを用いたカスタム開発概要

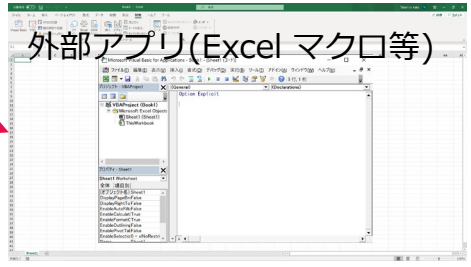
Inventor APIを用いたカスタマイズ手法

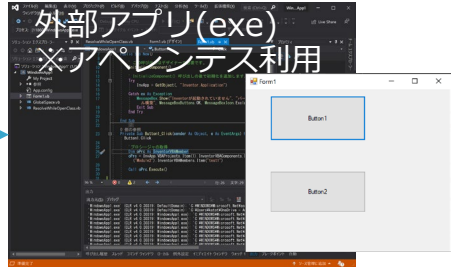


Inventor file
(*.ipt, *.iam,
*.idw, *.dwg)

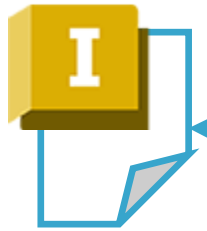




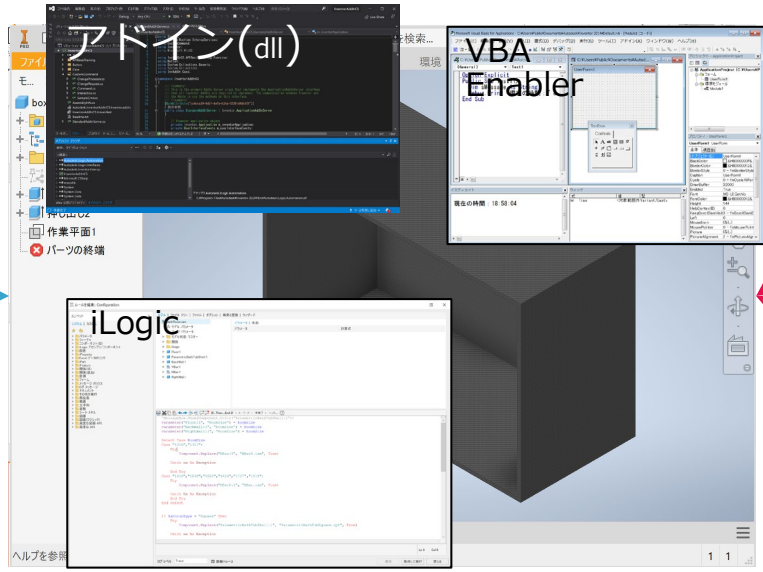


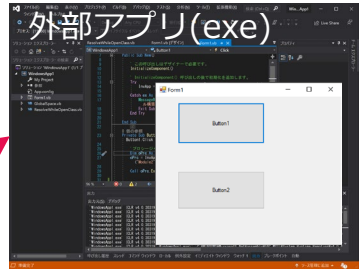


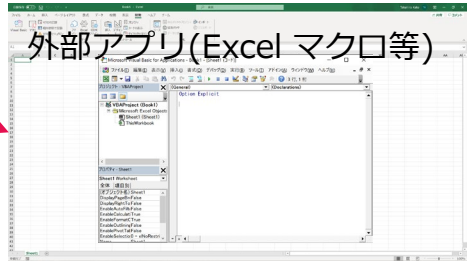
アペレンテスとほかの手法の違いは？

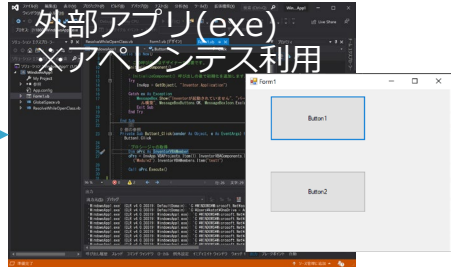


Inventor file
(*.ipt, *.iam,
*.idw, *.dwg)









アペレンテス

- アペレンテスはActiveXコンポーネント
- アペレンテスを使用するクライアントのインプロセスで動作
- アペレンテスはInventor APIのサブセットを提供
- スタンドアロンで動作 (Inventorのインストールは不要)
- 無償で利用可能 (Inventor Apprentice Serverインストーラを公開サイトよりダウンロード)
- アペレンテスはプロセス外からInventorに接続する場合と比較して効率的に動作

アペレンテス

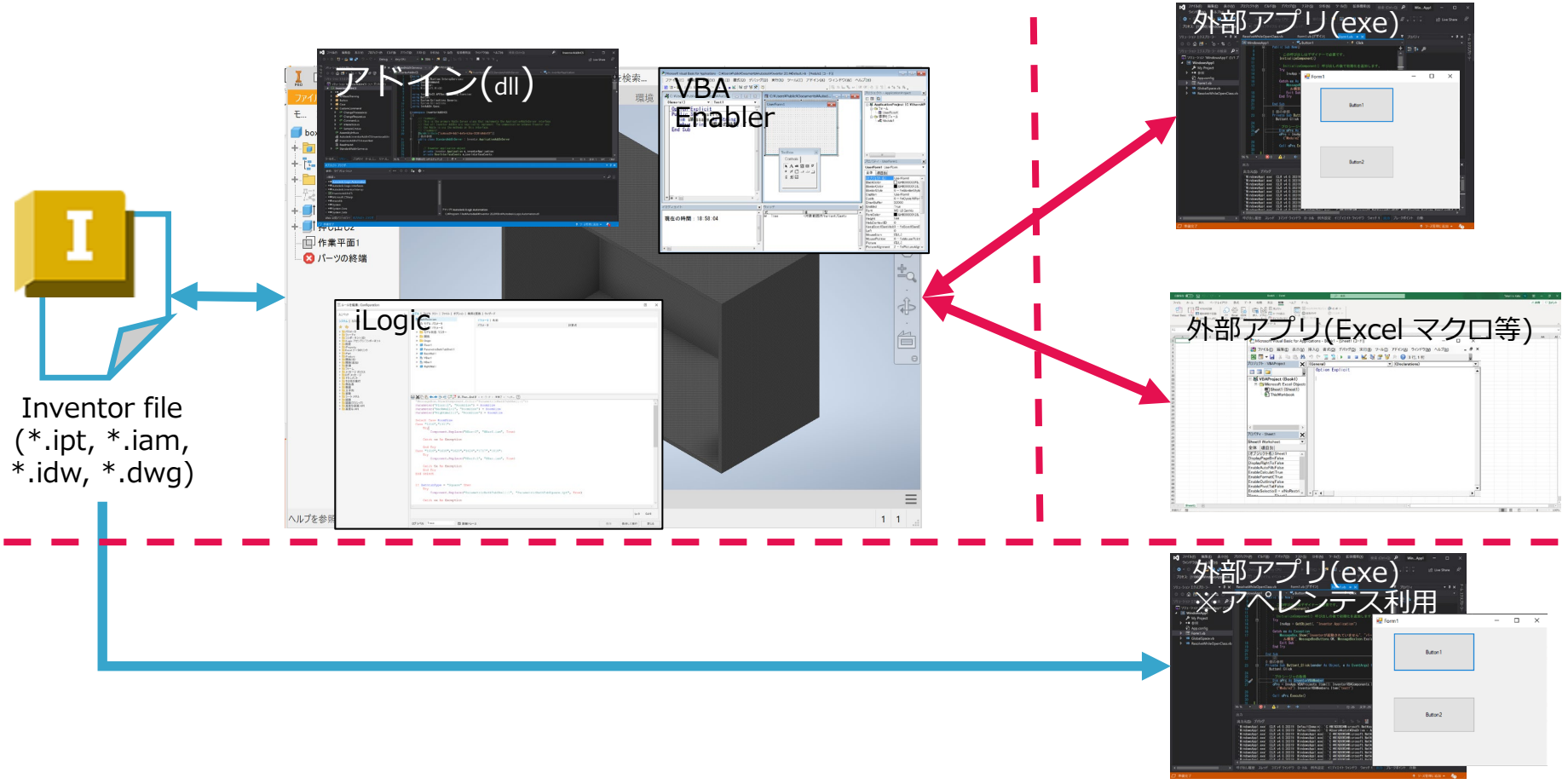
- アペレンテスの機能
  - アペレンテスにより以下に対して**読み取りアクセス**が可能
    - ✓ Assembly structure
    - ✓ B-Rep
    - ✓ Drawing sheets and views (limited access)
    - ✓ iParts
    - ✓ iAssemblies
    - ✓ BOM
  - iProperties、アトリビュート、ファイルリファレンスへの読み取り及び**書き込み**アクセス

アペレンテス

使用上の注意

- Inventorのプロセス内での利用は不可
  - ✓ Add-In内での利用は不可
  - ✓ InventorのVBAからの利用は不可
- “*new ApprenticeServerComponent*”でインスタンスを作成
- アペレンテスにはドキュメントコレクションは存在しない
- *Document*オブジェクト
  - ✓ *ApprenticeServerDocument* (*.ipt &*.iam)
  - ✓ *ApprenticeServerDrawingDocument* (*.idw)
- アペレンテスでは、以前のバージョンのInventorファイルを保存することは出来ない。アペレンテスで保存をする前に、Inventorで開いて、保存をすることでファイルをマイグレーションする必要がある
- Inventor 2022の新機能モデル状態へアクセスするAPIは、アペレンテスでは提供されず、アペレンテスのAPIでは、最後にInventorファイルが保存された際にアクティブなモデル状態へアクセスをする

外部アプリはとそのほかの手法の違いは？



外部プロセスからのInventorの操作

外部からのInventorの制御

- GetObject(Path, ProgID)またはCreateObject(ProgID)関数で、InventorのApplicationオブジェクトを取得
  - ✓ GetObject ・・・起動中のInventorのApplicationオブジェクトを取得
  - ✓ CreateObject・・・新しくInventorを起動し、起動したInventorのApplicationオブジェクトを取得

- InventorのProgIDは“Inventor.Application”
  - ✓ AutoCADのように**バージョンを指定して**Inventorを起動することはできない

- .Netでは、System.Runtime.InteropServices.Marshal.GetActiveObject（GetObject関数に相当）およびSystem.Activator.CreateInstance（ CreateObject 関数に相当）を使用

外部プロセスからのInventorの操作

使用上の注意

- Inventor APIは64bit版のライブラリのみであるため、APIを呼び出す外部プロセスも64bitである必要がある
  - ✓ 独自アプリの場合は、Visual Studioのビルドオプションで**64bitアプリをビルド**する
  - ✓ Excel のVBAの場合、**64bit版のExcel**を使用する

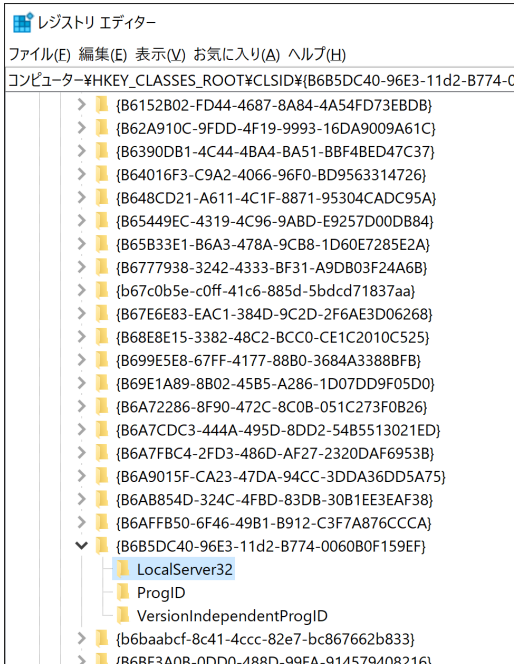
- 外部プロセスからInventorのエディタを通じてAPIを実行するため、インプロセスと比較してAPIの実行に時間がかかる（プロセス間通信によるオーバヘッド）

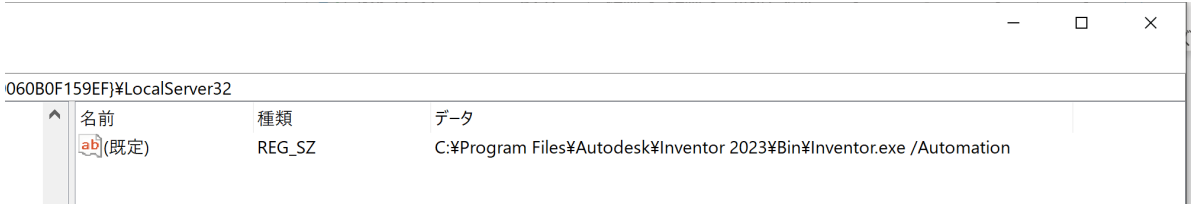
- 表示状態に依存するAPIは、Inventorのエディタを表示状態としてAPIを実行する必要がある

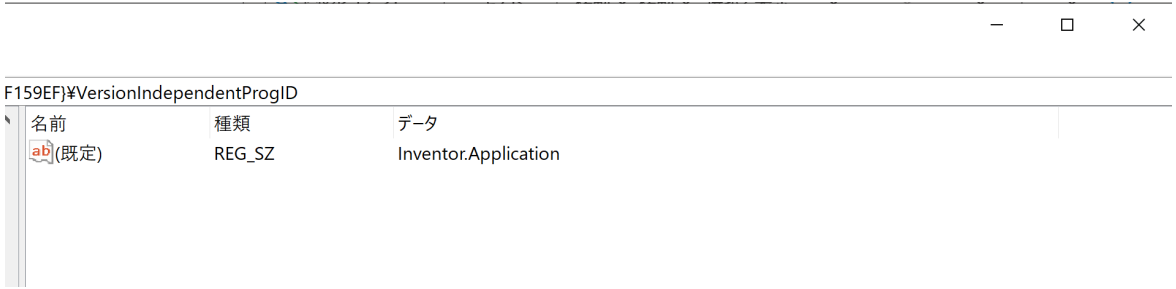

なぜ別プロセスをCreateObjectで Inventorを起動できるの？


外部プロセスからのInventorの起動

レジストリから情報を取得してプロセスを起動しているから







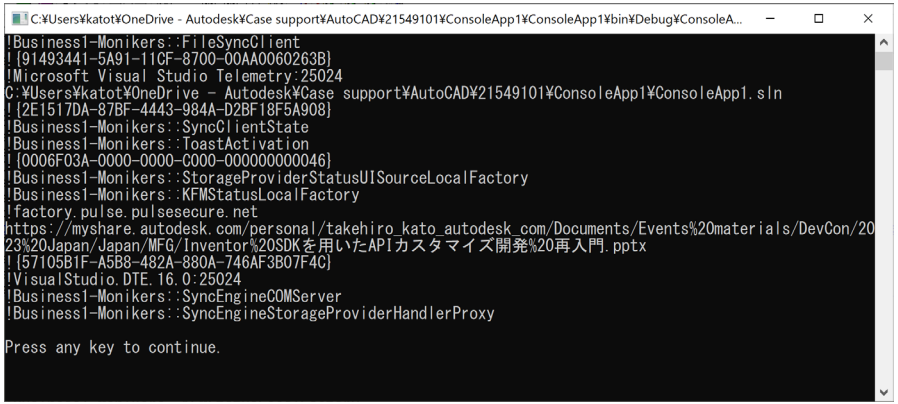

なぜGetObjectで起動中のInventorにアクセスできるの？


外部プロセスからのInventorの操作

ROT（Running Object Table）に登録されているから

ROT: 実行中（*R*unning）のCOMオブジェクト（ActiveX サーバ*O*bject) の一覧（*T*able）

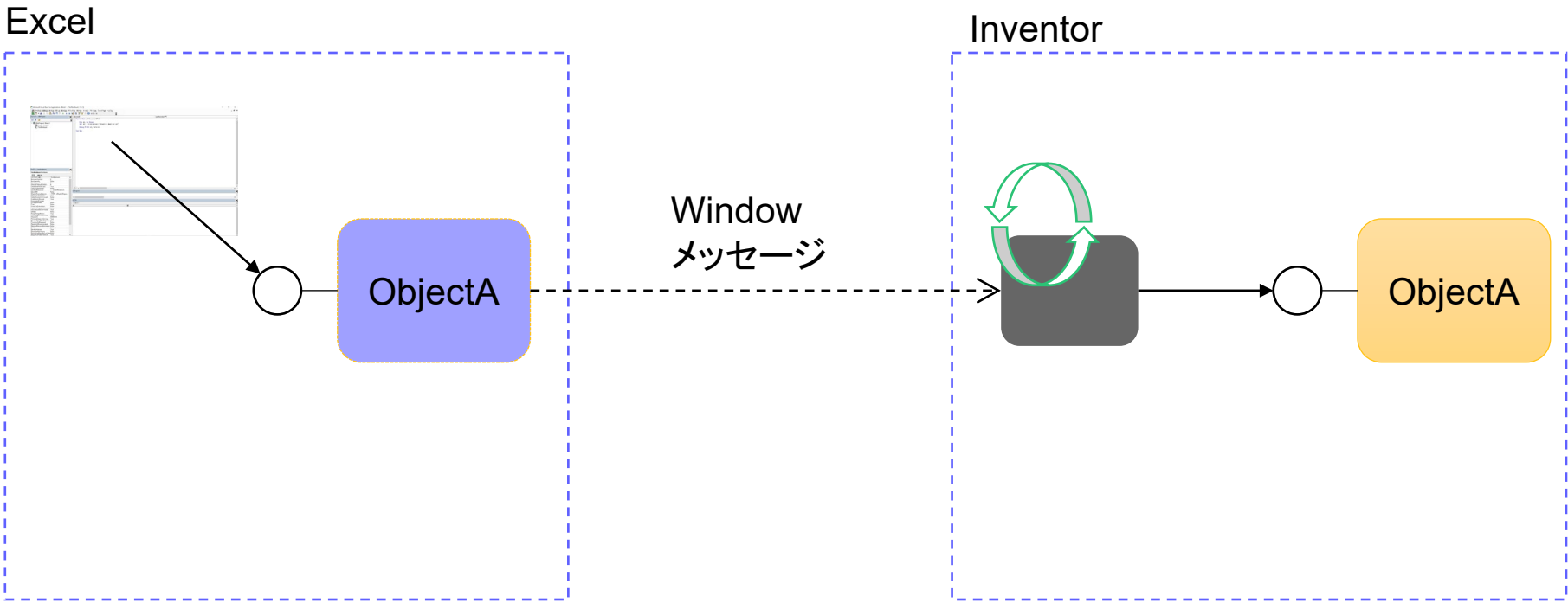
InventorはExe起動時に、ROTに登録している。ROTへの登録は任意なのですべてのアプリケーションのCOMオブジェクトを取得できるわけではありません



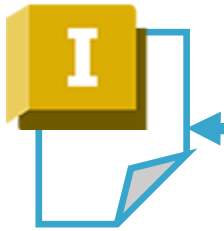

なぜ外部プロセスから、InventorのAPIを実行できるの？


外部プロセスからのInventorの操作

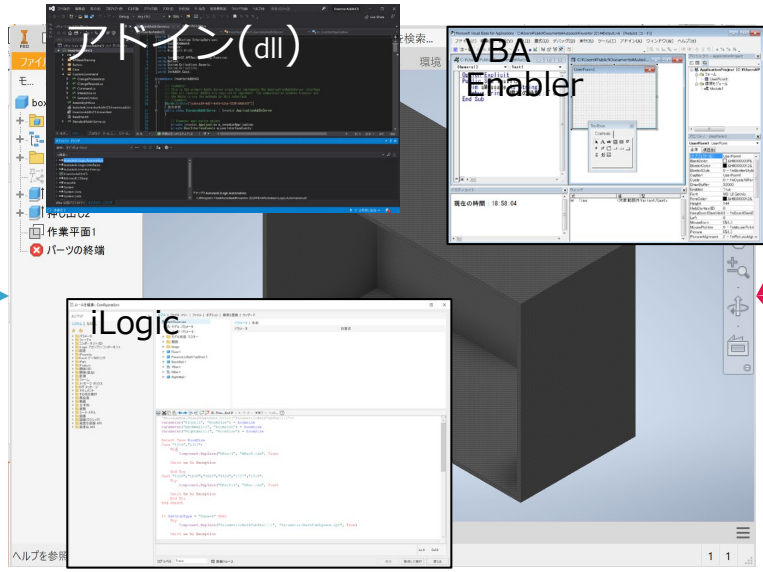
Windowメッセージ

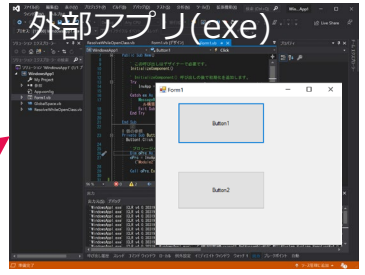


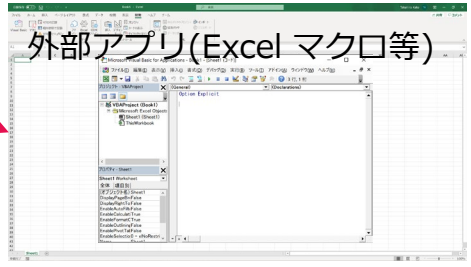
インプロセスのカスタマイズ手法

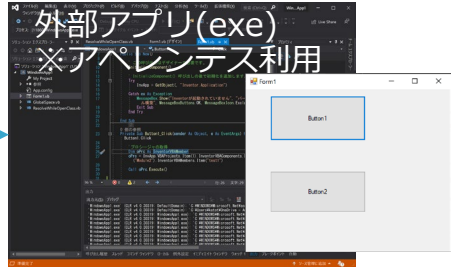


Inventor file
(*.ipt, *.iam,
*.idw, *.dwg)









InventorアドインとVBAの比較

- Inventorアドインの利点
  - ✓ スタートアップ時のロード
  - ✓ VB、C#などの新しい開発言語が使用可能（C++、F#、Delphiなども使用可能）
  - ✓ InventorのUIに完全に統合
  - ✓ 主にInventorのコマンドの開発に利用
  - ✓ 高度なInventorイベントのハンドリング
  - ✓ インストーラ開発が可能
  - ✓ ソースコード管理が容易
  - ✓ ソースコードのセキュリティ

- VBAの利点
  - ✓ 開発ツールが不要（Inventorアプリケーションに包含）
  - ✓ 容易なRapidプロトタイプが可能
  - ✓ 使いやすいオブジェクトブラウザー
  - ✓ Inventor objectを参照しながらのデバッグが容易

Inventorアドイン

Visual Studio アドインWizard

- Visual Studioで、Inventor Addinを作成するためのコードテンプレート、ビルド設定、Inventor APIライブラリへの参照設定等を含む、Visual Studioのプロジェクトを生成するヘルパーツール
- Addin Wizard（を含むDeveloper Tool）のインストーラーの場所

C:¥Users¥Public¥Documents¥Autodesk¥Inventor <version>¥SDK

配下のDeveloperTools.msi

- インストール手順
  - Visual Studio professional 2015以降がインストールされていることを確認
  - **管理者権限**でコマンドプロンプトを起動
  - コマンドプロンプトよりmsiexec.exeを iオプションで起動しインストーラを起動

    例）msiexec.exe /i <DeveloperTools.msi ファイルパス>
- アドインの作成
  - Visual Studio を起動し、プロジェクトの新規作成から、Inventor 2024のAddinテンプレートを選択（C++、VB、C#のいずれかが使用可能）

Inventorアドイン

Visual Studio アドインWizardで生成されたテンプレートソースコード

```
    <ProgIdAttribute("InventorAddIn2.StandardAddInServer"), _
    GuidAttribute("8b801945-a84a-4e48-a572-3fbff56b9ca9")> _
    1 個の参照
    Public Class StandardAddInServer
        Implements Inventor.ApplicationAddInServer

        Private WithEvents m_uiEvents As UserInterfaceEvents
        'Private WithEvents m_sampleButton As ButtonDefinition

#Region "ApplicationAddInServer Members"

        ' This method is called by Inventor when it loads the AddIn. The AddI
        ' to the Inventor Application object. The FirstTime flag indicates if
        ' the first time. However, with the introduction of the ribbon this a
        0 個の参照
        Public Sub Activate(ByVal addInSiteObject As Inventor.ApplicationAddI

        ' This method is called by Inventor when the AddIn is unloaded. The A
        ' unloaded either manually by the user or when the Inventor session i
        0 個の参照
        Public Sub Deactivate() Implements Inventor.ApplicationAddInServer.De

        ' This property is provided to allow the AddIn to expose an API of its
        ' programs. Typically, this  would be done by implementing the AddIn's
        ' interface in a class and returning that class object through this pr
        0 個の参照
        Public ReadOnly Property Automation() As Object Implements Inventor.A

        ' Note:this method is now obsolete, you should use the
        ' ControlDefinition functionality for implementing commands.
        0 個の参照
        Public Sub ExecuteCommand(ByVal commandID As Integer) Implements Inve

#End Region
```

- Activate

  addinがInventorにロードされた際に呼び出されるメソッド。主にアドインで実装する処理であるコマンドや、コマンドを起動するためのユーザインターフェイス（リボンのボタン等）の構築・登録の処理を記述

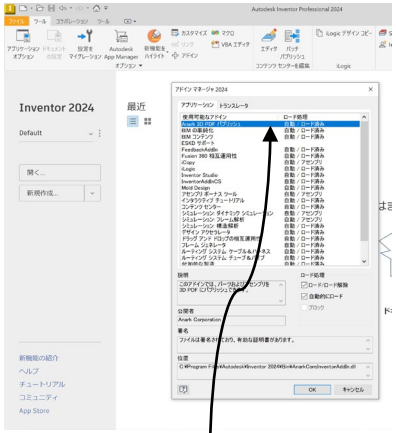
- Deactivate

  addinがアンロードされた際に呼び出されるメソッド。主にActivateメソッドで構築・登録をしたユーザインターフェイスの登録解除等の処理を記述

Inventorアドイン

レジストリ登録不要（Regfree）なアドイン

- レジストリ登録が不要なCOMコンポーネント
- レジストリに代わり、.addinファイルで、様々なセッティングを定義
- Inventorアプリケーションの所定のディレクトリ内に.addinファイルを置くことでInventorアプリケーション起動時に.addinファイルをロード
- カスタムアドインdllがどこにあるかを.addinファイルに記述。カスタムアドインdllは、任意のフォルダーに置く事が可能



*.addin

```
ddin Type="Standard">
  <ClassId>{9c93ff52-e2fa-4ec4-8c4a-5747ad0bafef}</ClassId>
  <ClientId>{9c93ff52-e2fa-4ec4-8c4a-5747ad0bafef}</ClientId>
  <DisplayName>Attribute Helper</DisplayName>
  <Description>Utility to manage API attributes.</Description>
  <Assembly>AttributeHelper.dll</Assembly>
  <LoadOnStartUp>1</LoadOnStartUp>
  <UserUnloadable>1</UserUnloadable>
  <OSType>Win64</OSType>
  <Hidden>0</Hidden>
  <SupportedSoftwareVersionGreaterThan>26..</SupportedSoftwareVersionGreaterThan>
  <DataVersion>1</DataVersion>
  <UserInterfaceVersion>1</UserInterfaceVersion>
</Addin>
```

*.dll

Inventor APIの概要

.addinファイルの配置先

- バージョンに非依存（addIn ファイルの SupportedSoftwareVersionxxx の記述を無視）

  C:¥ProgramData¥Autodesk¥Inventor Addins¥

- バージョンに依存(.addInファイルの SupportedSoftwareVersionxxx の記述が有効)
  - ✓ Inventor 2023まで

    C:¥ProgramData¥Autodesk¥Inventor <*Inventor version*>¥Addins¥
  - ✓ Inventor 2024から

    C:¥Program Files¥Autodesk¥Inventor <*Inventor version*>¥Bin¥Addins

- ユーザー毎にオーバーライド

  C:¥Users¥<user>¥<AppData¥Roaming¥Autodesk¥ Inventor <Inventor version>¥Addins¥

アドインの登録解除は、**.addin**ファイルを配置したディレクトリより
Autodesk.<AddInName>.Inventor.addin ファイルを削除

VBA（Visual Basic for Applications）

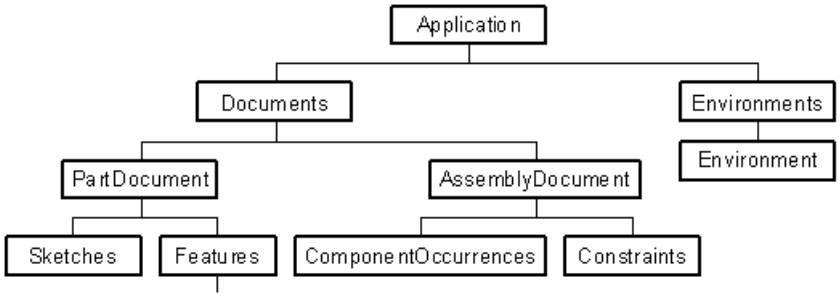
- Visual Basic for Applications (VBA)は、Microsoft により開発されたプログラミング環境
- Inventorの一部として、アプリケーション内に埋め込み無償で供給され、Inventorのデータ内にプログラムを埋め込むことが可能
- Inventorのリボンメニューのツール(T) → Visual Basic Editor(V)、またはAlt-F11で起動
- 主な使用用途は、エンドユーザーが“マクロ”を作るため。作成したマクロは、“マクロ“コマンドまたは、リボン タブの カスタマイズ ダイアログの マクロに登録して実行。（詳細は、製品 オンライン ヘルプの リボンタブの カスタマイズ ダイアログのリファレンス”を参照）
- 64BitOS上でネイティブに実行されるため、Microsoftがリリースしている**32Bit用のコントロールなどは動作しない**ので、フォームなどへの組み込みには注意が必要
- インタプリタ言語（コンパイルが不要）なため、実行時に**InventorのObject Modelを参照しながら**作業が可能

VBA（Visual Basic for Applications）

Inventor VBA環境 既定の変数

- ThisApplication変数（スコープ：グローバル）
  - ✓ Applicationオブジェクトの参照を提供

❑ Inventor API Object Model

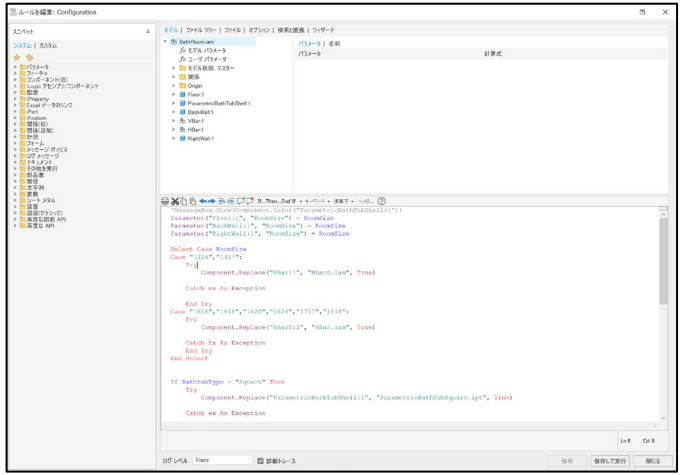


- ApplicationオブジェクトはInventor API Object Modelの最上位のオブジェクト

- ThisDocument 変数（スコープ：ドキュメントプロジェクト）
  - ✓ プロジェクト内に有るドキュメントの参照を提供

iLogic

- Inventorのエディタに組み込みこまれた、自動化処理を容易に作成する事を可能とする機能
- 様々な**iLogic固有の関数やコードスニペット**を利用することが可能
- 独自の入力フォームを作成する事も可能
- 作成したルールは、Inventorファイル内または外部ファイルに保存して実行が可能
- iLogicには、VB.netの文法で**Inventor APIを実行する**ことも可能。
- InventorのiLogicエディタには、ステップ実行や変数の値を参照する等のデバッグを行うための機能は無い

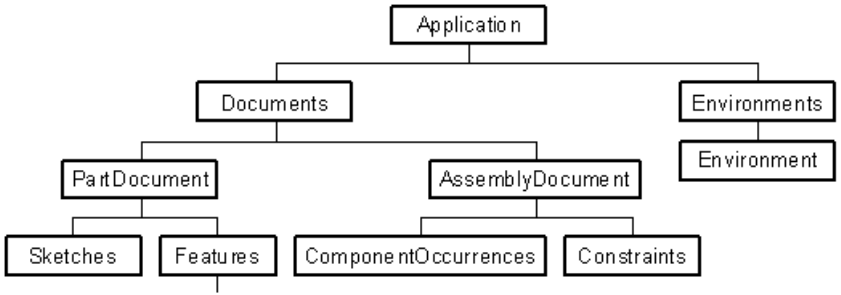


Inventor API
オブジェクトモデル概要

Inventor Object Model

Inventor Object Modelの概要

❑ Inventor API Object Model

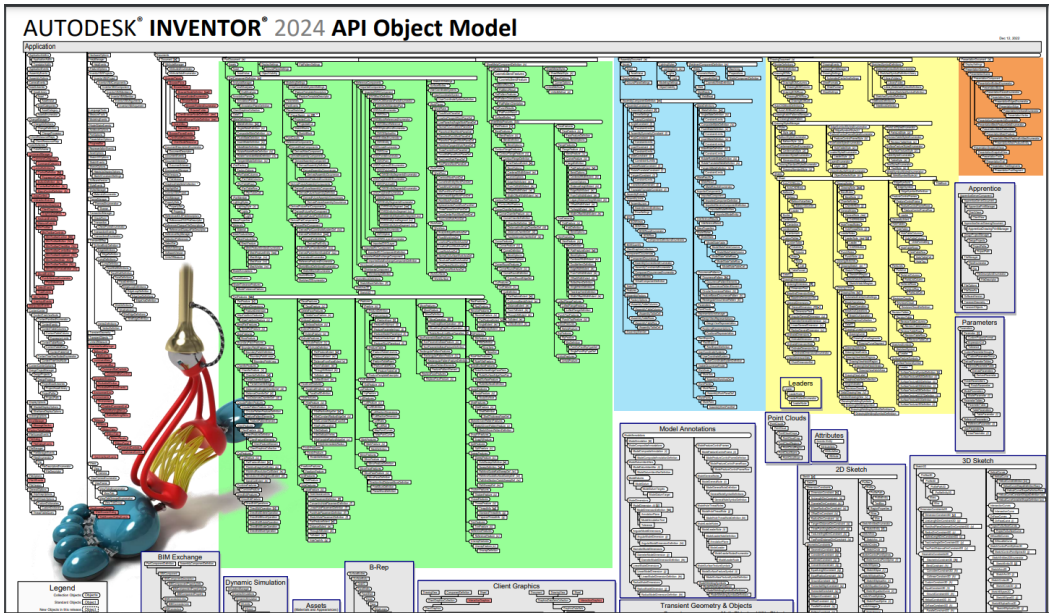


- InventorのAPIは、COM Automationの機能により Object 群として公開され、それぞれのオブジェクトがアプリケーション内で、いずれかと一致
- それぞれのオブジェクトはメソッド、プロパティ、イベントをサポート
- オブジェクトはObject Model を通してアクセスされる
- Object Modelの最上位のオブジェクトは Application オブジェクト。

Inventor オブジェクトモデル

Inventor オブジェクトモデル 構造図

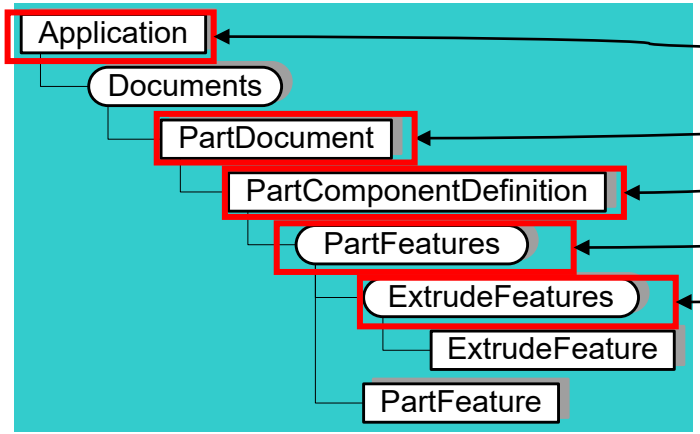
- C:¥Users¥Public¥Documents¥Autodesk¥Inventor <version>¥SDK 配下のDeveloperTools.msiをインストール後に、C:¥Users¥Public¥Documents¥Autodesk¥Inventor <version>¥SDK¥DeveloperTools¥Docs配下に展開されるInventor<version>ObjectModel.pdfファイル



Inventor オブジェクトモデル

- 基本的には、同一のオブジェクトを表すコレクションオブジェクト*XXXXXX*sの要素として、*XXXXXX*オブジェクトがあり、*XXXXXX*オブジェクトプロパティとして、その配下の*YYYYY*sオブジェクトコレクションがあり…といった構造をたどることでオブジェクトを取得することができる

  例）Application→Documentsコレクション → Document→…



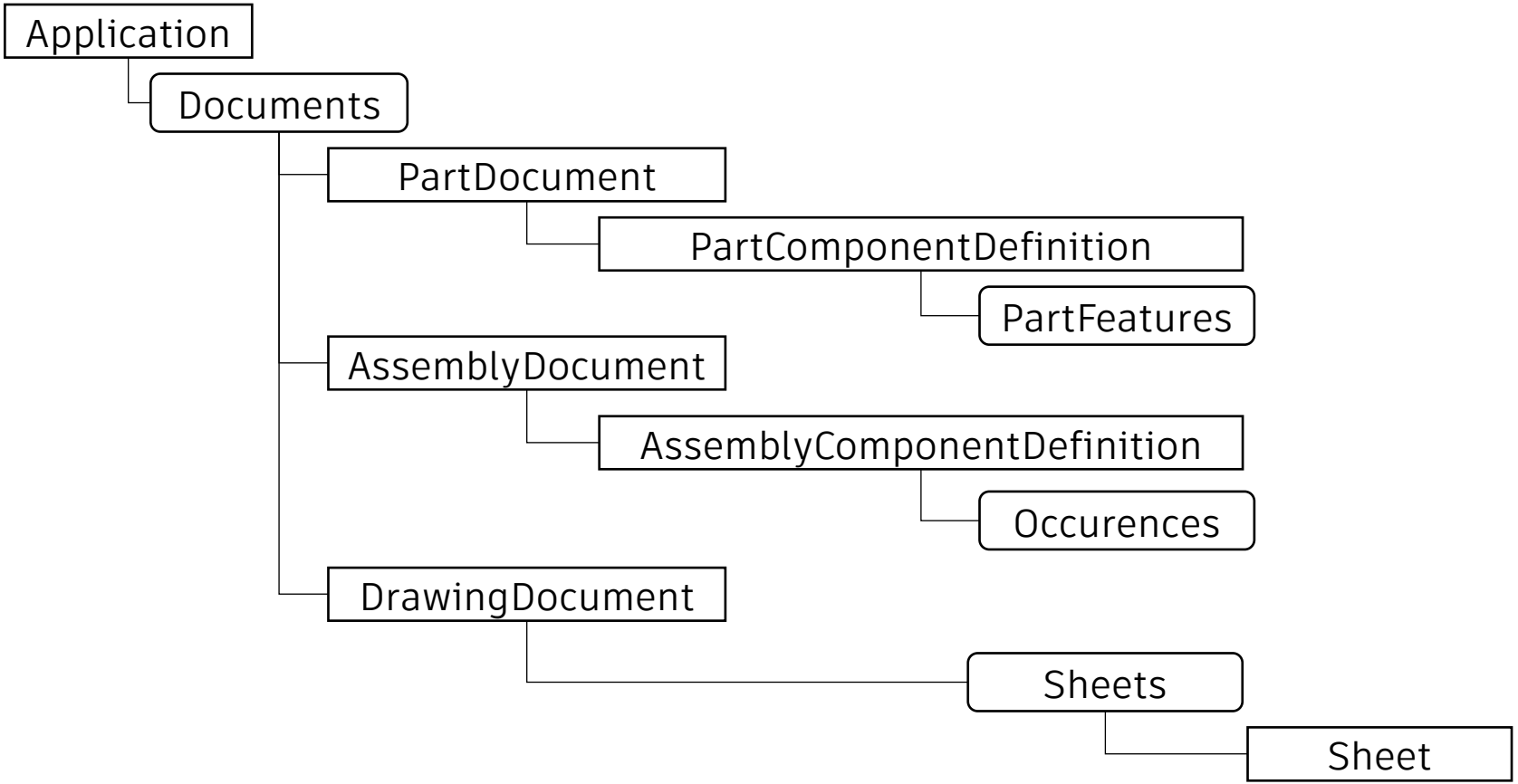

```
Public Sub GetExtrudeFeature()

    Dim oPartDoc As PartDocument
    oPartDoc = ThisApplication.ActiveDocument

    Dim oExtrude As ExtrudeFeature
    oExtrude =
oPartDoc.ComponentDefinition.Features.ExtrudeFeatures("Extrusion
1")
    MsgBox("Extrusion " & oExtrude.Name & " is suppressed: " &
oExtrude.Suppressed)

End Sub
```

Inventor ドキュメントのオブジェクトモデル



データベース単位

- 全てのInventorドキュメントは同じ内部単位（データベース単位）を使う
  - ✓ Length: センチメートル
  - ✓ Angle: ラジアン
  - ✓ Time: 秒
  - ✓ Mass: キログラム
- ドキュメント設定ダイアログで、エンドユーザーが指示した単位に内部単位を変換
- UnitsOfMeasure オブジェクトは、APIを通して同様の機能を提供

Inventor 2024
API アップデート

アドイン配置パスの変更

.addinファイルの配置先

- バージョンに依存(.addInファイルの SupportedSoftwareVersionxxx の記述が有効)
  - ✓ Inventor 2023まで
    C:¥ProgramData¥Autodesk¥Inventor <*Inventor version*>¥Addins¥
  - ✓ Inventor 2024から
    C:¥Program Files¥Autodesk¥Inventor <*Inventor version*>¥Bin¥Addins

レジストリフリーアドインへの完全移行

- Inventor 2022から、レジストリ登録を行う形式で作成されたアドインは、Inventorの Add-Insマネージャ ダイアログに表示されなくなる。

- Inventor 2023で レジストリ登録を行う形式で作成されたアドインを引き続き使用する場合、レジストリ キーの値を作成して有効にする必要がある。

  キーの場所: [HKEY_CURRENT_USER¥SOFTWARE¥Autodesk¥Inventor¥RegistryVersion27.0¥System¥Preferences]

  キー値の名前: LoadRegisterBasedAddins

  キー値データ: dword:00000001

- Inventor 2024からは、**レジストリ登録が必要なアドインはサポート対象外**となり利用できない。既存のレジストリ登録を行うアドインはRegfree形式への変換を行う必要がある。

Inventor 2024で追加・更新・削除されたAPI

https://help.autodesk.com/view/INVNTOR/2024/ENU/?guid=WhatsNew

- 0：削除されたオブジェクト
- 19：新規追加オブジェクト （237の新規関数）
- 6：削除された関数
- 30：既存オブジェクトへの新規関数の追加
- 2：関数の変更

Inventor 2023

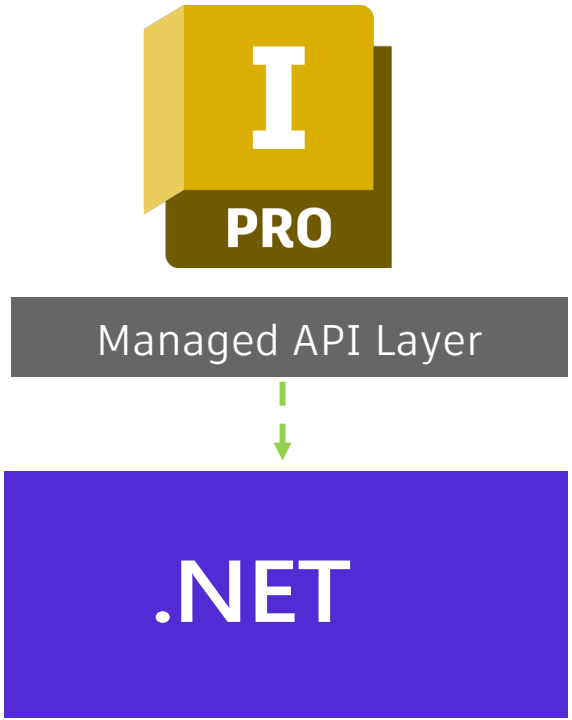
- 0：削除されたオブジェクト
- 20：新規追加オブジェクト (273の新規関数)
- 16：削除された関数
- 92：既存オブジェクトへの新規関数の追加
- 3：関数の変更

変更内容の詳細はAPIリファレンス（日本語版 Inventor 2024 API プログラミング用ヘルプ）の「新機能」を参照

.NET Core 対応 次期メジャーリリース

NET Core, NET 6.0/7.0/8.0 and NET Platform all are synonyms indicating modern technology.

- マイクロソフトは、クローズドなWindows テクノロジーから、オープンソースかつクロスプラットフォームライブラリへの移行の最中
- 新しい.NETは.NET Coreと呼ばれている
- .NET Core 自体Webアプリケーション開発で広範囲で利用されている
- 既存のInventor アドインはリビルドが必要
- Inventor APIのみの観点からは大きな変更は不要。
- .NET Frameworkの機能は.NET Coreのライブラリを利用する形に修正が必要（名前空間や提供APIが異なる）
- 3rd Partyライブラリは.Net Core対応版への変更が必要
- Inventor フォーラムのfeedback/beta プログラムで情報を発信



.NET Core テクノロジーを包含

Inventor カスタム開発 Tips


APIを使ってやりたいことはあるけど、どのInventor APIを使えばいいのかがわからない

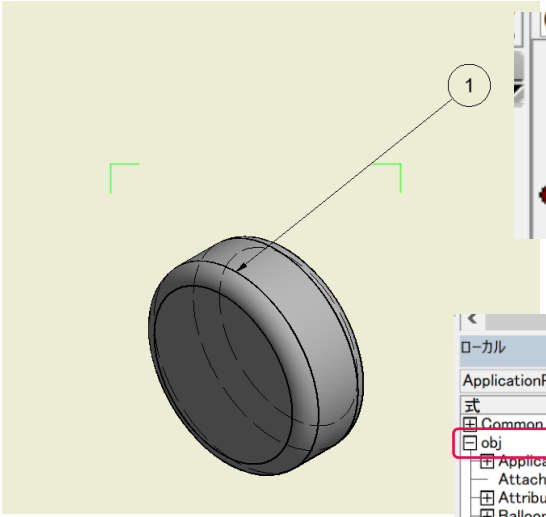

Inventor API リファレンス

- 日本語版のAPIリファレンスをchm形式ファイルで公開
  - ✓ 日本語版 Inventor 2021 API プログラミング用ヘルプ
  - ✓ 日本語版 Inventor 2022 API プログラミング用ヘルプ
  - ✓ 日本語版 Inventor 2023 API プログラミング用ヘルプ
  - ✓ 日本語版 Inventor 2024 API プログラミング用ヘルプ

※APIリファレンスにはサンプルコードが多数記載されています。

- セルフトレーニング形式のInventor API基礎トレーニングマテリアル
  - ✓ Autodesk Inventor 2021 の APIトレーニングマテリアル
  - ✓ Autodesk Inventor 2022 の APIトレーニングマテリアル
  - ✓ Autodesk Inventor 2023 の APIトレーニングマテリアル
  - ✓ Autodesk Inventor 2024 の APIトレーニングマテリアル

Inventor オブジェクトの調査




```
(General)                                                    pick
Option Explicit

Public Sub pick()
    Dim obj As Object
    Set obj = ThisApplication.CommandManager.pick(kAllEntitiesFilter, "Pick")
End Sub
```

ローカル

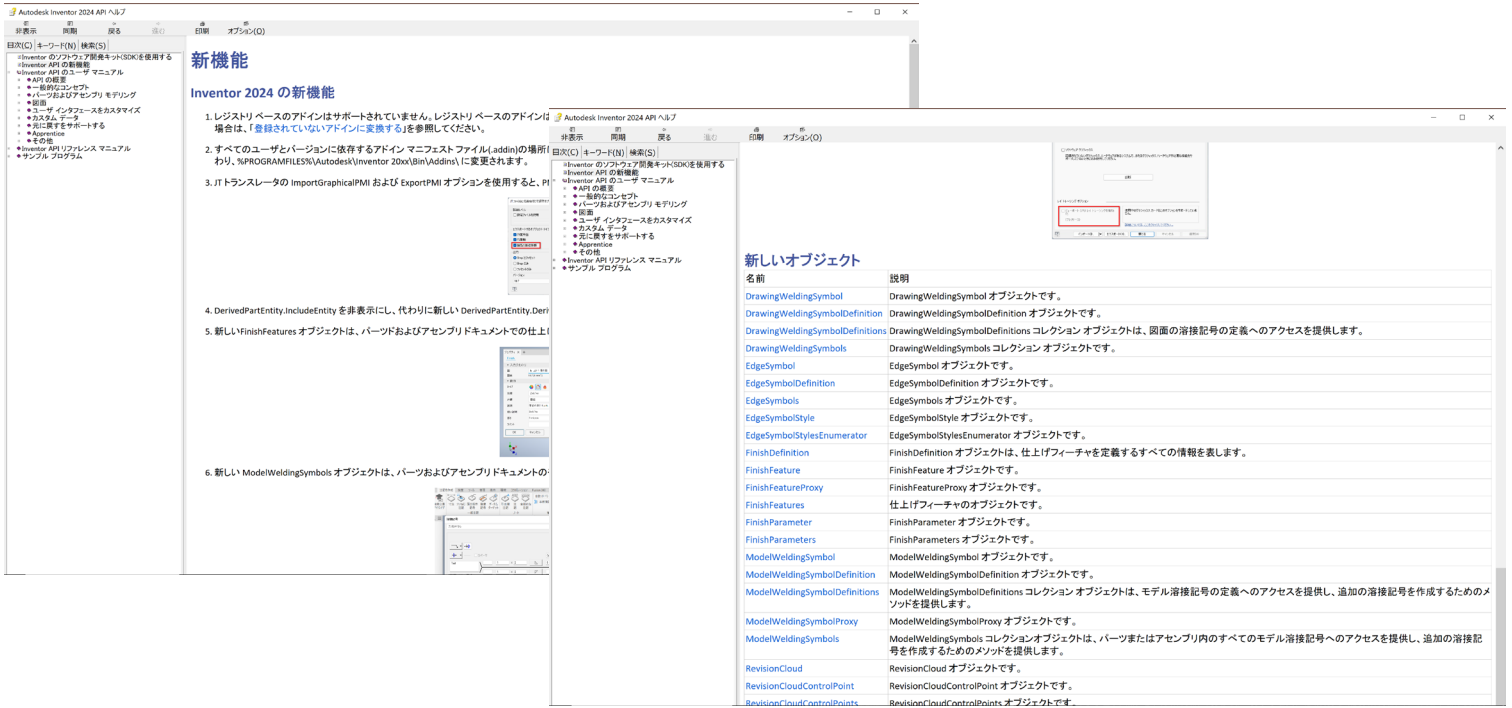
ApplicationProject.Common.pick

式	値	型
Common		Common/Common
obj		Object/Balloon
Application		Object/Application
Attached	True	Boolean
AttributeSets		AttributeSets/AttributeSets
BalloonValueSets		BalloonValueSets/BalloonValueSets
Layer		Layer/Layer
Leader		Leader/Leader
Parent		Sheet/Sheet
ParentView		DrawingView/DrawingView
PlacementDirection	kRightDirection	BalloonDirectionEnum
Position		Point2d/Point2d
Style		BalloonStyle/BalloonStyle
Type	kBalloonObject	ObjectTypeEnum

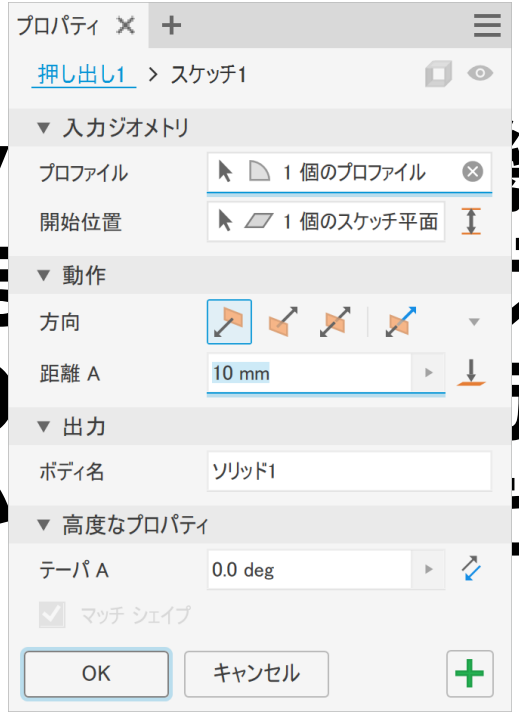

既存のソースコードを新しいバージョンに対応したいが、APIの更新点のリストなどはありますか？


Inventor API リファレンス

- 日本語版のAPIリファレンスの「新機能」のページに各バージョンでの更新点が記載されています。



APIからInv　　能を実行する
ことはできま　　えば押し出し
コマンドの　　方向や距離を
APIか　　たいです



Inventor コマンドの実行

- Inventorの各コマンド（カスタムコマンドを含む）はCommandManagerオブジェクトからControlDefinition オブジェクトとして取得し、取得したControlDefinition オブジェクトのExecuteメソッドを実行することで、コマンドを実行することが可能です

```
' Get the CommandManager object.
Dim oCommandMgr As CommandManager = ThisApplication.CommandManager
' Get control definition.
Dim oControlDef As ControlDefinition = oCommandMgr.ControlDefinitions.Item(“PartExtrudeCmd")
' Execute the command.
Call oControlDef.Execute
```

- ただし、コマンド実行時にダイアログが表示されるタイプのコマンド（例：押し出し コマンド）の場合、ダイアログ内で指定するパラメータ（押し出しコマンドの場合、押し出し方向や距離 等）の指定についてはコマンドの機能となり、**コマンドを実行するAPIから制御（指定）することが出来ません**。

参考：Inventor のAPIでコマンドを実行する方法


CreateObject()がエラーとなり、Inventorの起動に失敗します

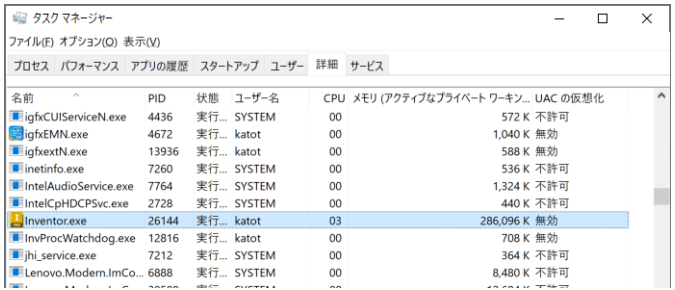

外部プロセスからのInventorの操作

使用上の注意

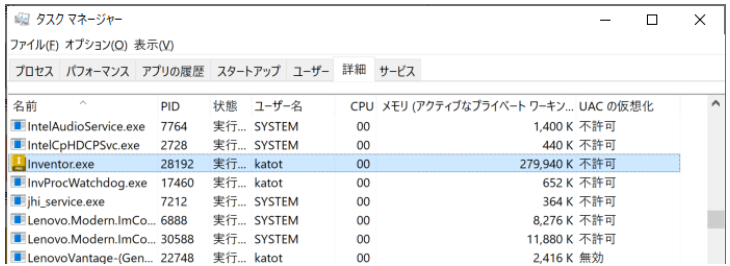
- Inventor APIは64bit版のライブラリのみであるため、APIを呼び出す外部プロセスも64bitである必要がある
  - ✓ 独自アプリの場合は、Visual Studioのビルドオプションで**64bitアプリをビルド**する
  - ✓ Excel のVBAの場合、**64bit版のExcel**を使用する

- GetObjectが失敗する場合…アプリケーションの「UAC仮想化」設定の違いに注意。「UAC仮想化設定」毎にROTが作られるため、異なる「UAC仮想化」設定を持つアプリからは、異なる「UAC仮想化設定」のROTにアクセスできない。

通常ユーザとして実行



管理者として実行




APIを使ったカスタム処理に時間がかかる


トランザクションと画面更新の抑止

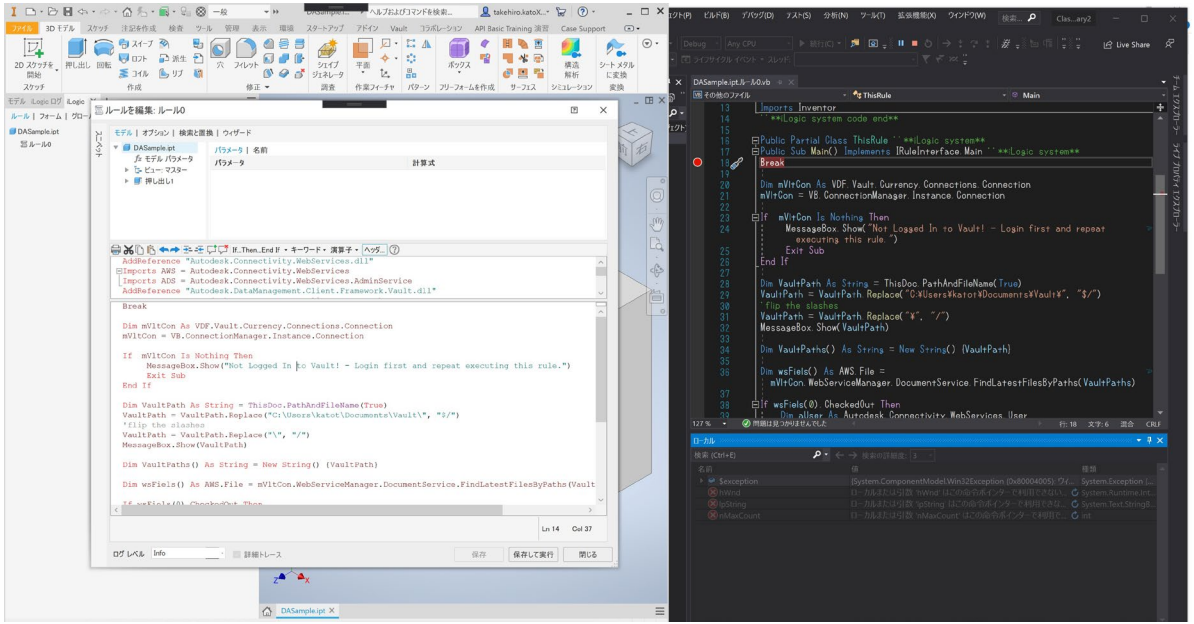
- ThisApplication.TransactionManager. StartTransaction() ~ Transaction.End()

- Application.ScreenUpdatingプロパティ
  - 一連の操作時に画面を更新(再描画)するかどうかを取得および設定
  - 各操作の後に画面を再描画しないように、一連の操作実行中は画面の更新をオフにします
  - 画面を更新するには、必ず画面の更新をオンする


InventorのiLogicでエラーが起きるのですが、デバッグで動作を確認することはできますか？


Visual StudioによるiLogicのデバッグ実行

iLogicのエディタにデバッグ実行機能はありませんが、Visual Studio 2019から、Inventorのプロセスにアタッチして、iLogicをデバッグ実行することは可能です。



- Break Pointの設定
- ステップ実行
- ローカル変数の値の参照

等が利用可能

参考ブログ記事：Visual Studioを利用してInventor iLogicのデバッグ実行する方法