



AutoCAD カスタマイズ&クラウドセミナー

カスタマイズ シナリオと展開

伊勢崎 俊明

Developer Technical Services



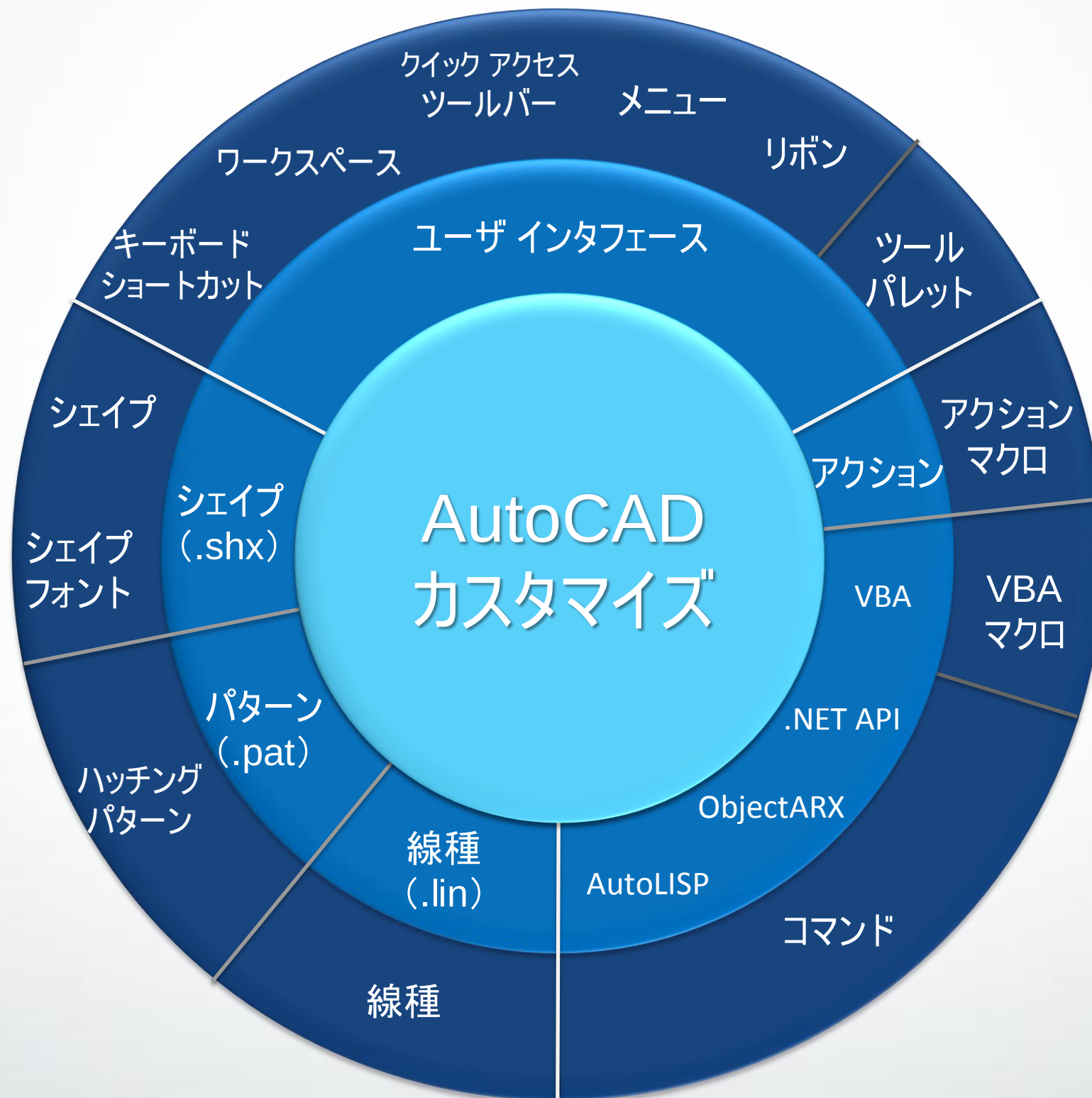
カスタマイズで重要なこと

1. 何をカスタマイズしたいかを明確にする
2. どうカスタマイズすればいいか学習する
3. カスタマイズがどこに保存、維持されているか知る

作図環境？：まず日頃の作図作業のおさらい

- 図面の作図や編集、印刷などの指示は **コマンド** でおこないます
 - 普通、コマンドは **ユーザ インタフェース** のボタンから起動します
 - コマンドは AutoCAD のさまざまな設定値を参照しています
- 設定値は **システム変数** に記憶されています
 - システム変数は各種設定ダイアログやボタンで変更できます
 - システム変数は図面ファイルやレジストリに保存されたりします
- コマンドやシステム変数には **定義ファイル** を参照するものがあります
 - 例) ハッチングはパターン ファイル(.pat)を参照しています
 - 例) 新規図面の作成に図面テンプレート(.dwt)を参照しています
- 定義ファイルは AutoCAD と一緒にインストールされます
 - さまざまな種類のファイルがいろいろな **フォルダ(場所)** にインストールされています

AutoCAD でカスタマイズ可能な領域

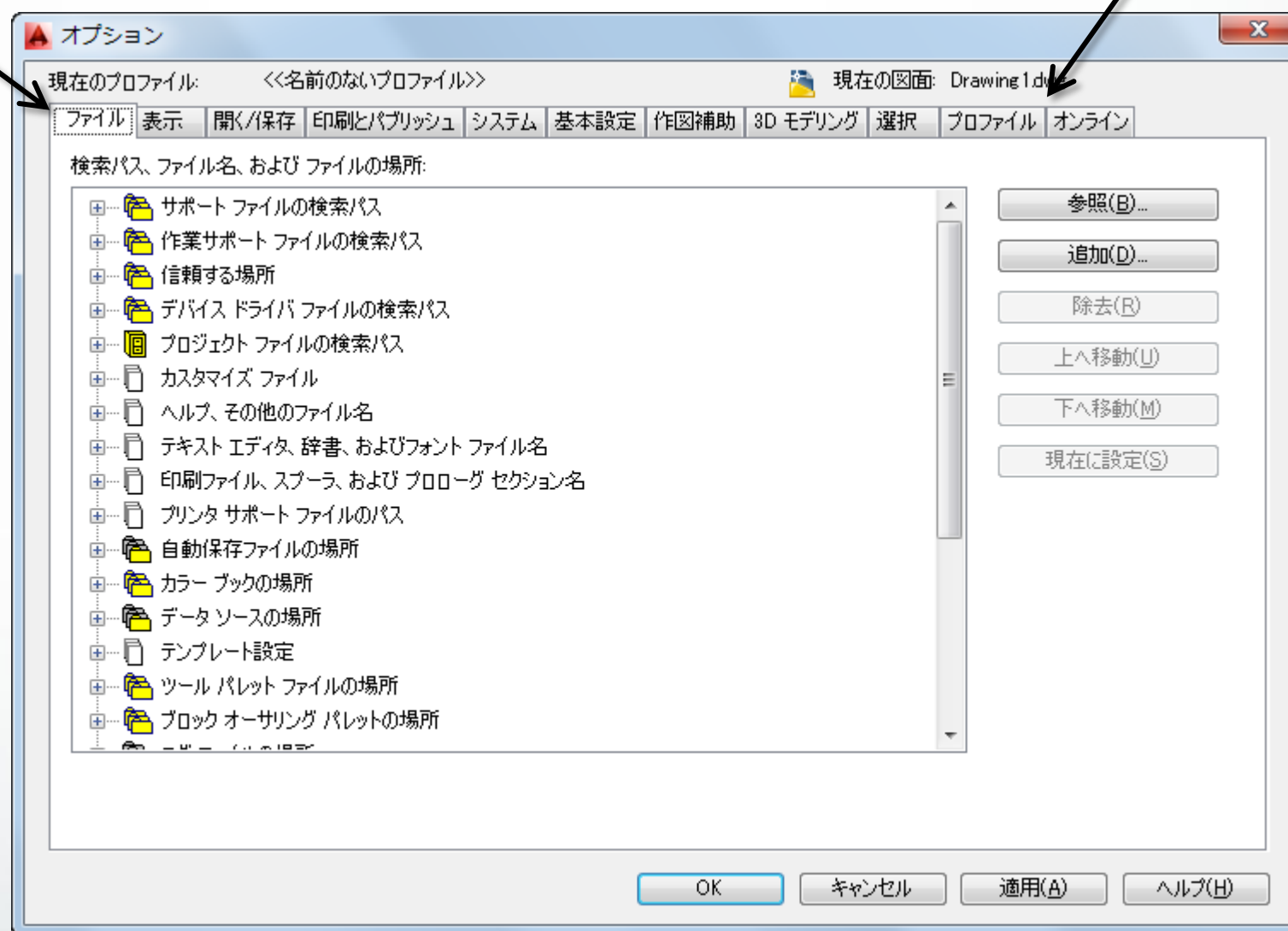


[オプション] ダイアログはカスタマイズの診断カルテ

- カスタマイズ関連ファイルの保管場所を把握することが可能です

[ファイル] タブ

[プロファイルファイル] タブ



コマンド

- AutoCAD の標準コマンド

- ZOOM、PAN、LINE、CIRCLE、PLOT、SAVEAS .コマンド など..
- オンライン ヘルプに使用方法が説明されています

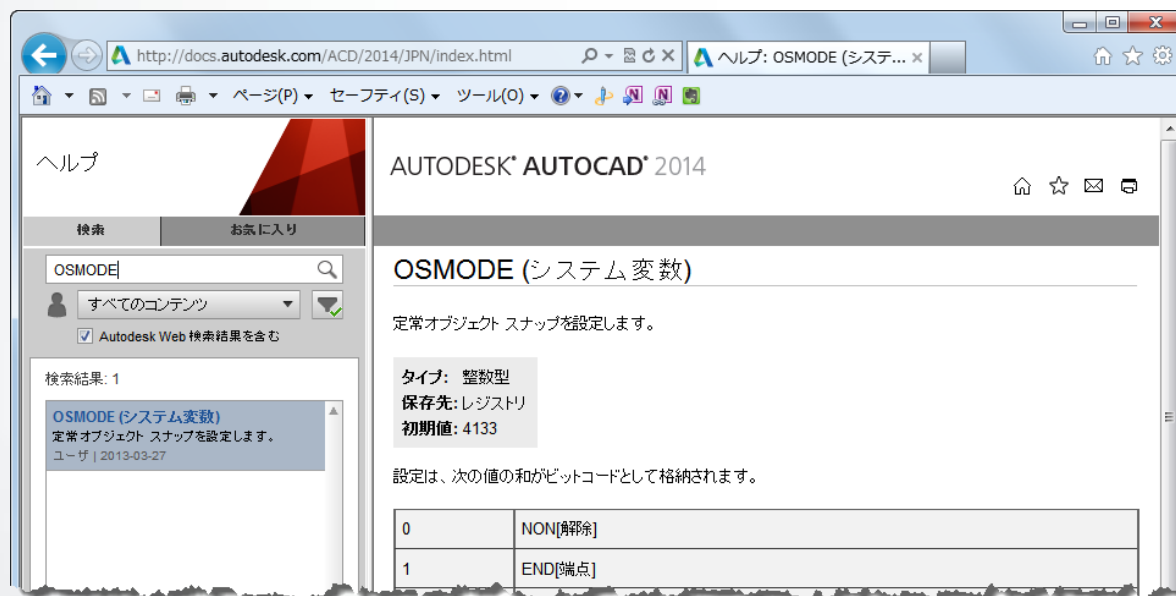


- 独自のコマンドを定義することも可能

- AutoLISP、ObjectARX、.NET API のプログラムで定義
- アクション レコーダーでコマンドを自動記録で定義

システム変数

- AutoCAD のシステム変数
 - OSMODE、SNAPMODE、CLAYER、CECOLOR コマンド など..
 - オンライン ヘルプに使用方法が説明されています

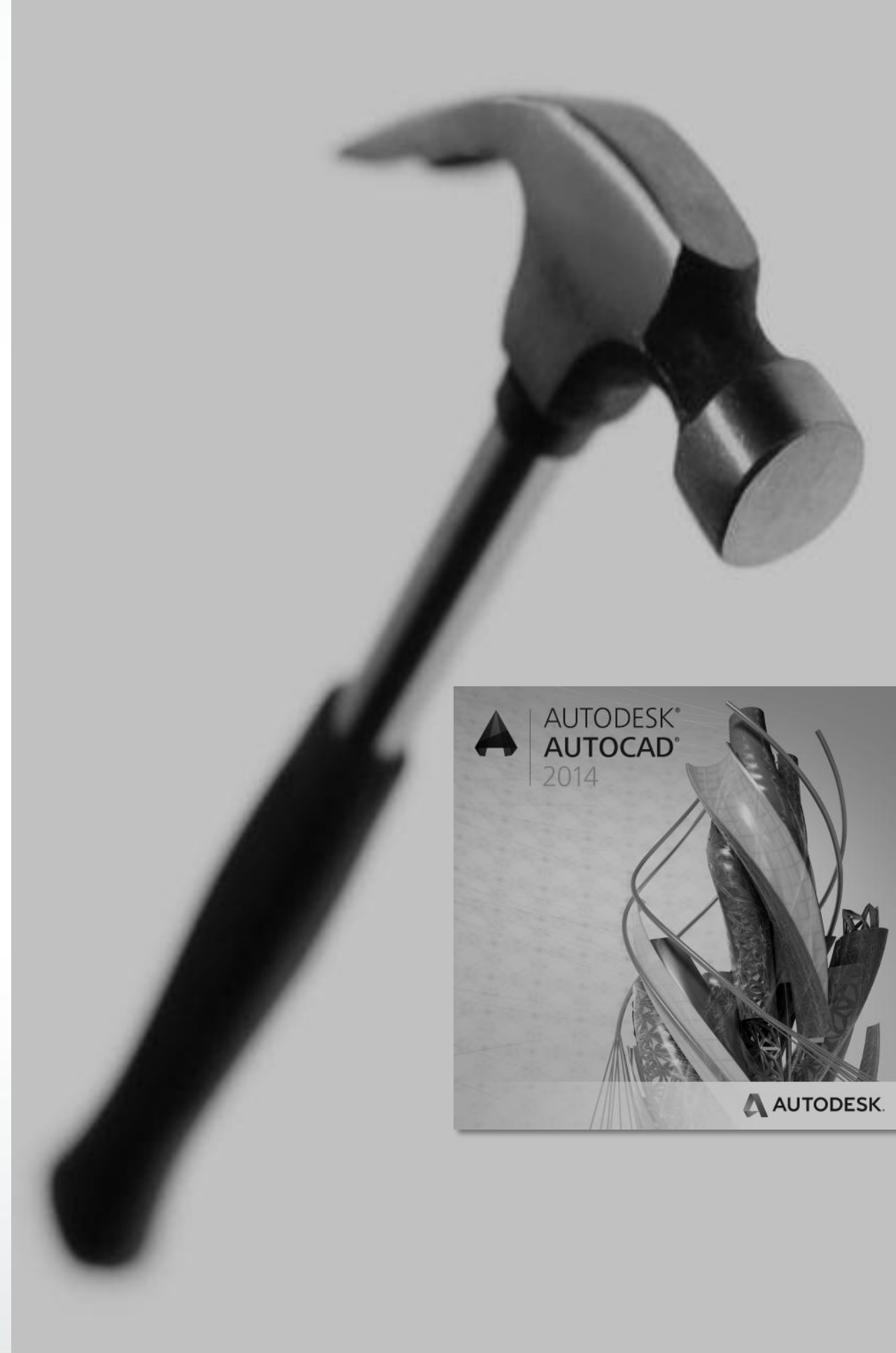


- システム変数によって保存される場所が異なります
 - 図面ファイル、システム レジストリ、保存場所なし
- 独自のシステム変数を定義することも可能
 - ObjectARX、.NET API のプログラムで定義

一般的なカスタマイズのシナリオ

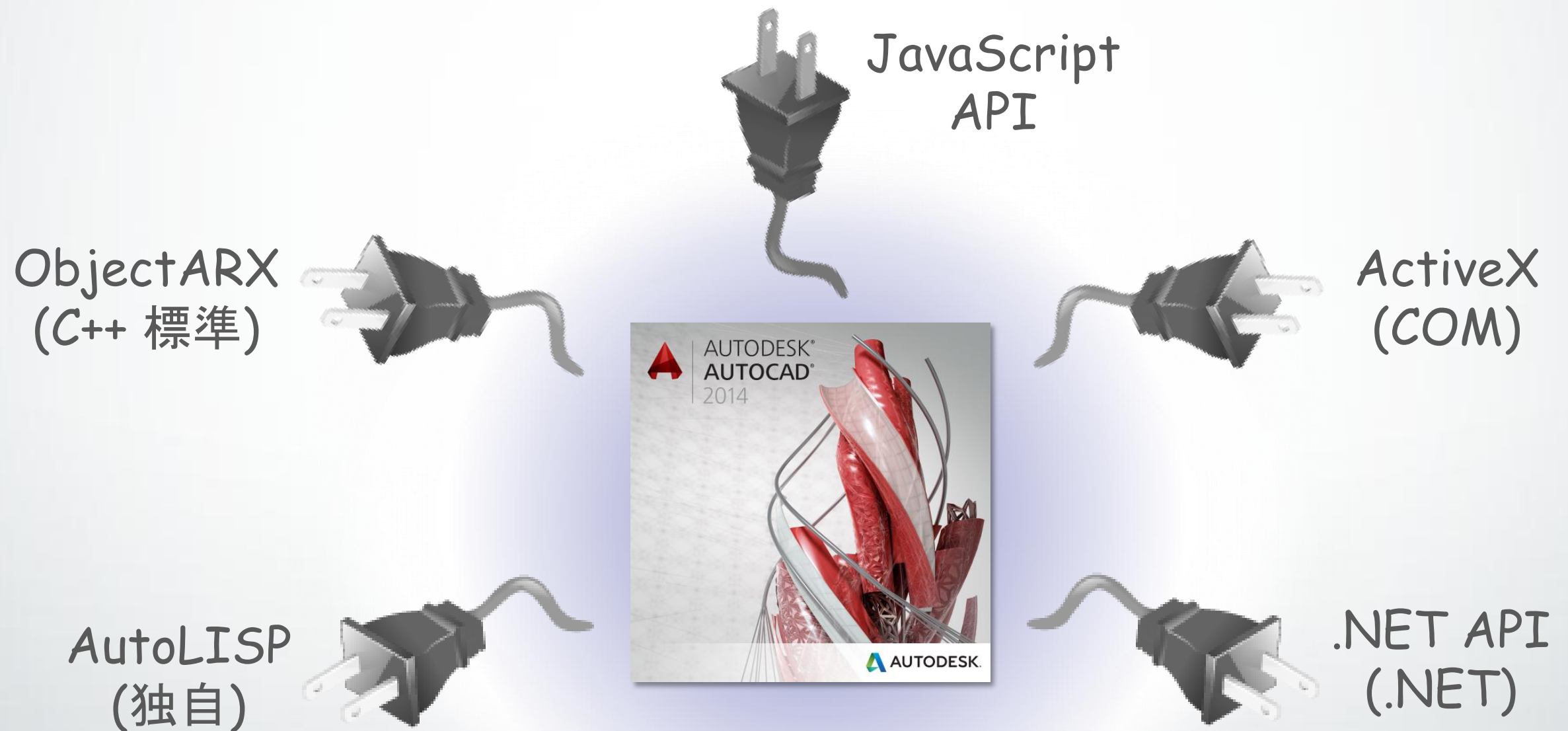
- コマンドの作成
 - 独自の作図コマンドを作成
 - 設定値の切り替えをコマンド化
- ユーザ インタフェースの登録
 - よく利用するコマンドをボタンとしてリボンに登録
 - よく利用する旧タイプのコマンドをリボンに登録
- カスタマイズした定義ファイルの AutoCAD へのロード
 - コマンドを定義した API モジュール
 - ユーザ インタフェース
- カスタマイズした定義ファイルの他コンピュータへの展開
 - 社内設計者への効率的な配布（インストール）

コマンドの作成



AutoCAD API

- 柔軟で強力な Application Programming Interface



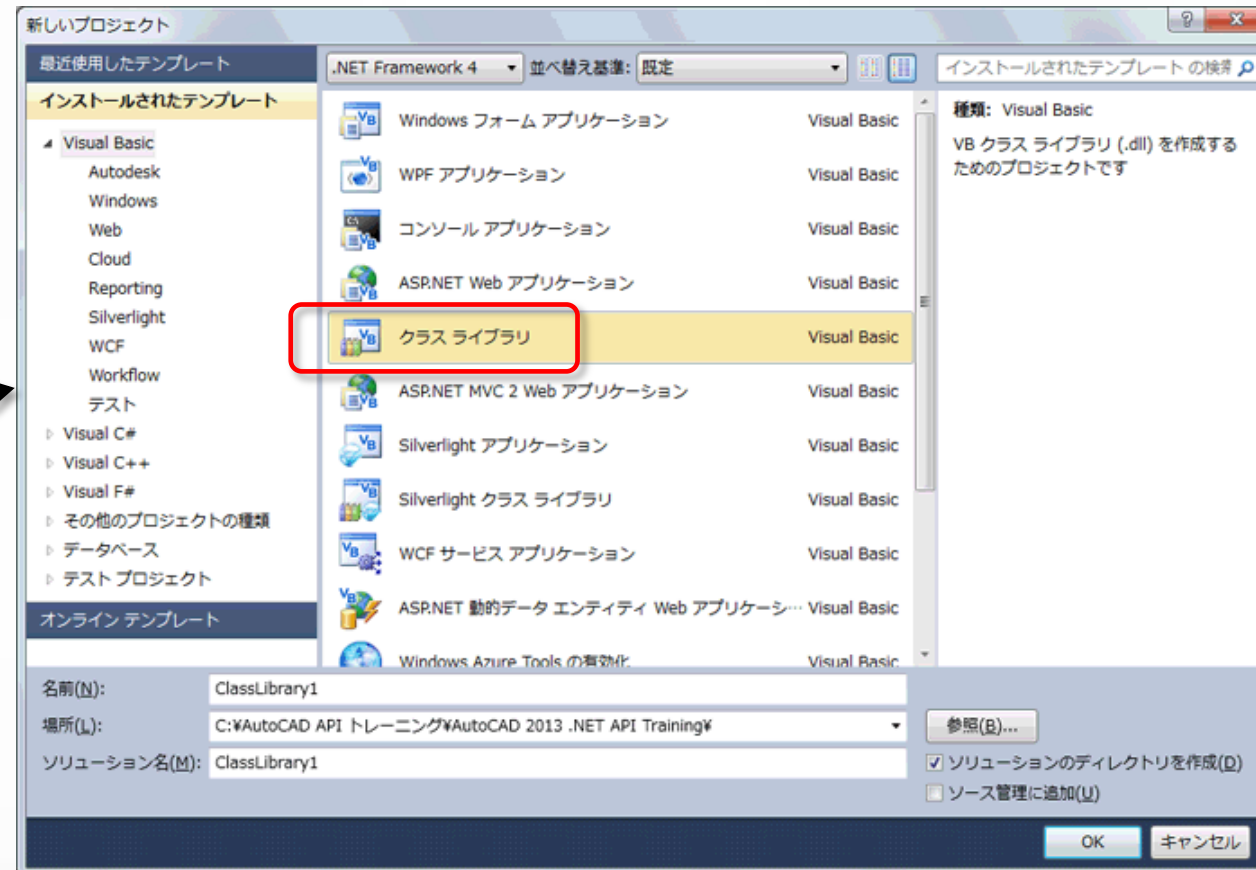
AutoCAD .NET アプリケーション作成の用意

- AutoCAD 2014
 - アセンブリ ファイルで API を公開
 - [accoremgd.dll](#)、[acdbmgd.dll](#)、[acmgd.dll](#)
 - .NET アプリケーションは AutoCAD にロードして実行
 - NETLOAD コマンド
- Visual Studio 2010 SP1
 - .NET Framework 4.0 をサポートする Visual Studio
 - AutoCAD .NET API アセンブリを参照
 - ビルド(コンパイル)してバイナリ ファイル(DLL ファイル)を作成
- ObjectARX SDK
 - AutoCAD .NET API に関するドキュメント(英語)とサンプルを提供
 - 本来は ObjectARX API 用の Software Development Kit

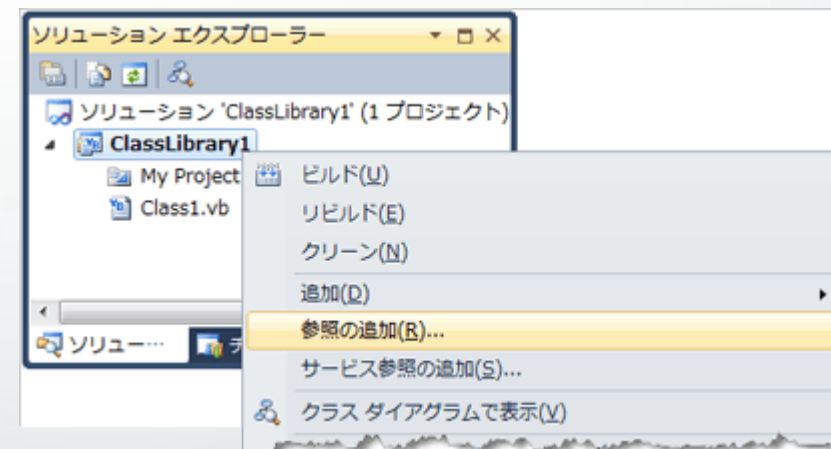
プロジェクトとコマンドの作成

1. Visual Studio 2010 で クラス ライブラリ プロジェクト作成

VB プロジェクトの場合

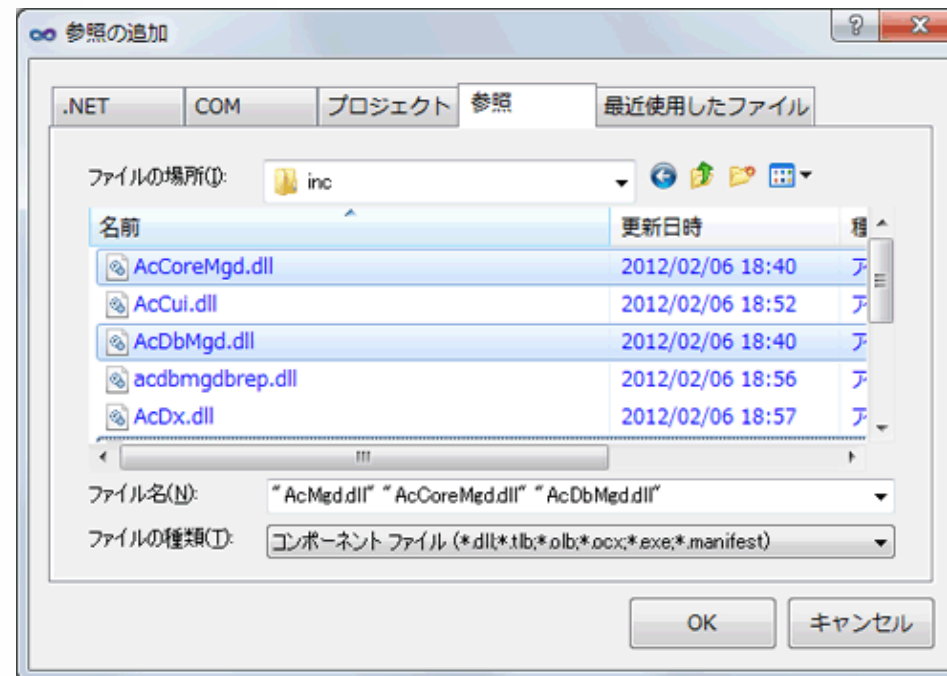


2. ソリューション エクスプローラから “参照の追加” を選択



プロジェクトとコマンドの作成

3. acmgdcore.dll、acdbmgd.dll と acmgd.dll を選択



4. グローバル スコープに名前空間を追加してクラスにコマンドを追加

```
Imports Autodesk.AutoCAD.ApplicationServices ← 名前空間のインポート
Imports Autodesk.AutoCAD.Runtime
Public Class Class1
    <CommandMethod("MyCLines") <— コマンド定義記述とコマンド関数
    Public Sub MyCLines()
        :
    End Sub
End Class
```

プロジェクトとコマンドの作成

5. 選択位置から交差中心線を作図

```
Dim oDb As Database = HostApplicationServices.WorkingDatabase
Dim oTr As Transaction = oDb.TransactionManager.StartTransaction
Dim oEd As Editor = Application.DocumentManager.MdiActiveDocument.Editor

Try
    Dim oRst As PromptPointResult = oEd.GetPoint(vbCrLf & "作図中心点を指示:")
    If oRst.Status <> PromptStatus.OK Then
        Throw New Autodesk.AutoCAD.Runtime.Exception(ErrorStatus.InvalidInput)
    End If

    Dim oPt As Point3d = oRst.Value

    Dim oLt As LinetypeTable = oTr.GetObject(oDb.LinetypeTableId, OpenMode.ForWrite)
    Dim oLtrId As ObjectId
    If Not oLt.Has("CENTER") Then
        oDb.LoadLineTypeFile("CENTER", "acadiso.lin")
    End If

    Dim oLtr As LinetypeTableRecord = oTr.GetObject(oLt("CENTER"), OpenMode.ForRead)
    oLtrId = oLtr.ObjectId
```

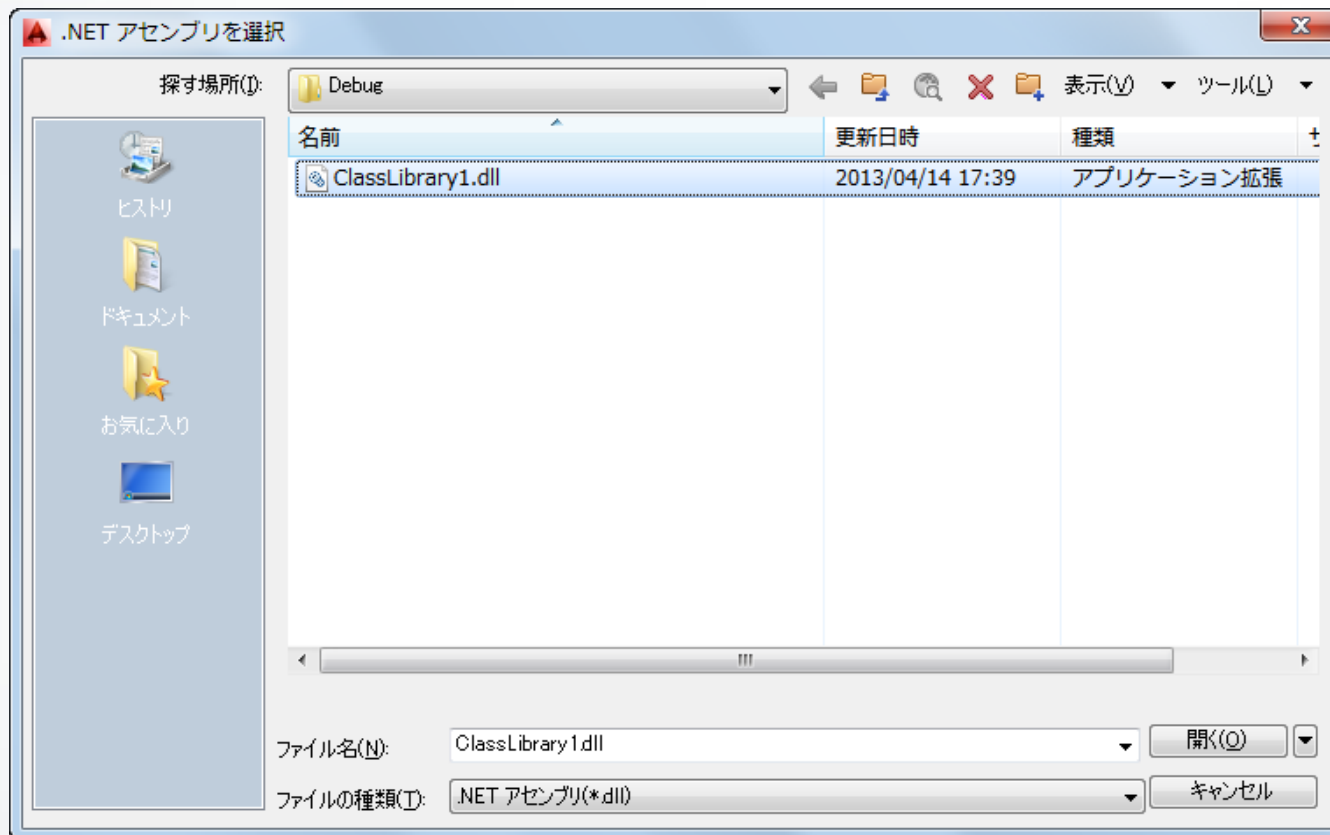
プロジェクトとコマンドの作成

5. 選択位置から交差中心線を作図(続き)

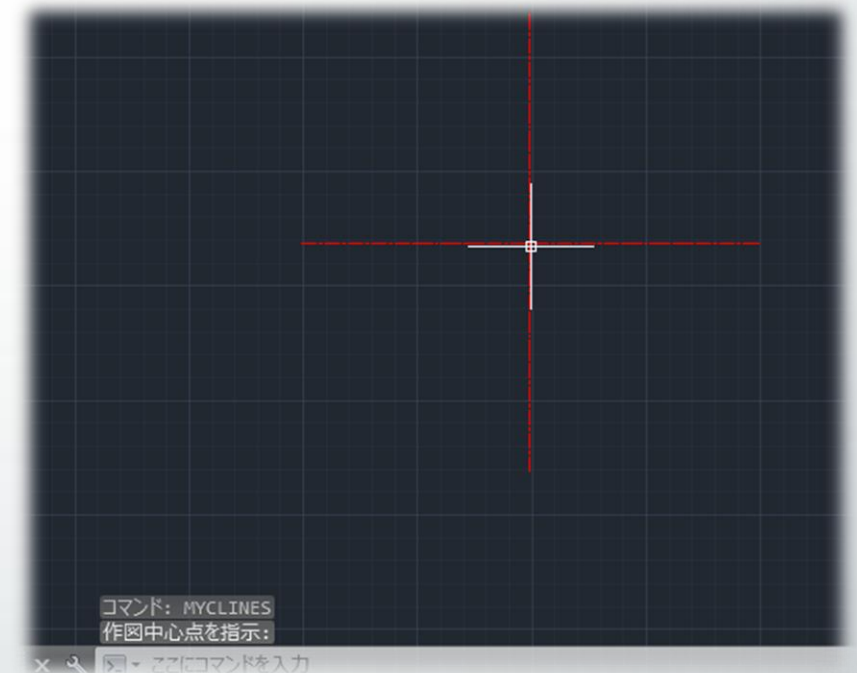
```
Dim oHLine As Line = New Line(New Point3d(oPt.X - 100.0, oPt.Y, 0.0), _  
                               New Point3d(oPt.X + 100.0, oPt.Y, 0.0))  
Dim oVLine As Line = New Line(New Point3d(oPt.X, oPt.Y - 100.0, 0.0), _  
                               New Point3d(oPt.X, oPt.Y + 100.0, 0.0))  
  
oHLine.LinetypeId = oLtrId  
oVLine.LinetypeId = oLtrId  
oHLine.ColorIndex = 1  
oVLine.ColorIndex = 1  
  
Dim oBt As BlockTable = oTr.GetObject(oDb.BlockTableId, OpenMode.ForRead)  
Dim oBtr As BlockTableRecord = oTr.GetObject(oBt("*MODEL_SPACE"), OpenMode.ForWrite)  
oBtr.AppendEntity(oHLine)  
oBtr.AppendEntity(oVLine)  
oTr.AddNewlyCreatedDBObject(oHLine, True)  
oTr.AddNewlyCreatedDBObject(oVLine, True)  
oTr.Commit()  
Catch oEx As Exception  
    MsgBox(oEx.ToString())  
Finally  
    oTr.Dispose()  
End Try
```

プロジェクトとコマンドの作成

5. コンパイル後に NETLOAD コマンドでアセンブリをロード



6. MYCLINES コマンドを実行



アドオン アプリケーションの提供形式

- API と用途別に異なる実行モジュール

- AutoLISP: **.lsp** ファイル 、**.fas** ファイル 、**.vlx** ファイル 

- COM (VBA): **.dvb** ファイル 

- ObjectARX: **.arx** ファイル 、**.dbx** ファイル 

- .NET API: **.dll** ファイル 

- 実行にはモジュールを **AutoCAD** にロード

- ロード後に定義した機能が利用可能に

AutoCAD

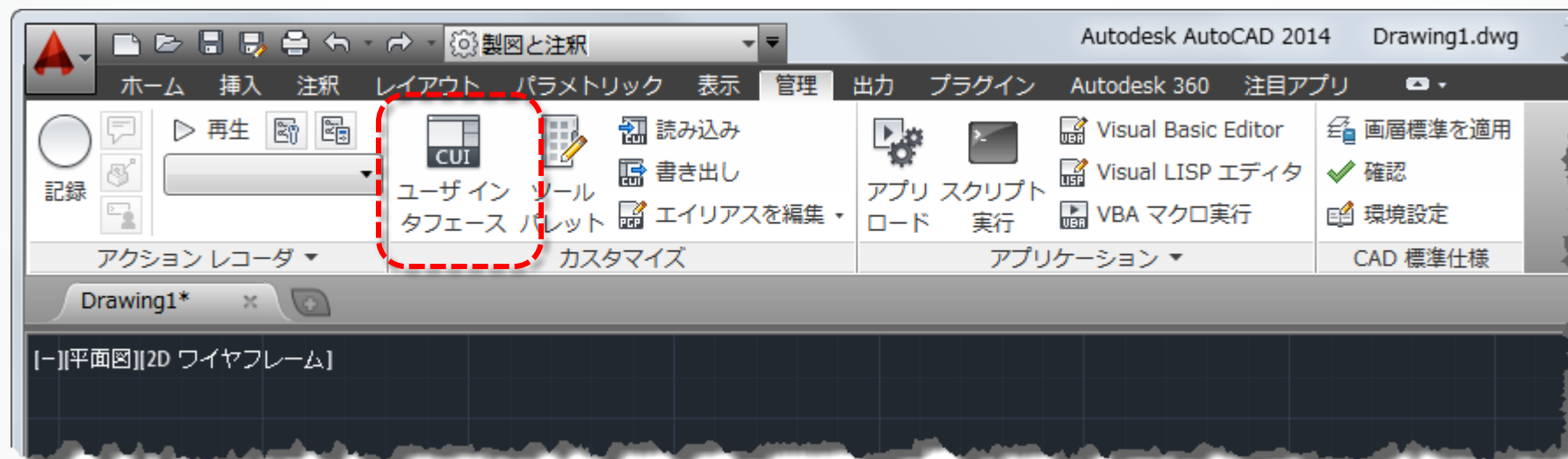
ユーザ インタフェースの登録



[ユーザ インタフェースをカスタマイズ] コマンド

- CUI コマンド

- ダイアログ上の操作でリボン インタフェースを定義



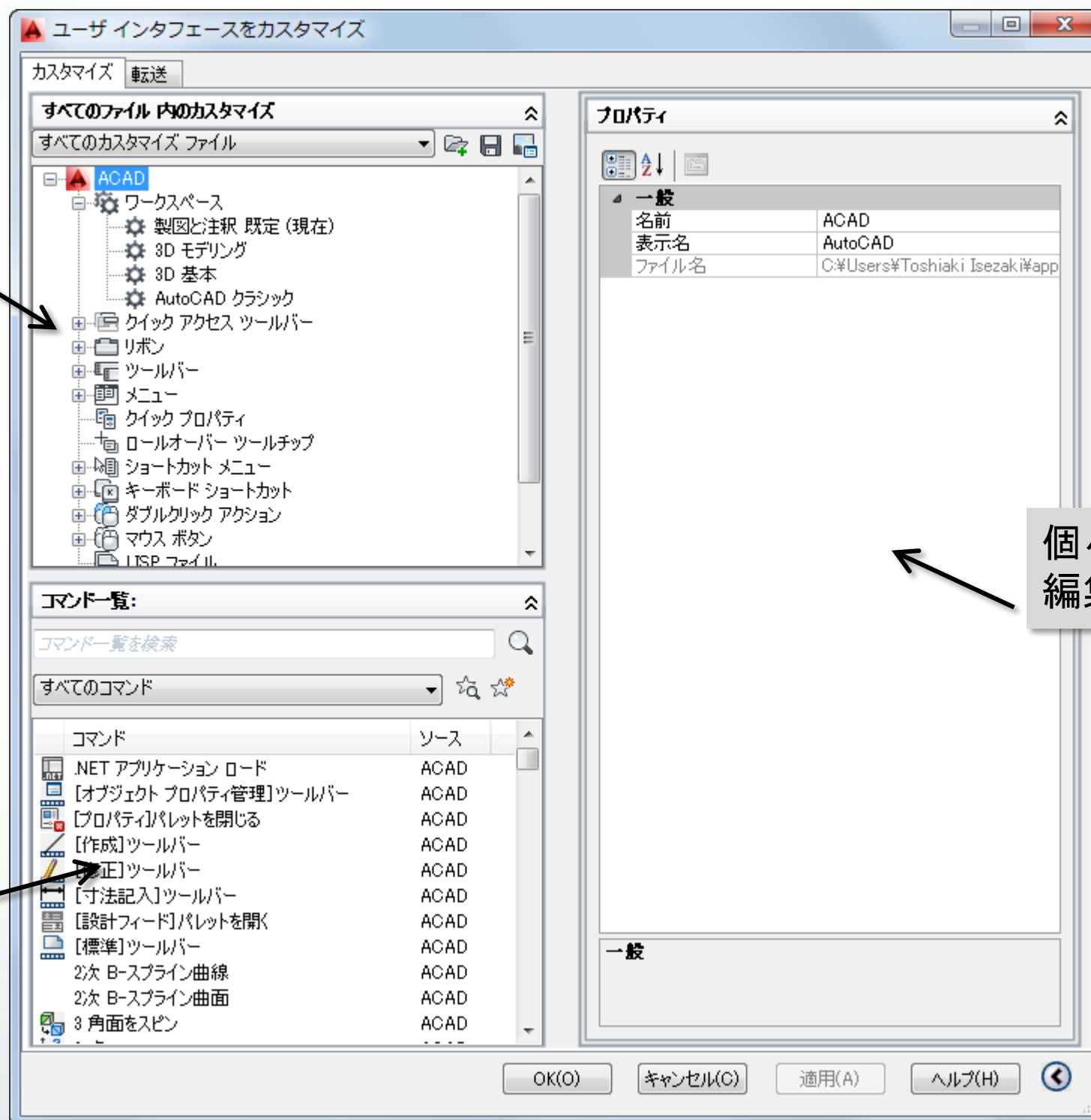
- リボン上のボタンのカスタマイズが可能

- アイコン
 - 実行されるコマンド
 - ツールチップ内容
 - など

[ユーザ インタフェースをカスタマイズ] ダイアログ

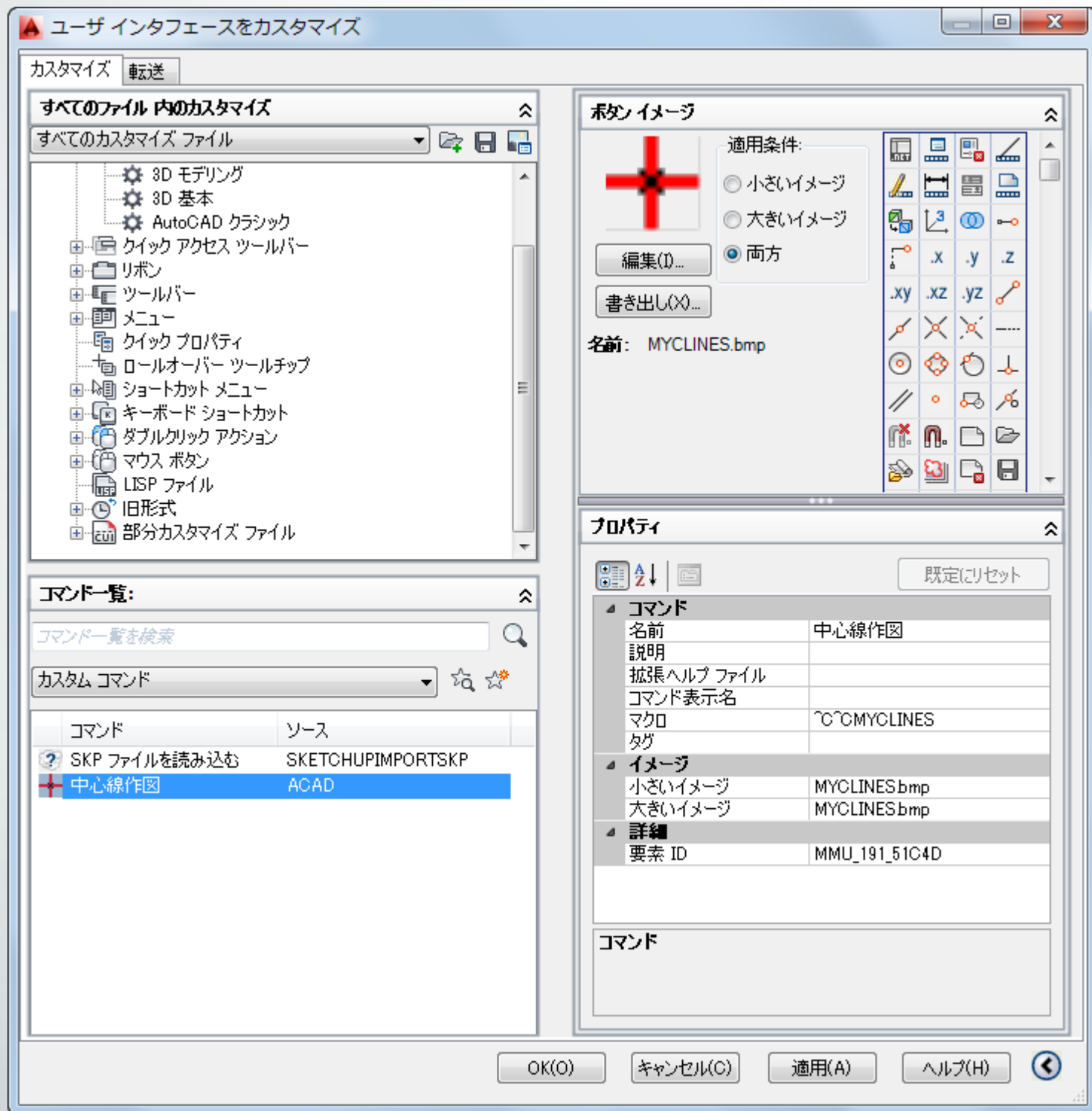
実際にカスタマイズボタンを配置するインタフェースを選択するエリア

カスタマイズボタンを選択、作成するエリア



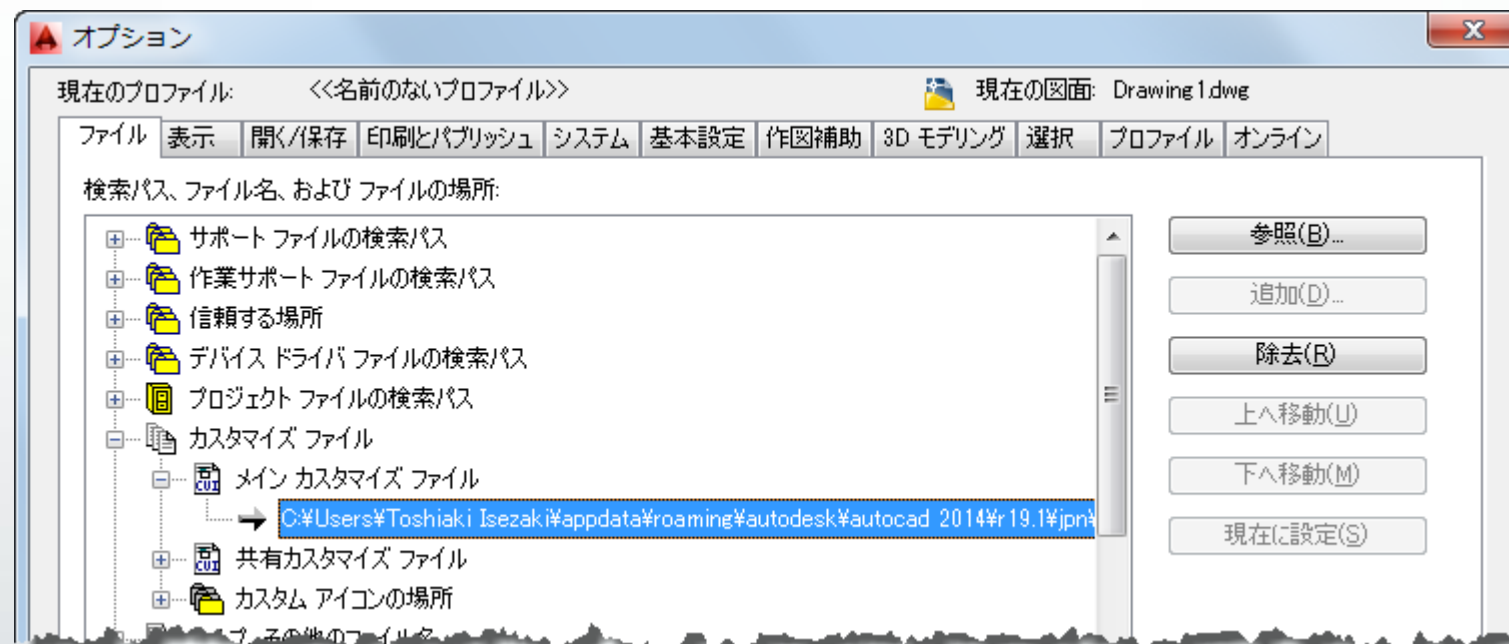
個々のボタンの詳細を編集するエリア

[ユーザ インタフェースをカスタマイズ] ダイアログ



ユーザ インタフェースの定義ファイル

- 2 つのカスタマイズ ファイル(.cuix ファイル)
 - メイン カスタマイズ ファイル
 - acad.cuix = 全メニュー項目を含む AutoCAD 標準メニュー
 - 部分 カスタマイズ ファイル
 - 独自ファイル名の .cuix ファイル = 必要な一部メニューのみを含むメニュー
- 標準のメインカスタマイズ ファイルの場所
 - ローミング フォルダ下に保存されている

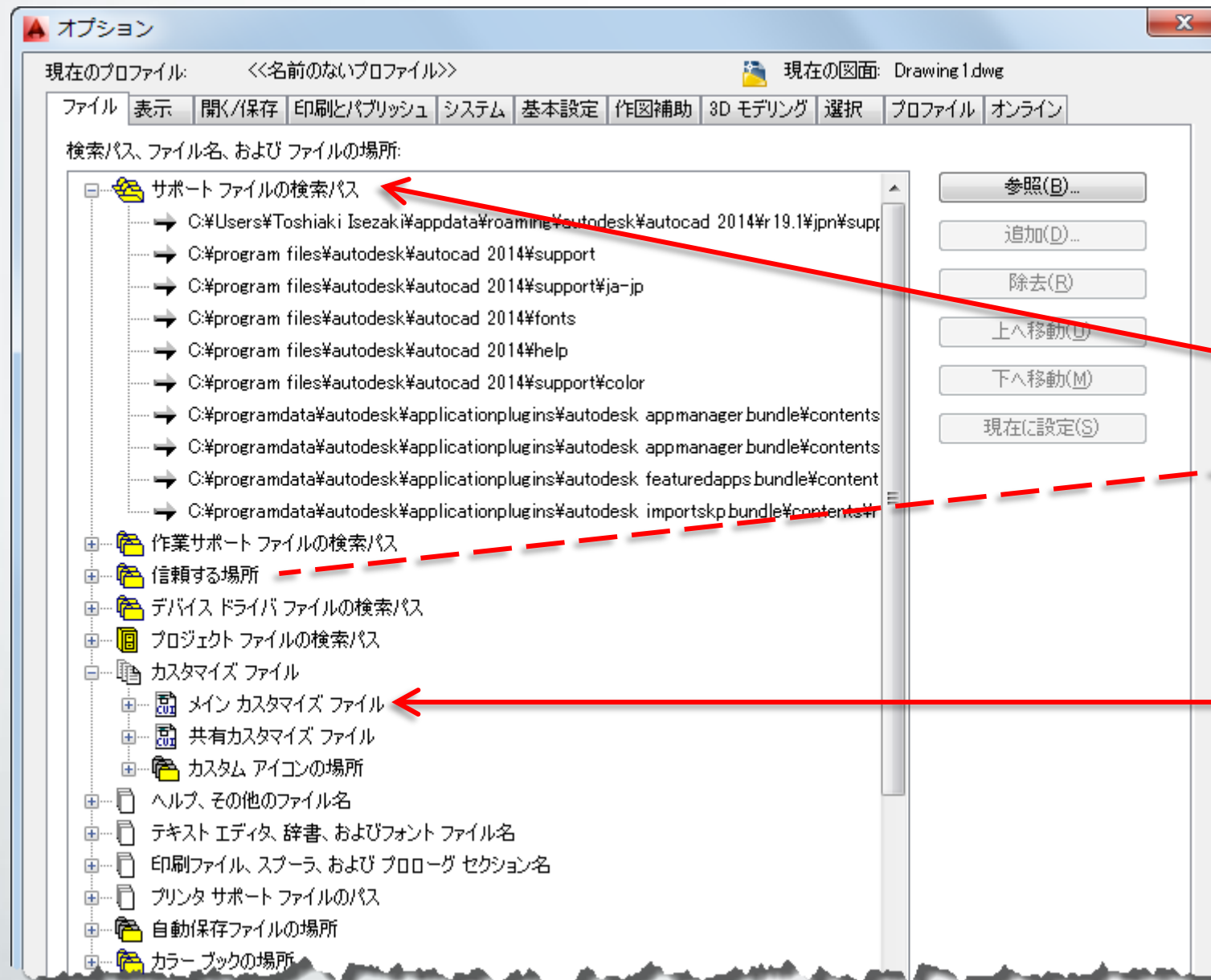


定義ファイルのロード



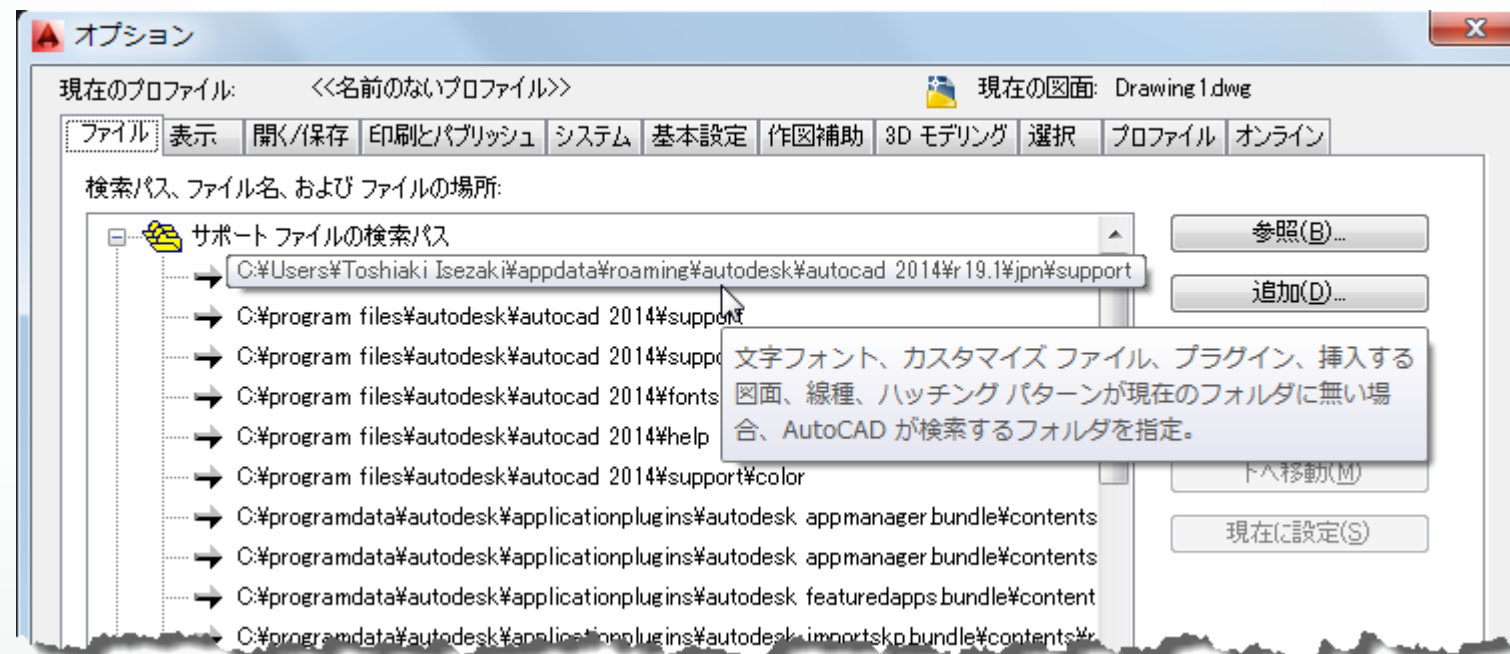
カスタマイズしたファイルの定義ファイルはどこに？

- [オプション] ダイアログに設定されているパスを調査



カスタマイズしたファイルのありか

- ログインユーザ別への参照フォルダ移動
 - ローミングと呼ばれる機能
- AutoCAD は自動でローミング フォルダを認識



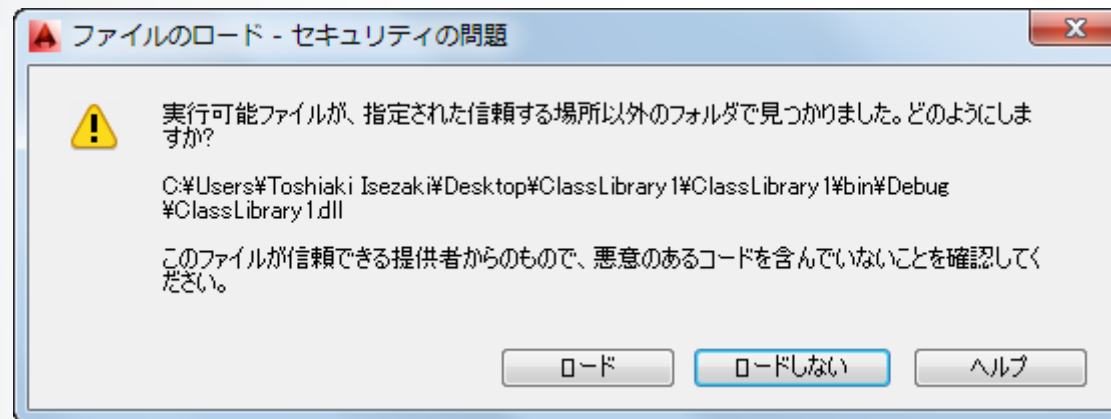
- システム変数が現在の情報を提供
 - ROAMABLEROOTPREFIX システム変数
 - LOCALROOTPREFIX システム変数

.NET アセンブリのロード方法

- NETLOAD コマンド
 - 設計者に毎回手動でロードさせるのは難
 - 自動ロードさせるのがベスト
- 自動ロードの方法
 - 初期化モジュールを使った方法
 - AutoCAD 起動時に自動ロード・実行される acad.lsp の利用
 - デイマンド ロードを使った方法
 - システム レジストリへの登録が必要
 - 管理者権限でのインストールを要求
 - 自動ローダーを使った方法(おすすめ)
 - XML パッケージファイルでのロード

API カスタマイズ ファイルのセキュア ロード

■ API カスタマイズ ファイル ロード時に警告ダイアログ



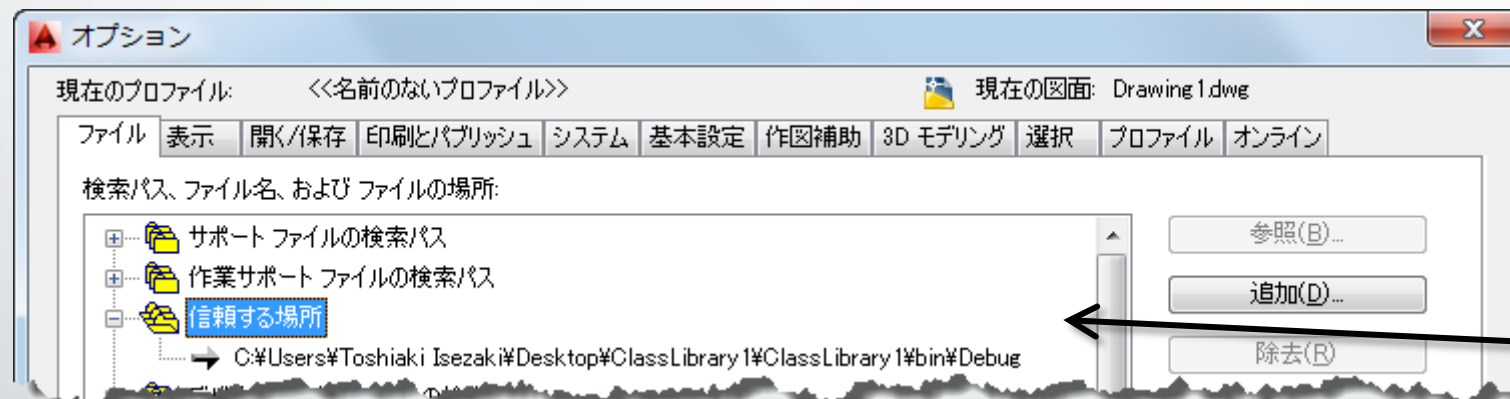
悪意のあるプログラムから AutoCAD 環境を保護するため AutoCAD 2014 から導入

■ 警告ダイアログの表示の抑止方法

■ アドオン アプリケーション用既定フォルダを利用する(おすすめ)

■ *ApplicationPlugins* フォルダ

■ 信頼するフォルダにロード ファイルのパスを指定



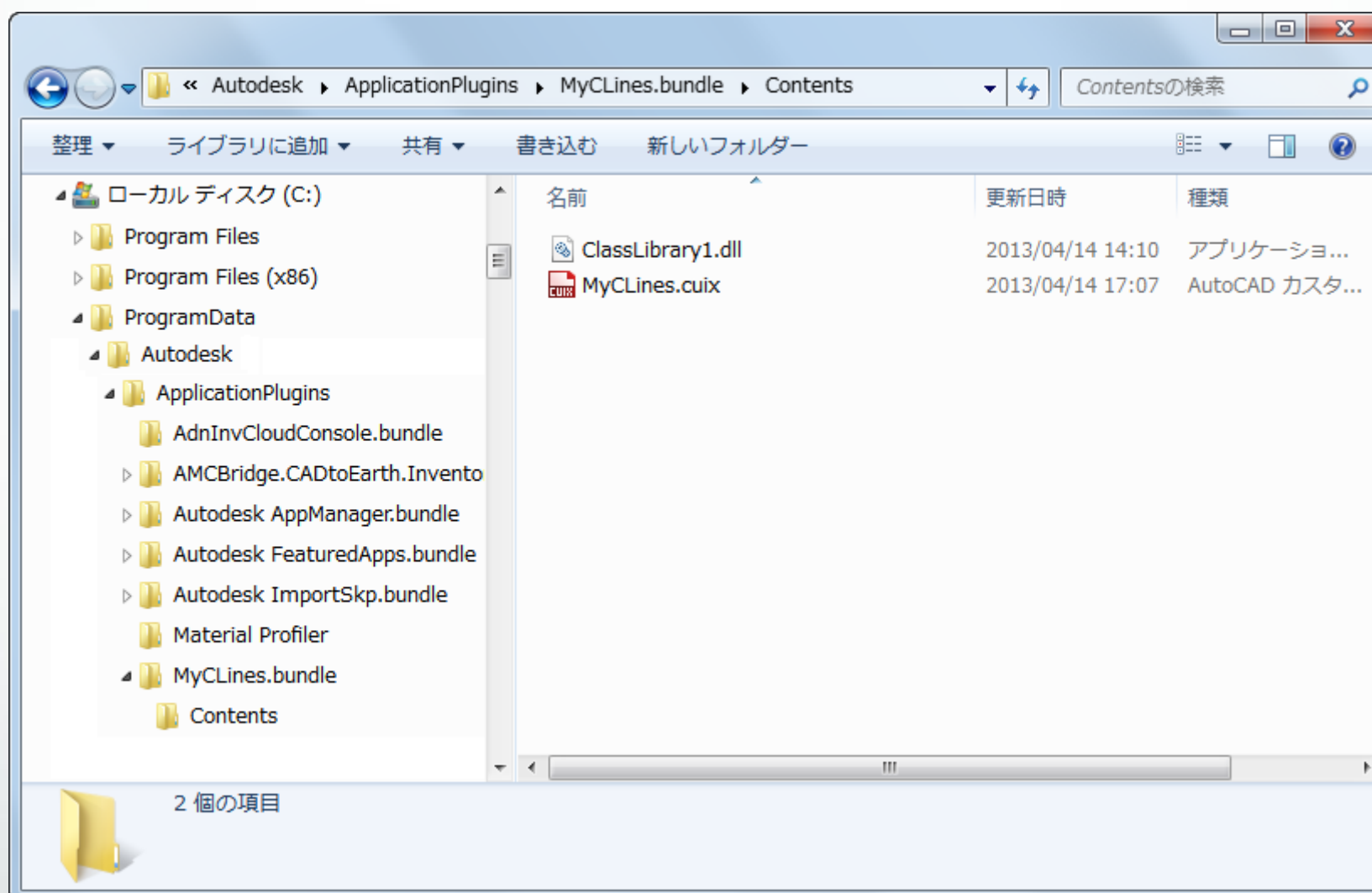
設定は TRUSTEDPATHS システム変数でも指定可能(複数パス可能)

自動ローダーの利点

- .cuix と API モジュールの一括ロードが可能
 - Autodesk Exchange Apps のロード機構
 - 購入&ダウンロードですぐに利用できるようにするため
- 準備
 - API カスタマイズ ファイル(.NET API のアセンブリ ファイル)
 - 部分カスタマイズ ファイルの用意
 - カスタム コマンドのボタンを配置したリボン パネル
 - 作成したリボン パネルをプラグイン タブに配置
 - PackageContents.xml ファイル
 - アプリケーション情報やファイルのありかを記した XML ファイル

自動ローダーを利用したカスタマイズの配置と構造

- C:\ProgramData\Autodesk\ApplicationPlugins フォルダ
 - *XXXXXX.bundle* フォルダ配下に各ファイルを配置



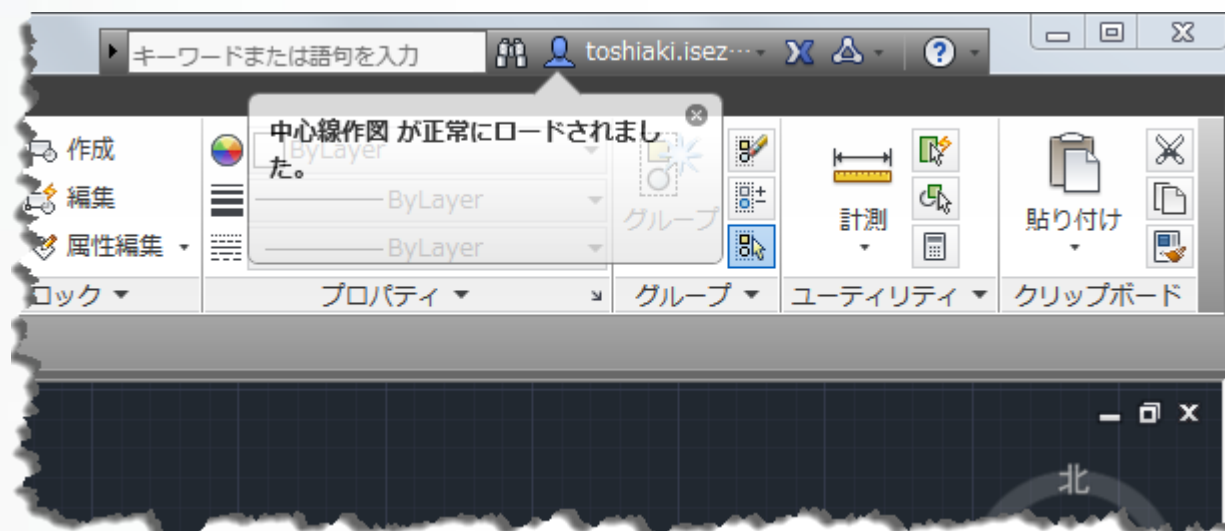
PackageContents.xml ファイルの記述

■ アプリケーション パッケージ

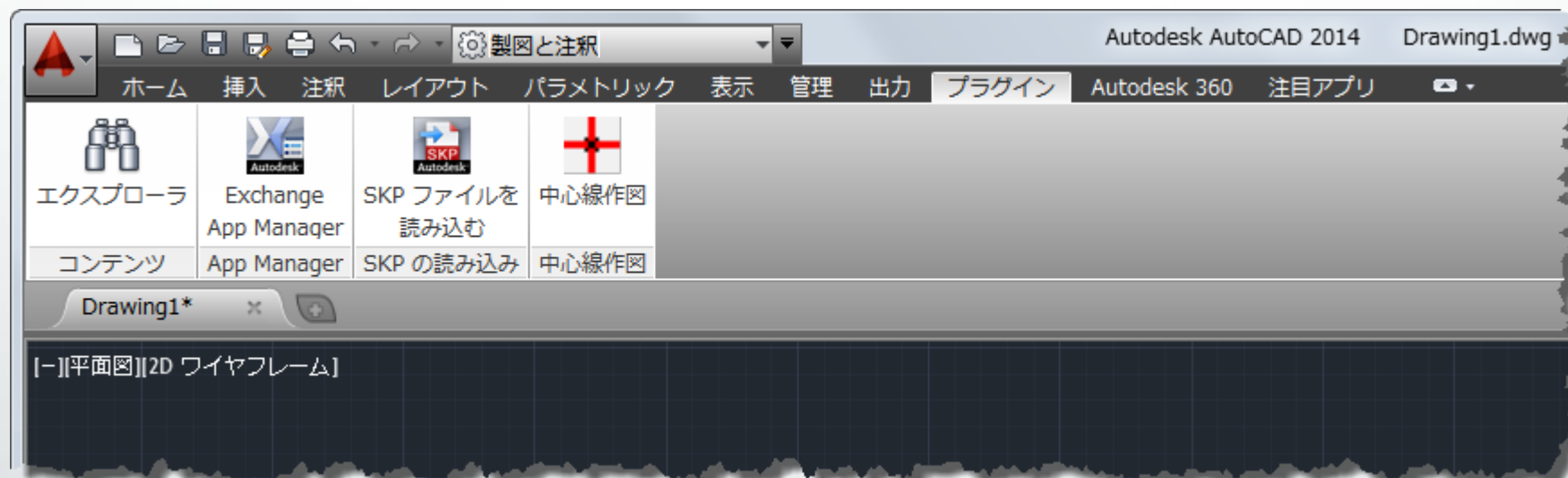
```
<?xml version="1.0" encoding="utf-8" ?>
<ApplicationPackage
  SchemaVersion="1.0"
  AppVersion="1.0"
  Author="オートデスク株式会社"
  ProductCode="{0008A469-F169-4A5F-ACCA-201996F94D60}"
  Name="中心線作図">
  <CompanyDetails
    Name="オートデスク株式会社"
    Url="http://adndevblog.typepad.com/technology_perspective"
  />
  <Components>
    <ComponentEntry
      AppName="ClassLibrary1"
      ModuleName="./Contents/ClassLibrary1.dll"
    >
  </ComponentEntry>
    <ComponentEntry
      AppName="ClassLibrary1"
      ModuleName="./Contents/MyCLines.cuix"
    >
  </ComponentEntry>
  </Components>
</ApplicationPackage>
```

自動ローダーを使ったカスタマイズ環境の再生

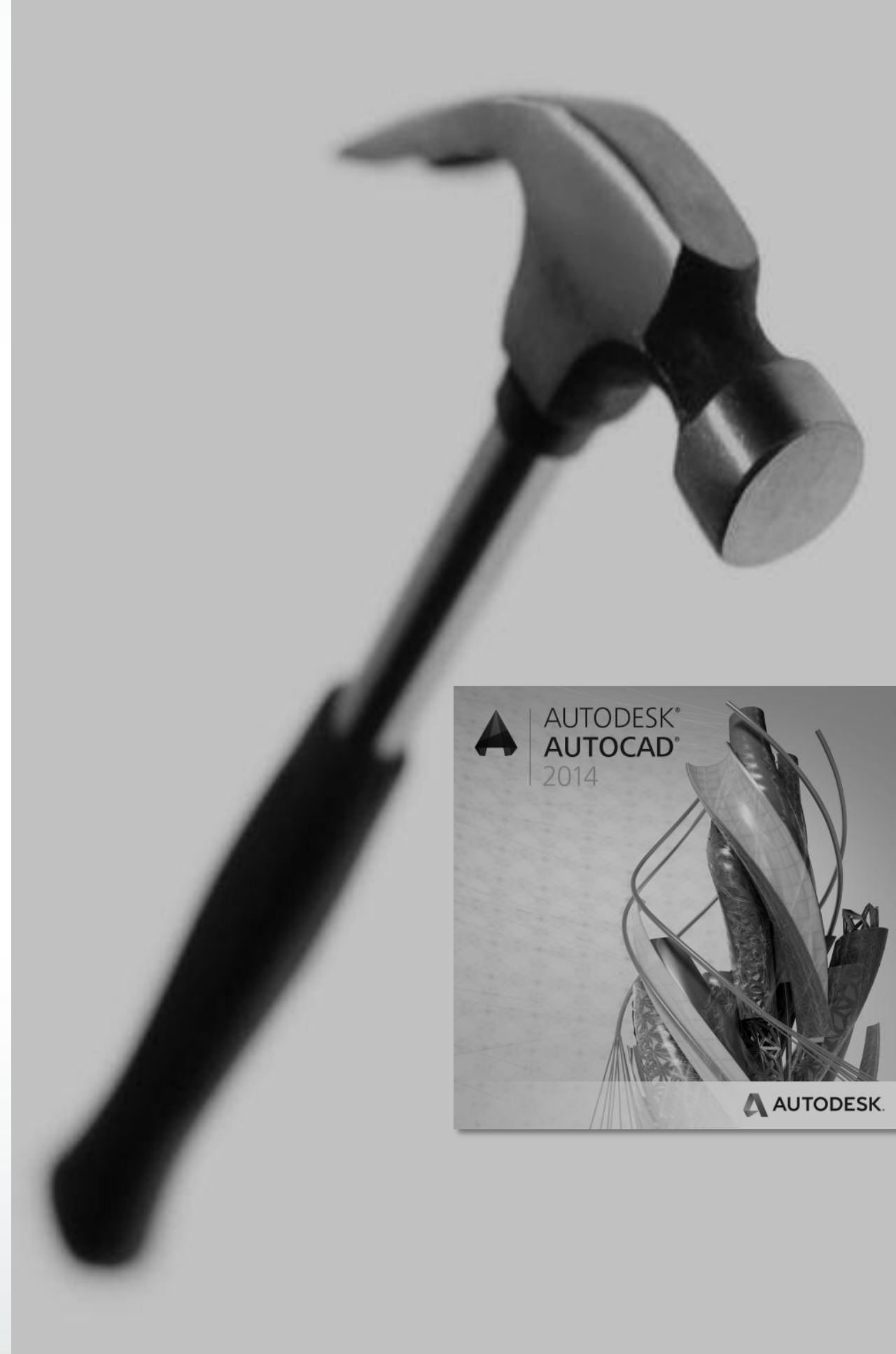
- 初回ロード時にメッセージを表示



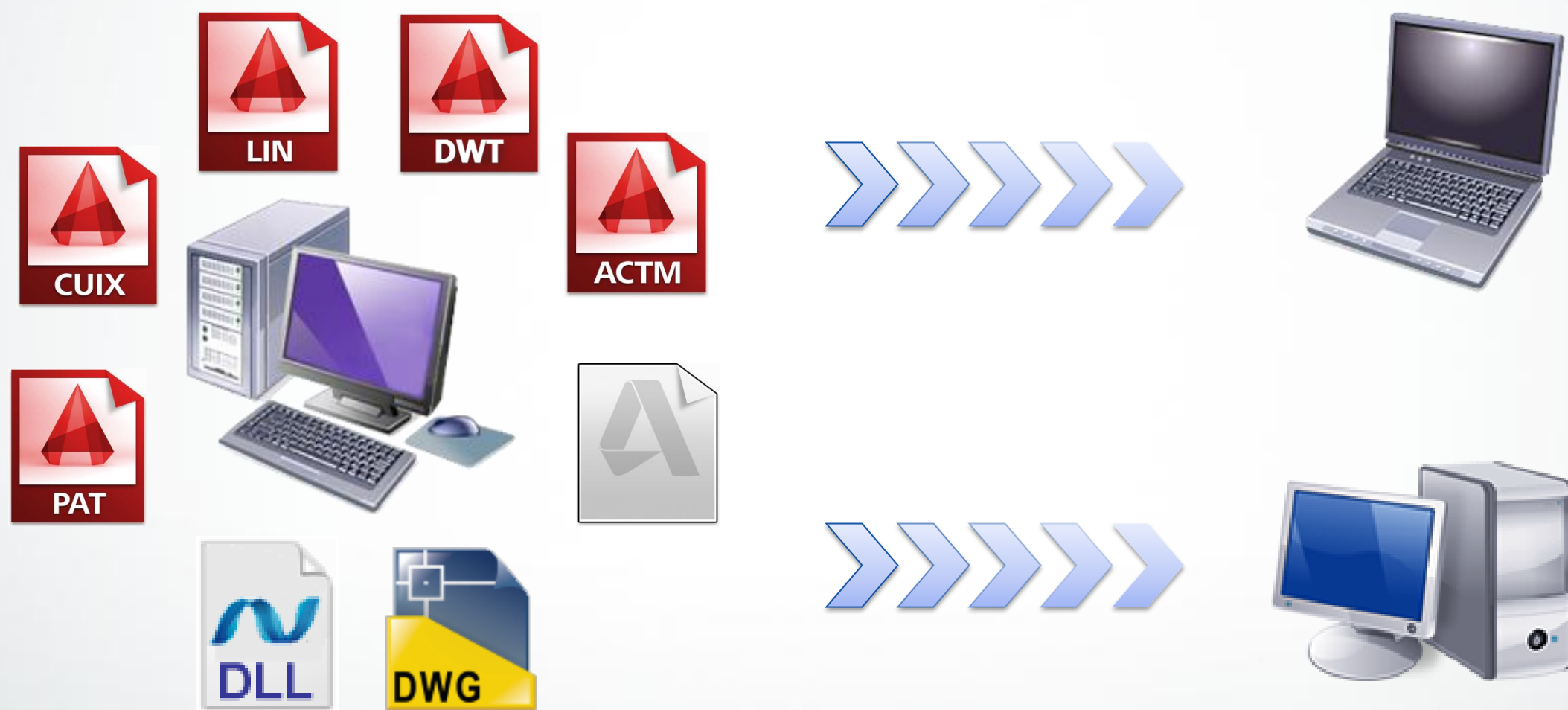
- [プラグイン] リボン タブに部分メニューの情報を表示



定義ファイルの展開



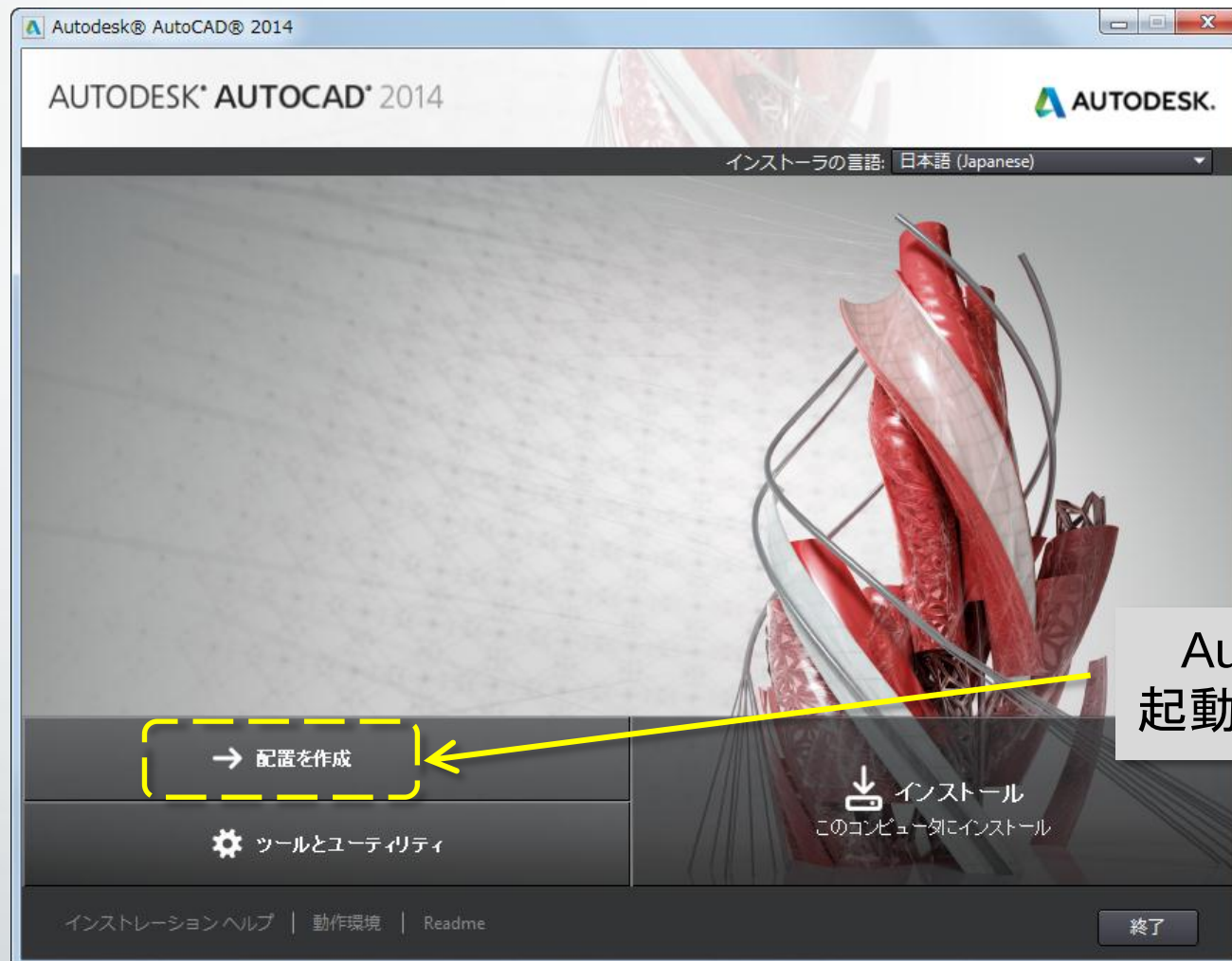
別のコンピュータへの作図環境全体の移植は？



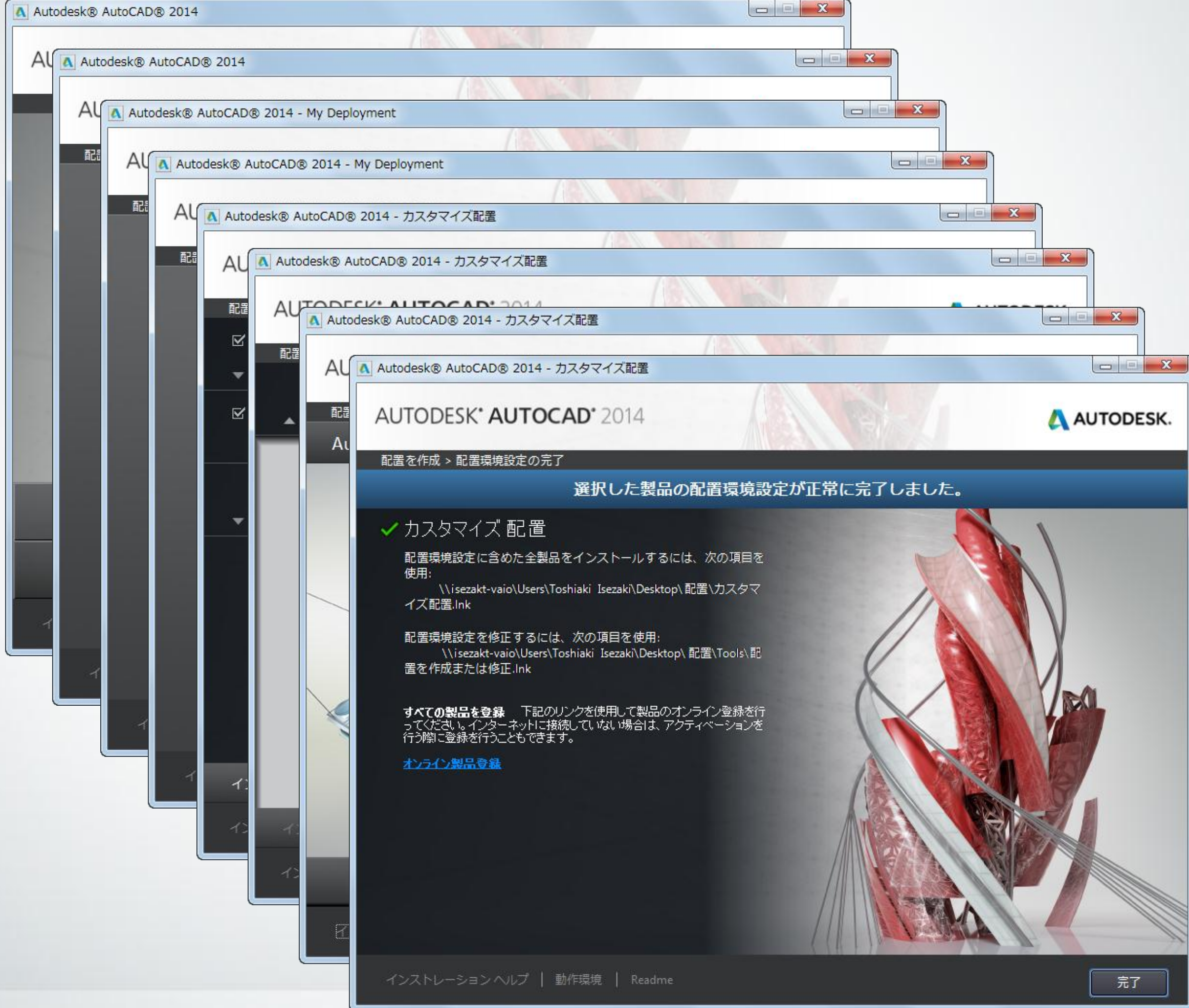
参照パス、参照情報も一緒にコピーしなければなりません !!

AutoCAD とカスタマイズ ファイルをインストール

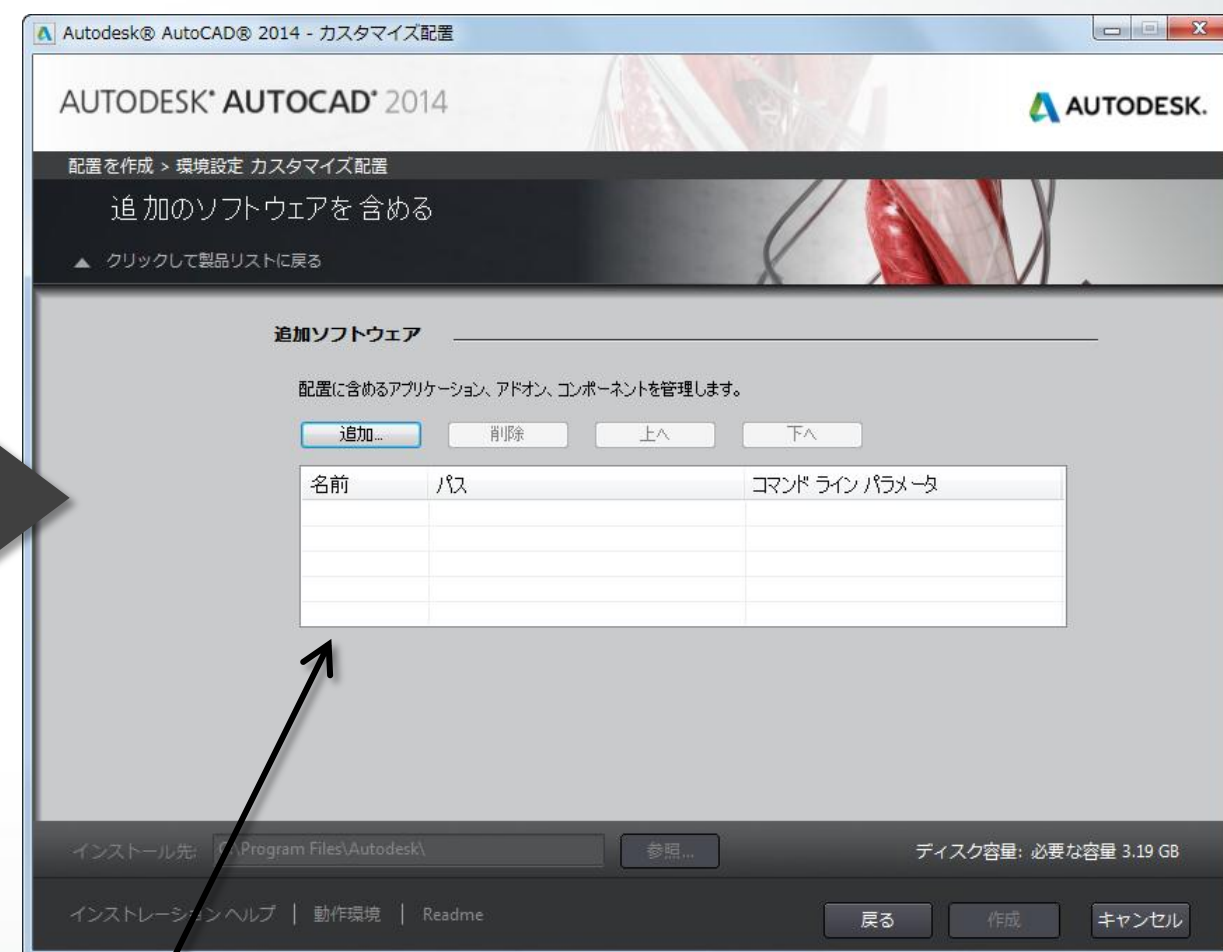
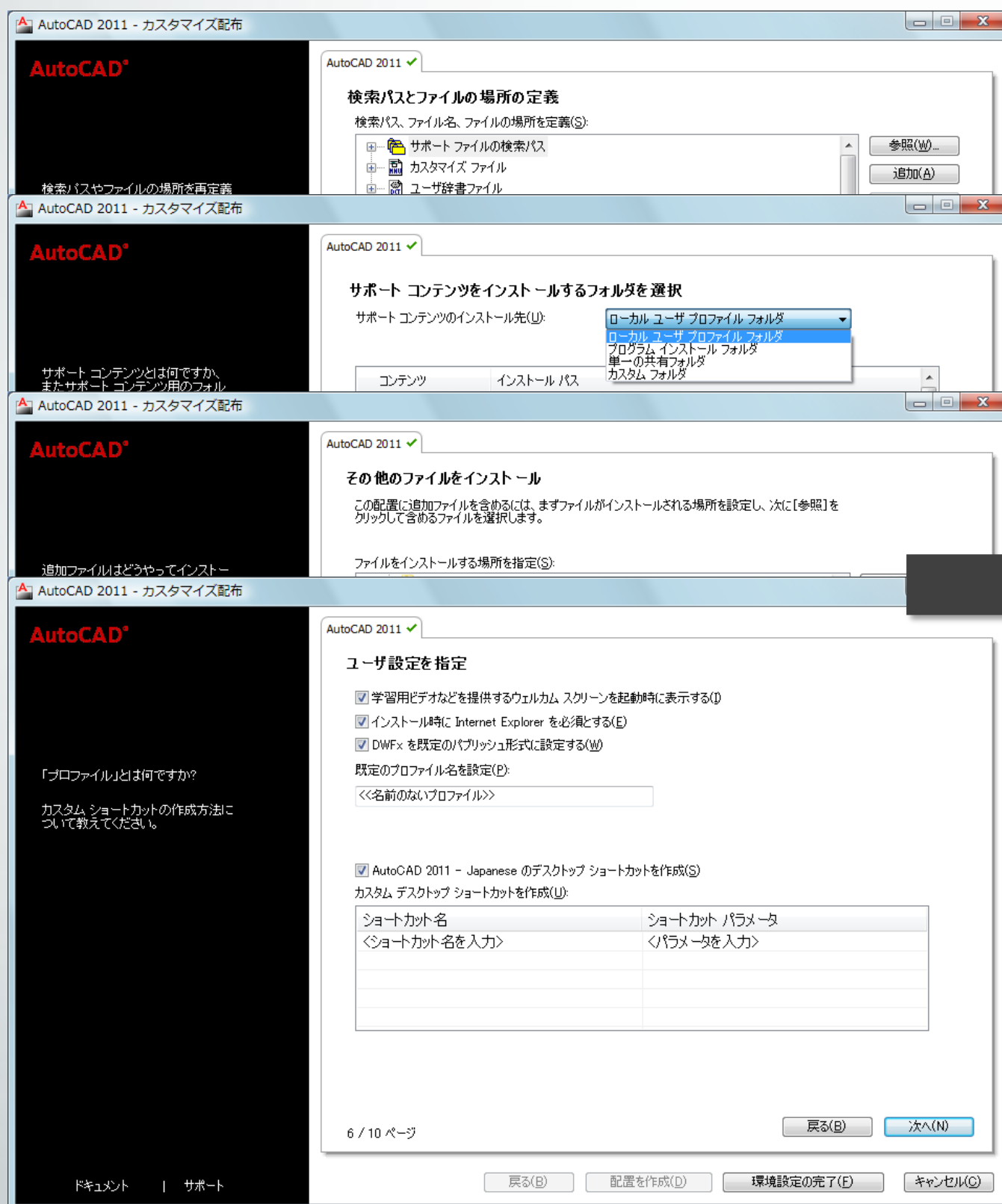
- 配置イメージを利用してカスタマイズ込みの AutoCAD をインストール
- 配置イメージとは？
 - サーバーの共有フォルダにインストール イメージを作成する (DVD-ROMの代わり)
 - 元々はネットワーク ライセンスのクライアントに AutoCAD をインストールする手法



AutoCAD の DVD-ROM から
起動したインストーラから作成します



配置イメージ作成の従来バージョンからの変更点



Exchange Apps モジュール形式の配布を前提
(自動ローダーで利用するバンドルフォルダ構造)



Autodesk is a registered trademark of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product and services offerings, and specifications and pricing at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.