

# AutoCAD 2015へのカスタマイズ移植

オートデスク 株式会社  
伊勢崎 俊明

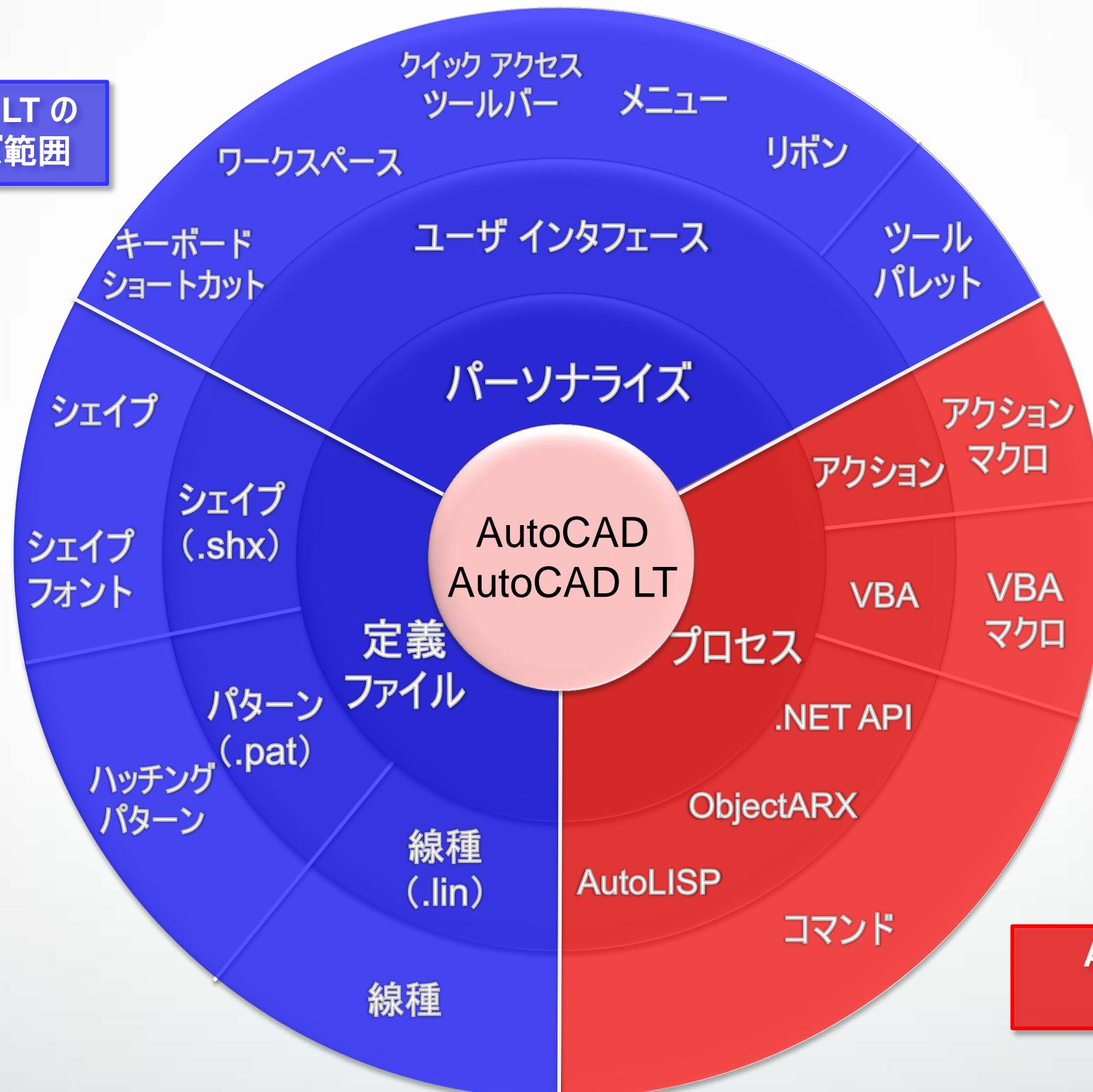


# 参照情報 - ドキュメント

- 一般情報
  - <http://www.autodesk.co.jp/developautocad>
  - <http://www.autodesk.co.jp/tech-faq>
- AutoCAD 2015 オンラインヘルプ
  - AutoLISP 開発者ガイド/リファレンス
  - ActiveX/VBA 開発者ガイド/リファレンス
  - AutoCAD .NET API 開発者ガイド
- ObjectARX SDK
  - <http://www.autodesk.com/objectarx>
  - ObjectARX Developer Guide(英語)
  - ObjectARX Reference Manual(英語)
  - .NET API Reference Manual(英語)

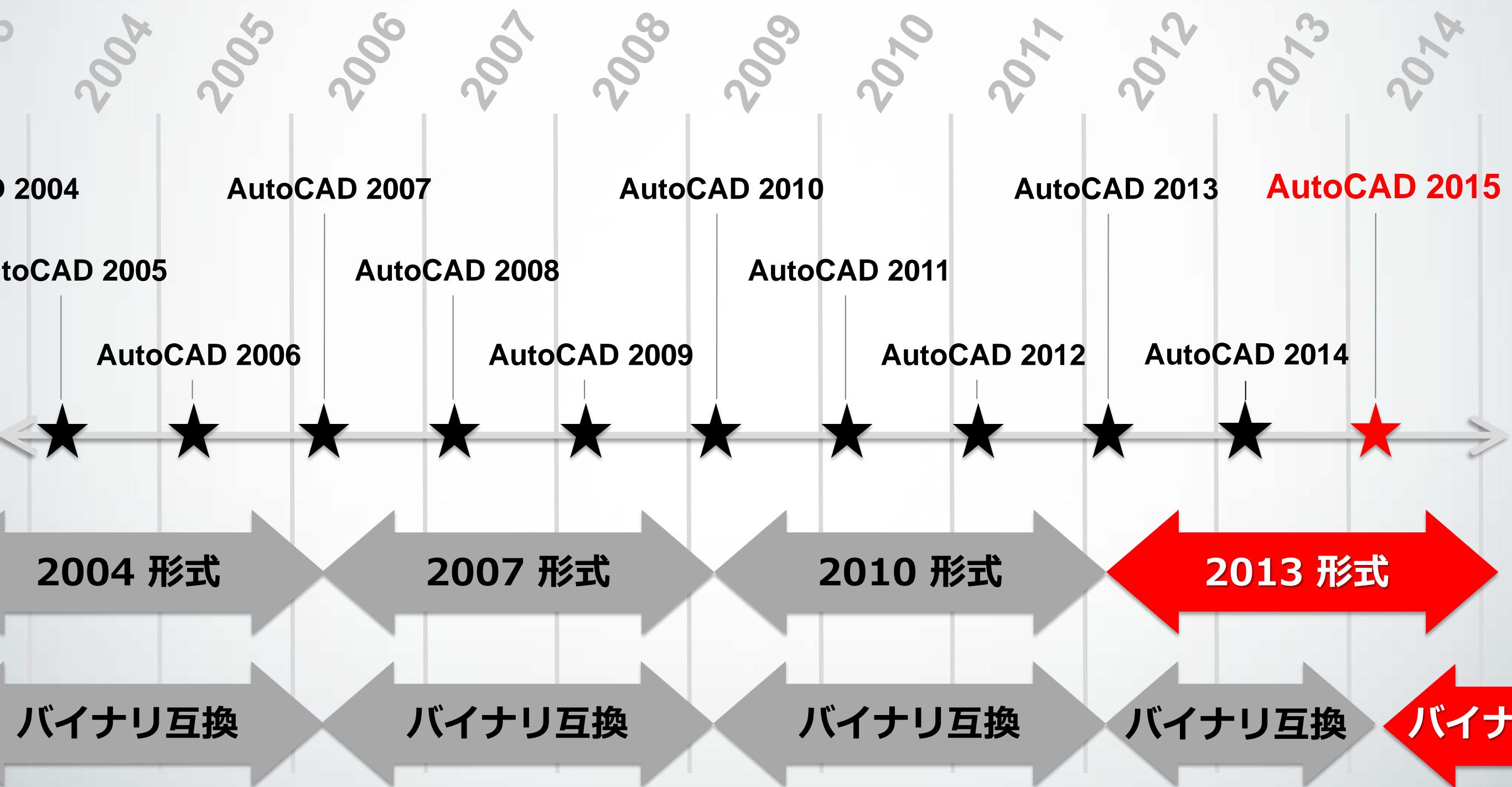
# カスタマイズ可能な項目

AutoCAD と AutoCAD LT の  
両方で可能なカスタマイズ範囲



AutoCAD でのみ可能な  
カスタマイズ範囲

# 図面ファイルとアドオン アプリケーションの互換性

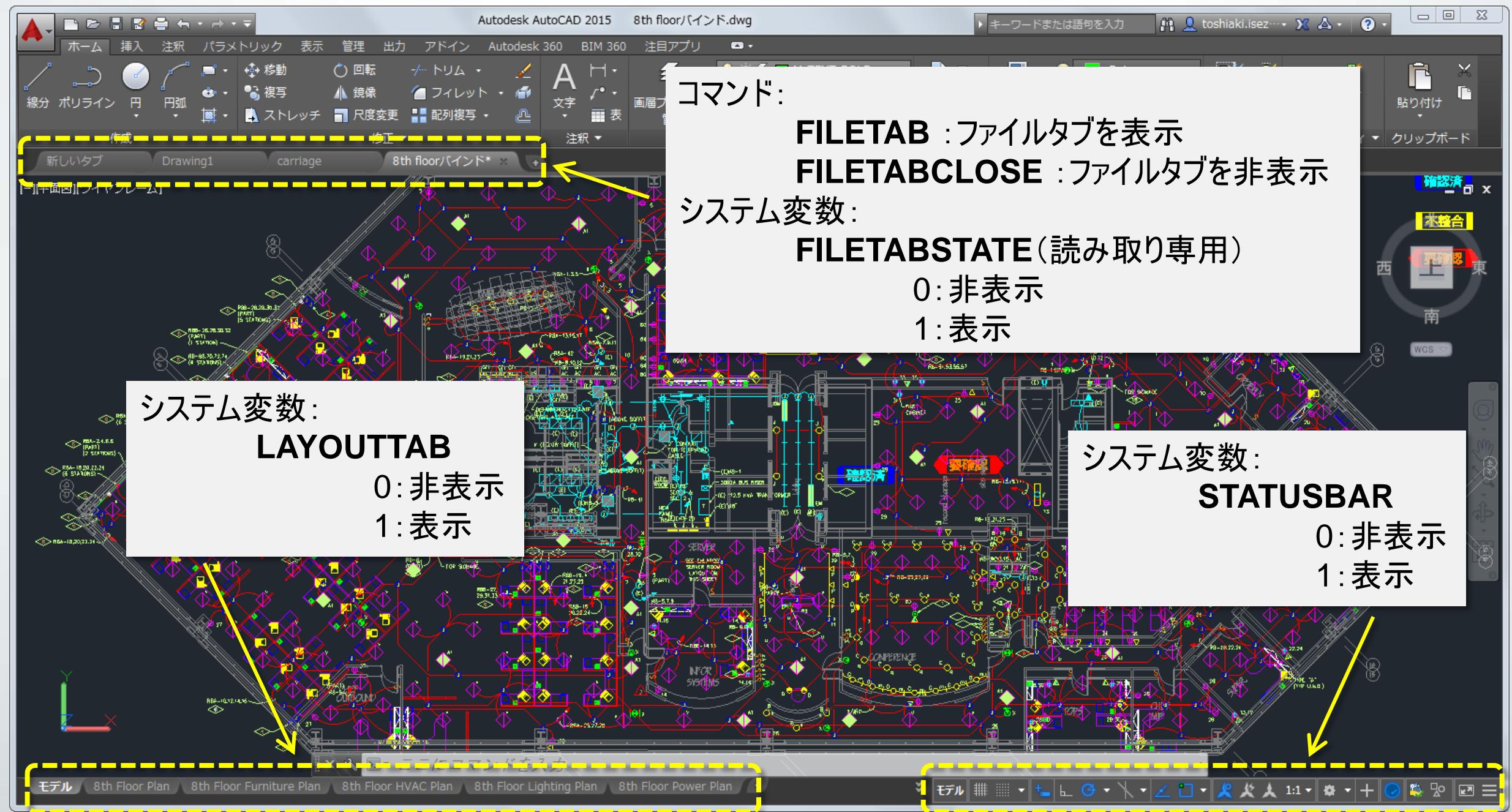


# アジェンダ – ご紹介する項目

- ユーザインターフェース カスタマイズの移植
  - ダークテーマ サポート
  - ツールバー、プルダウンメニュー、ワークスペース
- API カスタマイズで必須となる移植作業
  - [新しいタブ] を持つ MDI 環境への対応
  - ファイバー削除への対応
  - 新しいステータスバー
  - コードセキュリティ対策
- API カスタマイズで追加可能な機能

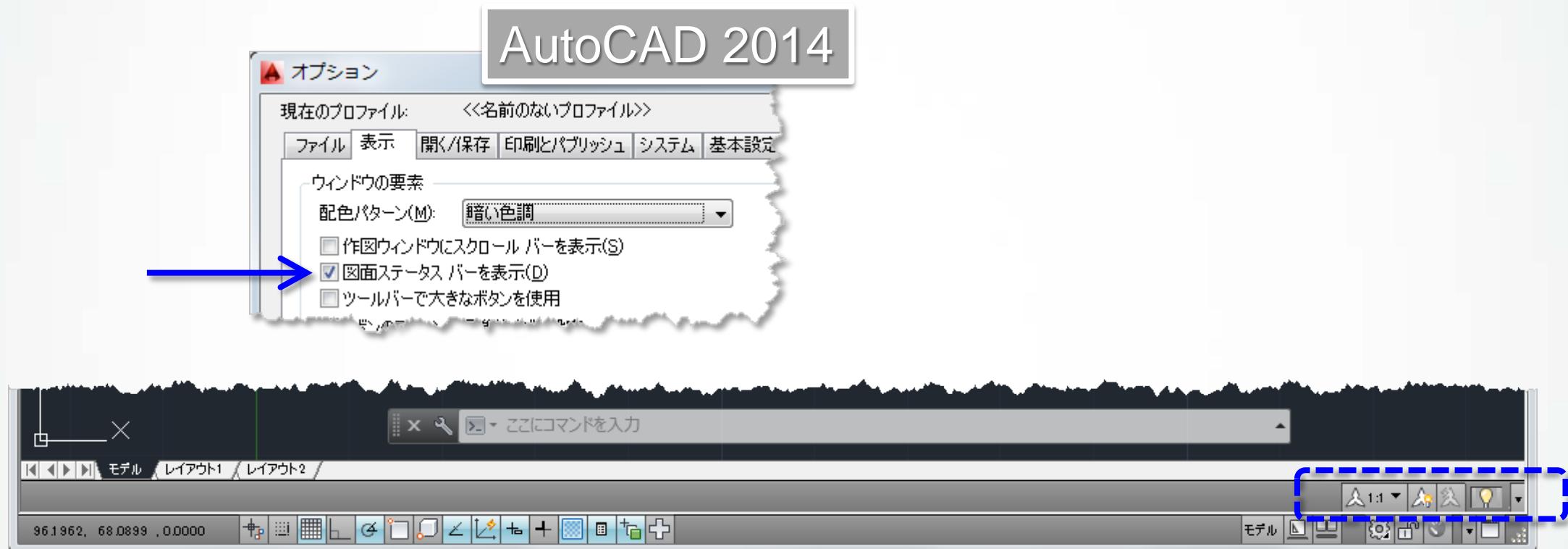
# ユーザインターフェースの移植作業

# タブ関連をコントロールするシステム変数/コマンド



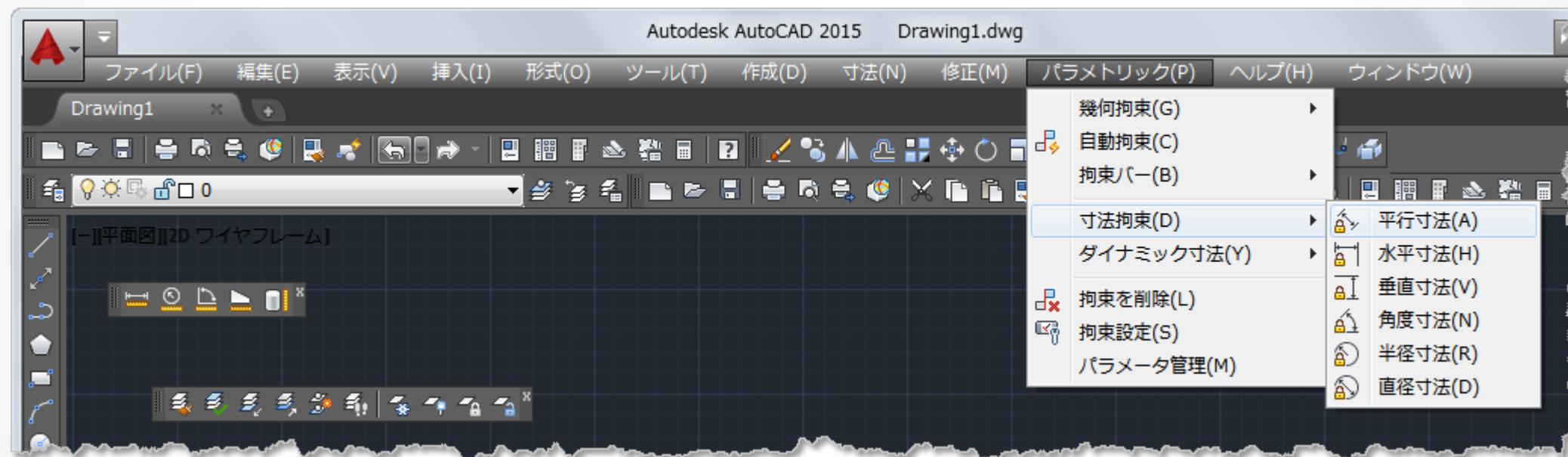
# ステータスバーの注意事項

## ■ 図面ステータスバーの表示機能削除



# ワークスペースとツールバー

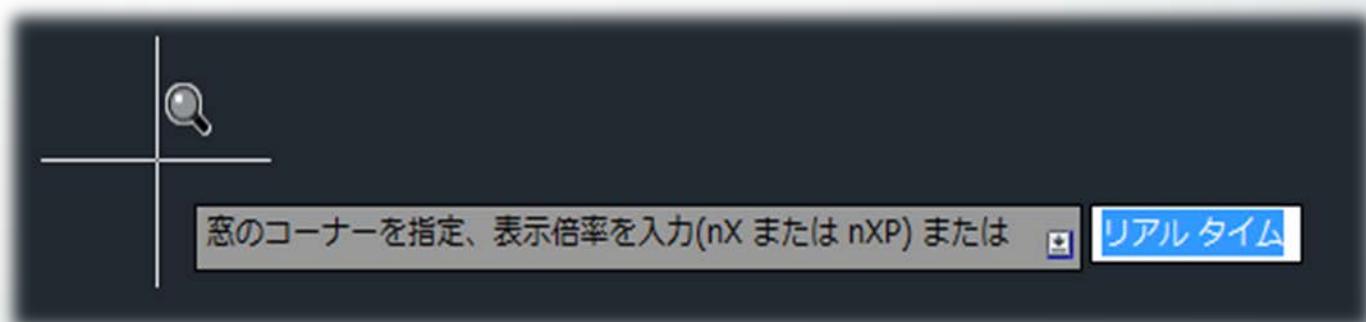
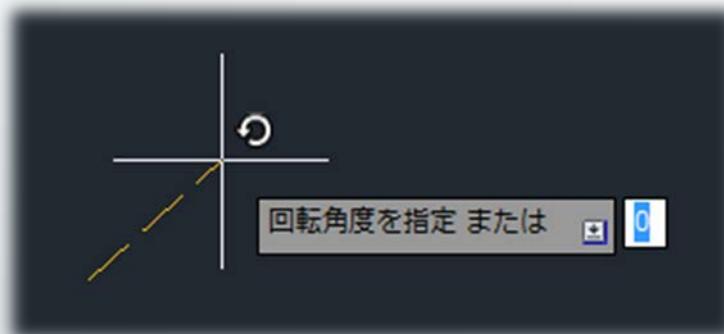
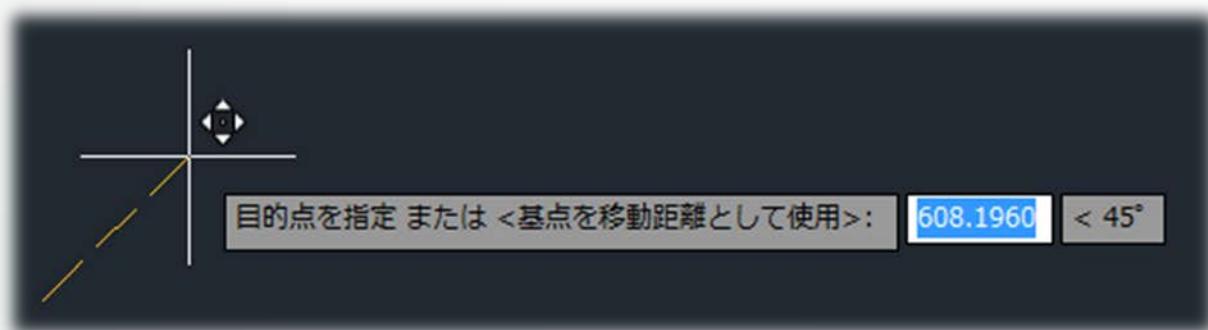
- [クラシック] ワークスペースは削除されました
  - プルダウンメニューとツールバーは引き続き利用可能



- QA-8844 Autodesk AutoCAD 2015 でクラシック表示にすることができますか?  
[http://tech.autodesk.jp/faq/faq/adsk\\_result\\_dd.asp?QA\\_ID=8844](http://tech.autodesk.jp/faq/faq/adsk_result_dd.asp?QA_ID=8844)

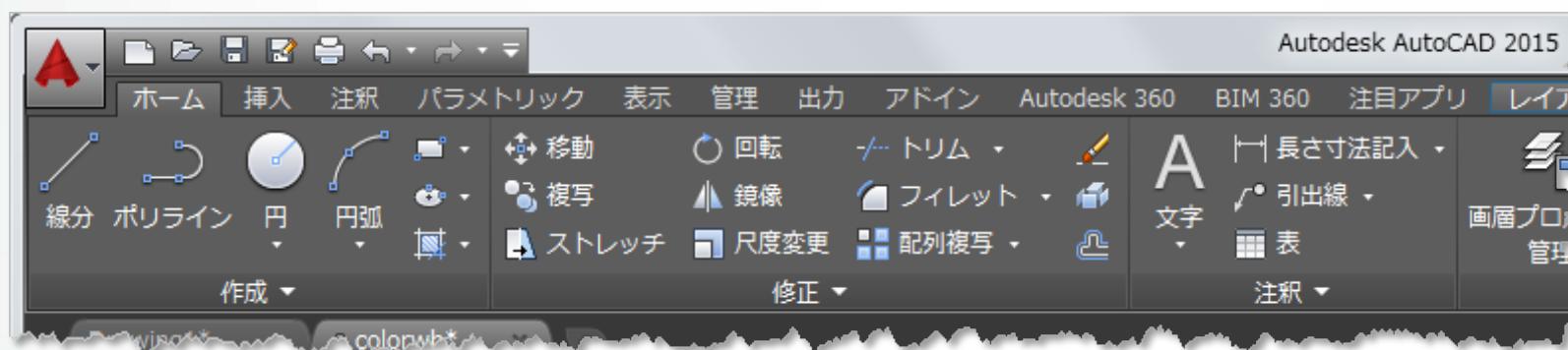
# マウスカーソルのバッヂ表示

- コマンド実行中のクロスヘア カーソル右上の操作支援表示
  - 現時点で表示/非表示のコントロールは不可
  - SP1 で新システム変数 **CURSORBADGE** 導入予定



# ユーザ インタフェースのモダン化

- ダークテーマ が既定
  - リボン、アイコン、WPF パレット



- システム変数 COLOR THEME で制御可能
  - 0: ダーク テーマ
  - 1: ライト テーマ
- ダイアログボックスはダークテーマ適用対象外

# ダークテーマとライトテーマへの対応

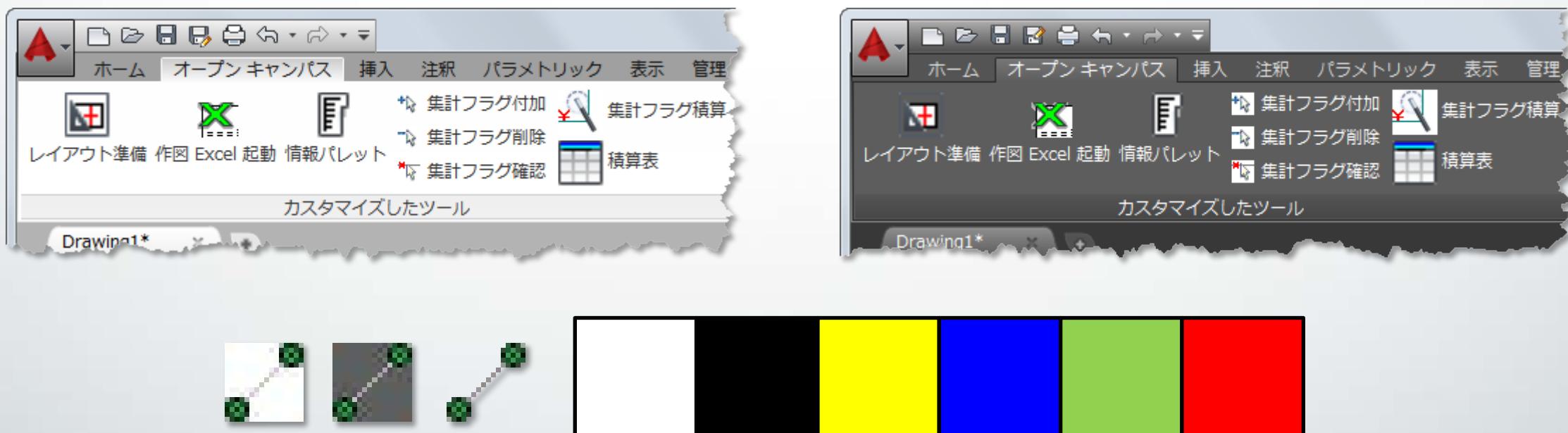
- テーマ別の CUIx ファイルを用意する方法

- オンラインヘルプに概要説明あり

<http://help.autodesk.com/view/ACD/2015/JPN/?guid=GUID-9AA3EEC1-031C-4E88-A966-4CBC07AD112C>

- アイコン画像の工夫で対応

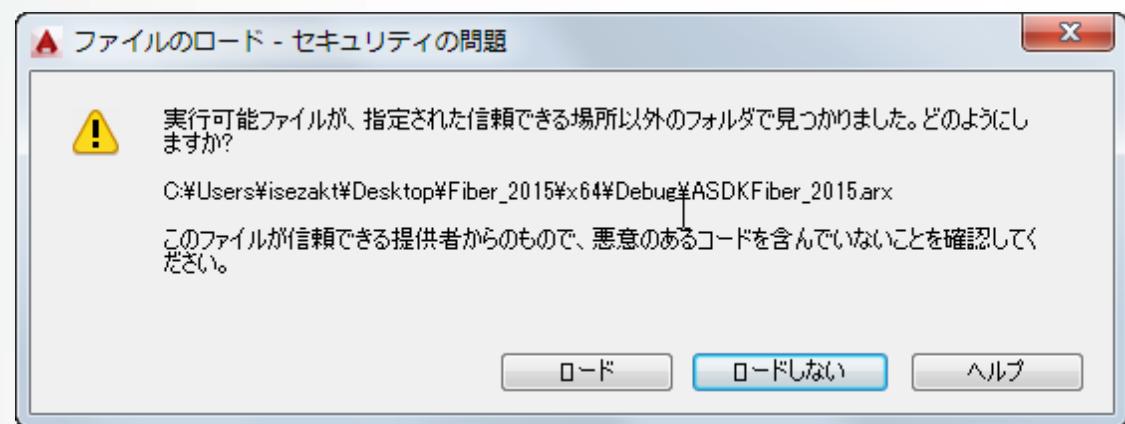
- AutoCAD 2015 で PNG ファイルをサポート
  - PNG ファイルの透過性を利用可能



# API カスタマイズで必須となる移植作業

# API カスタマイズ ファイルのセキュア ロード

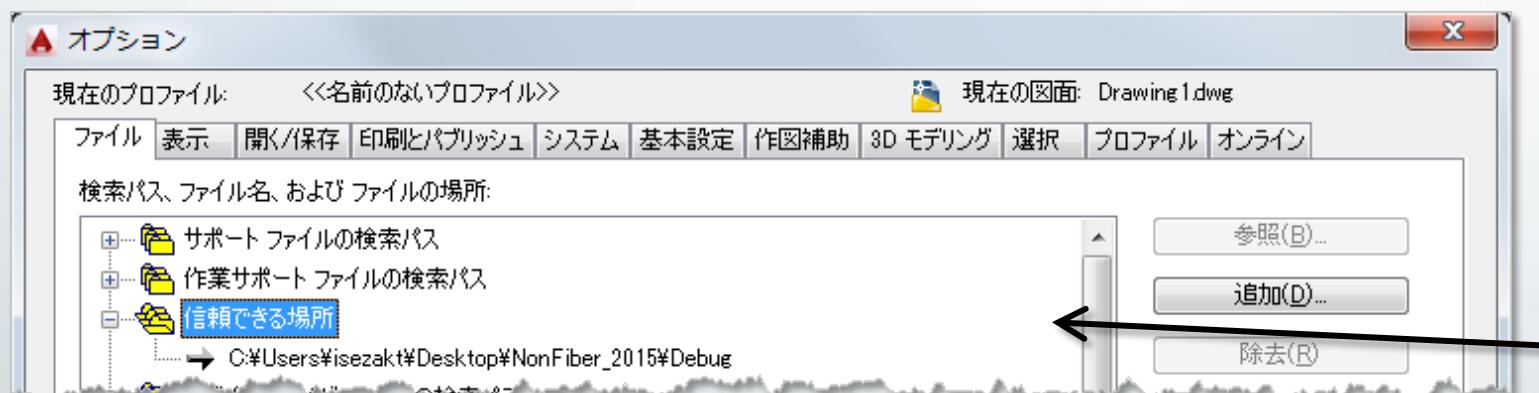
## ■ API カスタマイズ ファイル ロード時に警告ダイアログ



恶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(推奨)
  - *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)

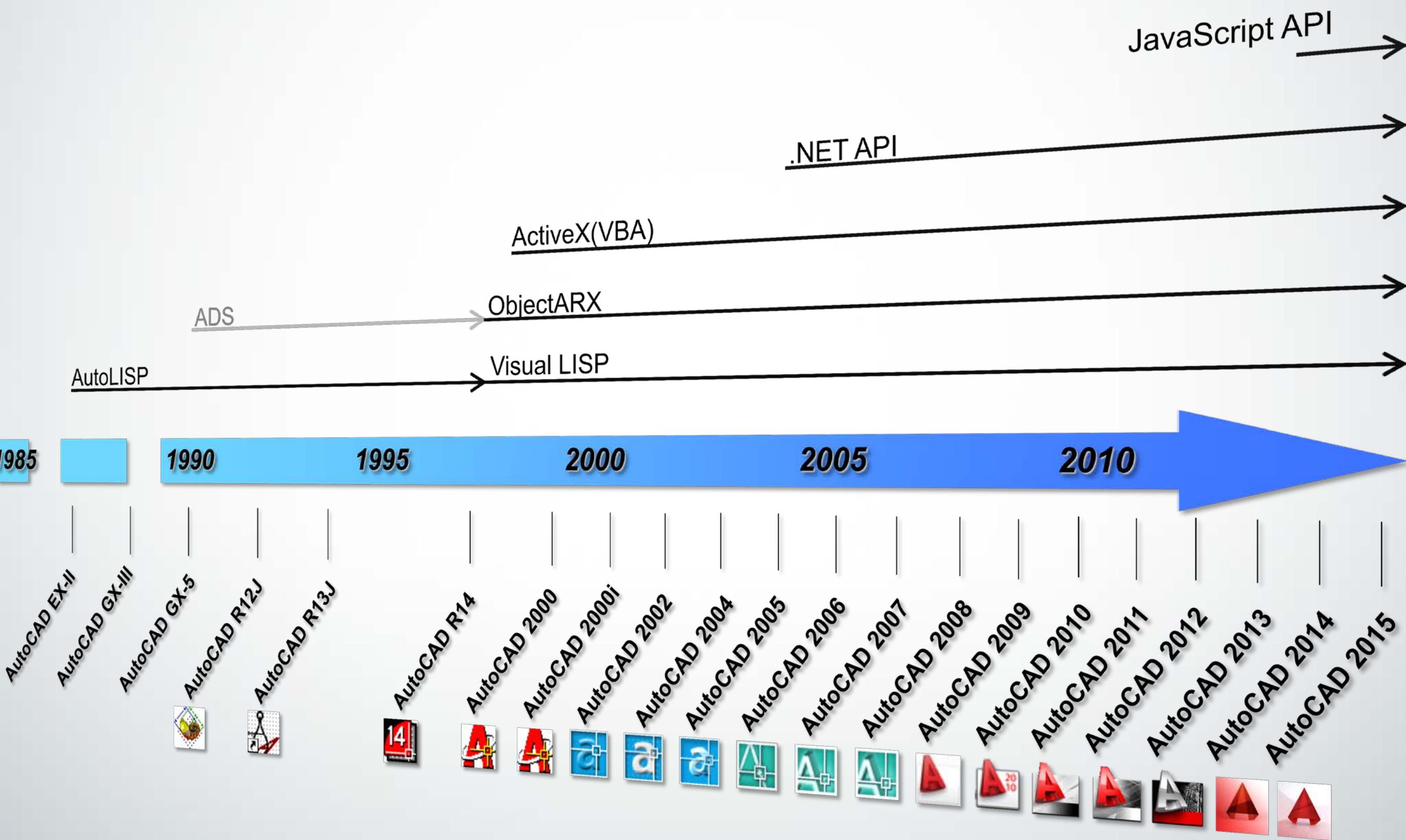
# 移植作業の基礎

- **バイナリ非互換リリース**
  - メジャー バージョン番号は R20.0 に増加
  - Visual Studio 2012 Update 4
    - .NET Framework 4.5
    - Visual C++ 11
- **DWG ファイル形式 – 変更なし**
- **Windows XP** – AutoCAD 2015 ではサポートを中止
  - Microsoft も2014年4月でサポートを停止

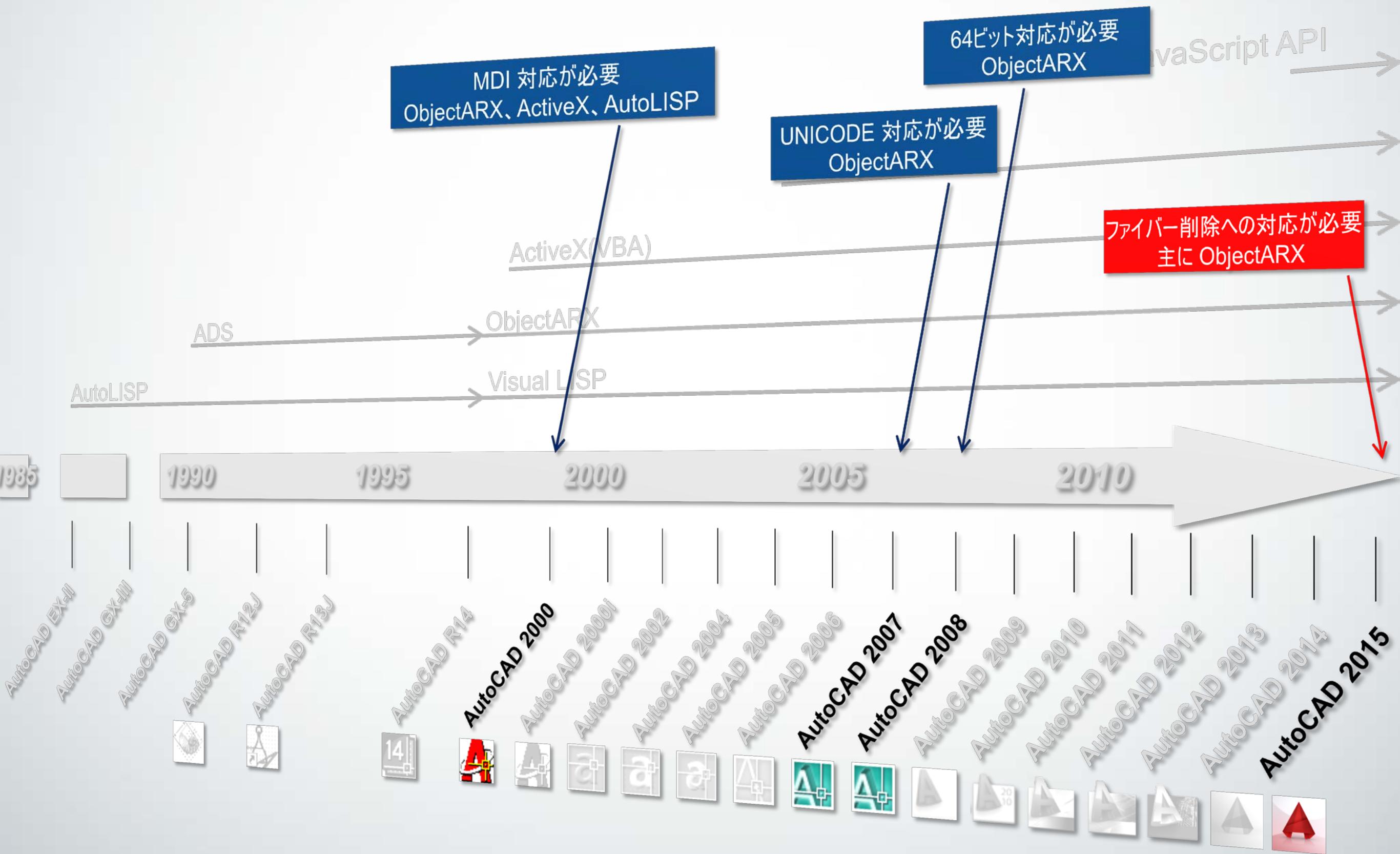
# 移植作業の基礎

- バイナリ非互換リリース
  - **ObjectARX**
    - ObjectARX SDK 2015 で再ビルド
  - **.NET API** – 変更された API を使用しない限り動作...
    - 2015 用アセンブリで再ビルドを推奨
  - **AutoLISP** – 変更された API を使用しない限り動作
    - 2015 用アセンブリで再ビルドを推奨
    - 2015 用タイプライブラリを参照することを推奨 (COM利用時)
  - **ActiveX/VBA** – 変更された API を使用しない限り動作
    - 2015 用タイプライブラリを参照することを推奨 (外部アプリ)
    - 2013 以前からの移行では 64 ビット VBA への移植が必要
  - **JavaScript API** – 再構成

# アドオン アプリケーション移植作業の壁

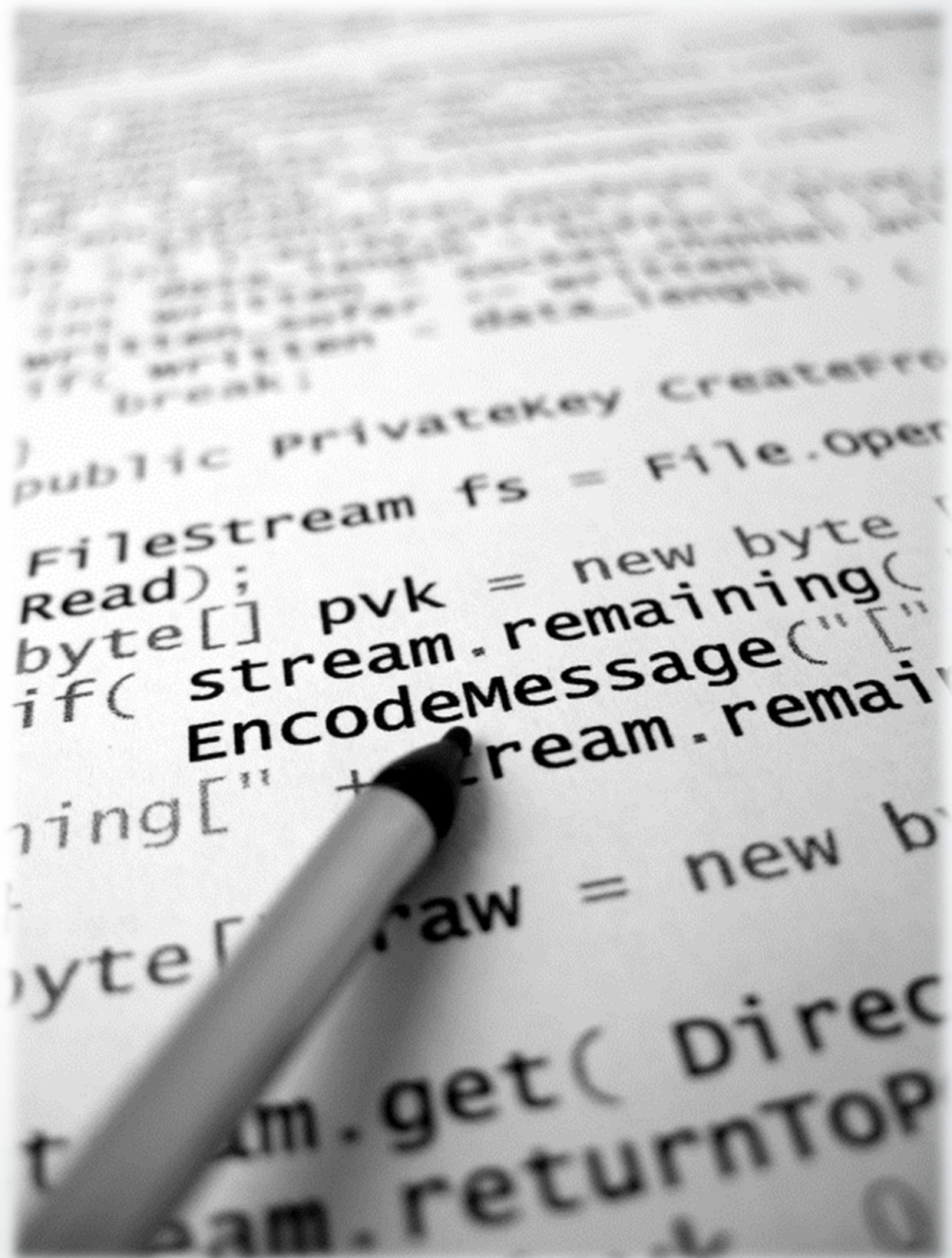


# アドオン アプリケーション移植作業の壁



# 移植作業の基礎

- テスト
- テスト
- テスト



# ファイバー削除

- ファイバーとは？
  - マルチタスクを実現するための古い手法
    - すでに Microsoft は Windows 上でサポートを停止
  - AutoCAD では AutoCAD 2000 から採用
    - 子ウインドウ(図面)毎のコマンド実行をサポート 
- ファイバー削除の利点
  - Visual Studio は全てのブレークポイントをヒット
  - .NET で新しい API が有効に(例.acedCommand)
- ファイバー削除の欠点
  - acedCommand コードは再構成の必要性あり
  - コマンド実行中のドキュメント切り替えは未サポート 



# acedCommand

- acedCommand/acedCmd の置き換え対象
  - acedCommand**S**/acedCmd**S** – サブルーチン
    - ‘完結する’ コマンドの提供に使用
    - PAUSE は未サポート
    - 容易な移植作業 - ‘**S**’ を追加
  - acedCommand**C**/acedCmd**C** – 共通ルーチン
    - ‘未完結な’ コマンドの提供に使用
    - PAUSE をサポート
    - より複雑な移植作業 – コールバック関数を使用

# acedCommandS への移植例

- 前バージョンまでのコード

```
void foo(void)
{
    acedCommand(RTSTR, _T("_LINE"),
                RTSTR, _T("0,0"),
                RTSTR, _T("111,111"),
                RTSTR, _T(""),
                RTNONE);
}
```

- 2015 でのコード

```
void foo(void)
{
    acedCommandsS(RTSTR, _T("_LINE"),
                  RTSTR, _T("0,0"),
                  RTSTR, _T("111,111"),
                  RTSTR, _T(""),
                  RTNONE);
}
```

# acedCommandC への移植例

## ■ 前バージョンまでのコード

```
void foo(void)
{
    acedCommand( RTSTR, _T( "_LINE" ) ,
                 RTSTR, _T( "0,0" ) ,
                 RTSTR, _T( "111,111" ) ,
                 RTSTR, PAUSE,
                 RTNONE );
}
```

## ■ 2015 でのコード

```
void foo(void)
{
    acedCommandC( &myCallbackFn, NULL,
                  RTSTR, _T( "_LINE" ) ,
                  RTSTR, _T( "0,0" ) ,
                  RTSTR, _T( "111,111" ) ,
                  RTNONE );
}
```

>> 続く

# acedCommandC への移植例 ~ 続き

- 2015 でのコード

続き >>

```
static int myCallbackFn(void *pData)
{
    int nReturn = RTNONE;
    if (isCommandActive())
        nReturn = acedCommandC( &myCallbackFn, NULL,
                               RTSTR, PAUSE,
                               RTNONE );
    return nReturn;
}

static Adesk::Boolean isCommandActive()
{
    struct resbuf rb;
    acedGetVar(_T("CMDNAMES"),&rb);
    if (_tcsstr(rb.resval.rstring, _T("LINE")))
        return Adesk::kTrue;
    return Adesk::kFalse;
}
```

# ファイバー削除対応の注意点

- acedCmdNF.h ヘッダー インクルード時の問題
  - QA-8870 ビルド時に SDK 内のヘッダーで warning C4819 エラーになる  
[http://tech.autodesk.jp/faq/faq/adsk\\_result\\_dd.asp?QA\\_ID=8870](http://tech.autodesk.jp/faq/faq/adsk_result_dd.asp?QA_ID=8870)
- acedCommand で別の API プロセスを起動する場合
  - acedCommandC に置き換えが必要
    - 例) VBARUN コマンドで VBA マクロを実行、AutoLISP 関数を定義して内部でacedCommand呼出し
    - 処理実装の見直しが必要になる場合があります
- イベント ハンドラ、リアクタの発生順序
  - AutoCAD 2015 で変化している可能性大
  - 処理実装の見直しが必要になる場合があります

# ドキュメント切り替え

- コマンド実行中の切り替えは今後サポートされません
  - \*キャンセル\* が送信されます
- MdiActiveDocument が **null** を返す可能性をもあり
  - 既存コード全域でチェックが必要
  - イベント処理内、モードレス ダイアログ、パレット…



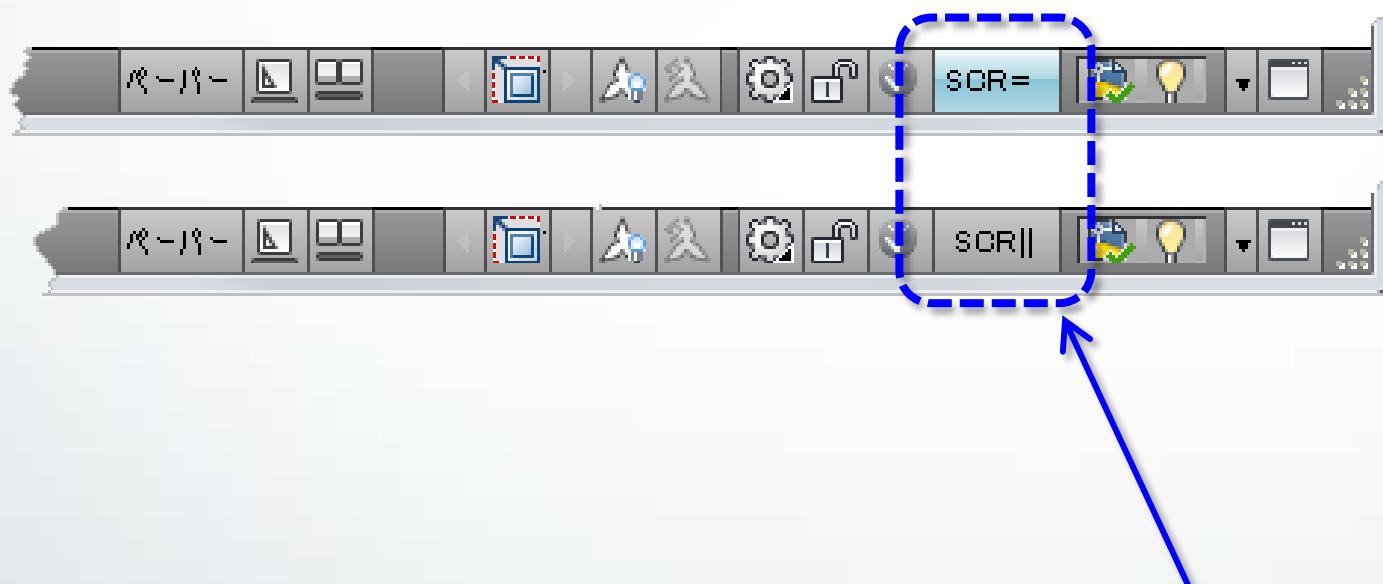
[新しいタブ] の採用で必ずしも  
図面がアクティブではなくなります

# ユーザ インタフェースのモダン化

- ステータスバー は WPF に移行



- 旧 MFC API は .NET で置き換えが必要
  - Autodesk.AutoCAD.StatusBar 名前空間



独自ステータスボタンを実装した ObjectARX アプリは注意

# ObjectARX コード セキュリティ



- Microsoft が提唱/提供する
  - 危険性を持つ関数に警告を与える
- AutoCAD は **banned.h** を使ってコンパイル
  - 主に文字列操作時の危険性を低減させる目的
  - バッファオーバーフロー ...
- 同様の作業を実施をお願いします
- 詳細は...
  - QA-8892 セキュリティ強化に伴う banned.h ヘッダー利用について

[http://tech.autodesk.jp/faq/faq/adsk\\_result\\_dd.asp?QA\\_ID=8892](http://tech.autodesk.jp/faq/faq/adsk_result_dd.asp?QA_ID=8892)

# サンプルはどこに？

- GitHub(無償のプログラム リポジトリ)に記載

<https://github.com/ADN-DevTech/AutoCAD-SendCommand>

- 誰でもダウンロードが可能です



# API カスタマイズで追加可能な機能

# .NET API でのネイティブ コマンドの呼び出し

- P/Invoke による ObjectARX 関数呼び出しの必要はもうなし
  - QA-8310  
.NET API アドオンからAutoCADの標準コマンドを同期的に呼び出したい  
[http://tech.autodesk.jp/faq/faq/adsk\\_result\\_dd.asp?QA\\_ID=8310](http://tech.autodesk.jp/faq/faq/adsk_result_dd.asp?QA_ID=8310)
- Editor.Command メソッド:acedCommandS 相当

```
[CommandMethod("MyCommand")]  
public static void Foo()  
{  
    Editor ed = Application.DocumentManager.MdiActiveDocument.Editor;  
    ed.Command(new Object[] { "_.LINE", "0,0,0", "10,10,0", "" });  
}
```

# .NET API でのネイティブ コマンドの呼び出し

- Editor.CommandAsync メソッド:acedCommandC 相当
  - .NET 4.5 の **async** と **await** キーワードを使用

```
[CommandMethod( "MyCommand" )]  
public static async void Foo()  
{  
  
    Editor oEd = Application.DocumentManager.MdiActiveDocument.Editor;  
  
    // 指定箇所で半径 2.0 の円を作成  
    await oEd.CommandAsync( "_CIRCLE" );  
  
    // ユーザ入力を待つ  
    await oEd.CommandAsync( Editor.PauseToken );  
  
    // 半径入力を提供  
    await oEd.CommandAsync( 2.0 );  
  
}
```

# DWGなしのMDIドキュメント ウィンドウ

- ドキュメント ウィンドウに固有コンテンツを含める
  - [新しいタブ]
    - JavaScript コンテンツを持つ HTML ページを埋め込み
    - 現時点ではオフィシャルなサポートではありません



# 自動ローダーの拡張

- Autodesk Exchange Apps 対応
- レジストリ、システム変数、**SetEnv** をサポート
- **WPF** ユーザコントロール サポートの拡張
  - コンテキストタブと選択ルール
  - F1 拡張ヘルプ Xaml
  - DWG ウィンドウ Xaml ツールチップ
  - ‘Big A’ メニュー（アプリケーションメニュー）エントリ
- **JavaScript** と **VBA** サポートを追加

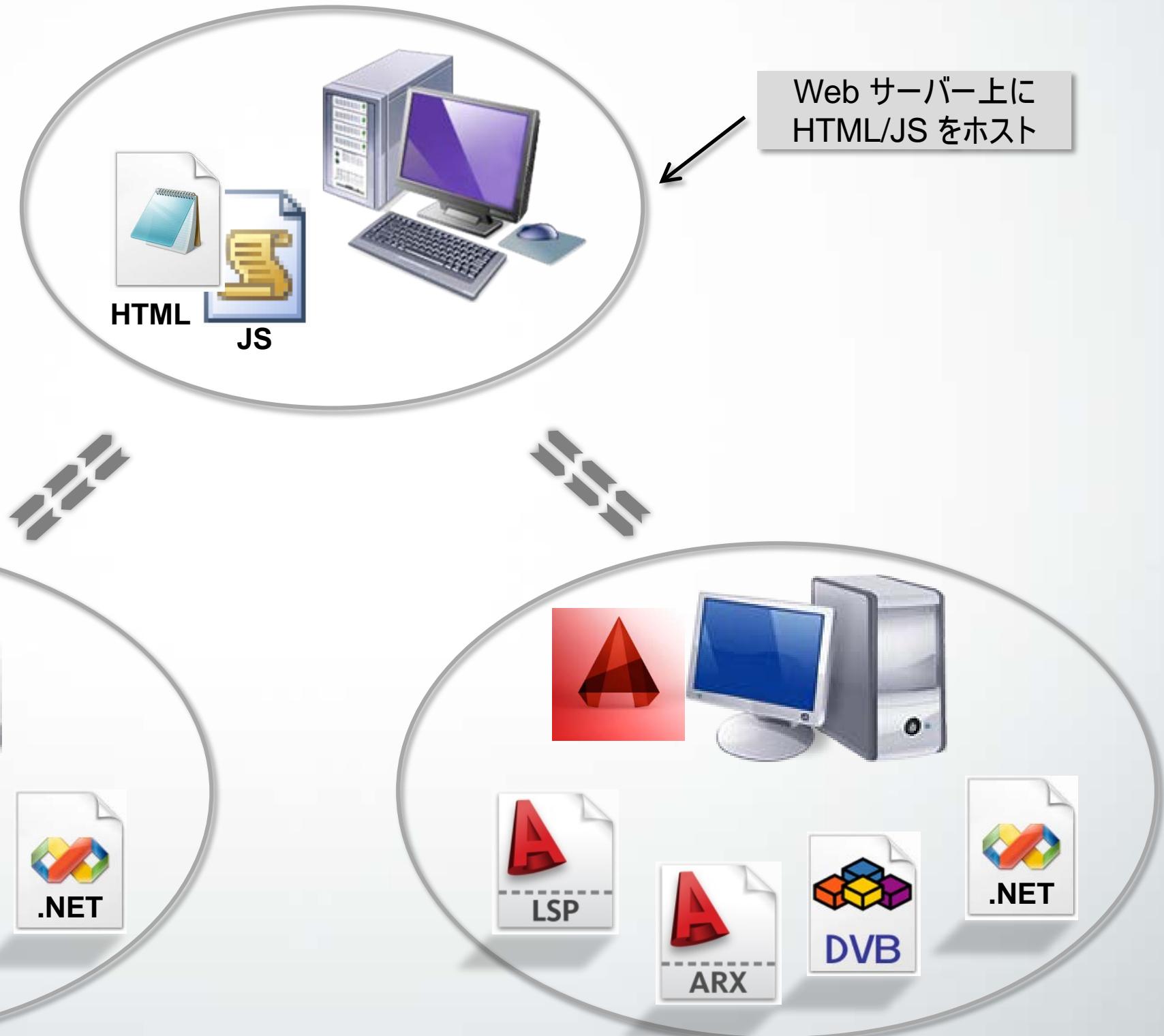
# AutoCAD JavaScript API

JS

- バージョン 1.0 リリース
  - 前バージョンの JavaScript API は ‘プレビュー’ の位置付け
  - オブジェクトプロパティサポート(読み込み専用)
  - バインド可能なオブジェクトレイヤをサポート
  - 安定性の強化
    - アプリのフリーズが他のアプリに影響しない
- ドキュメントの整備
  - AutoCAD 2015 JavaScript API ガイド(英語)  
<http://app.autocad360.com/jsapi/v2/GettingStart/index.html>
  - AutoCAD 2015 JavaScript API リファレンス(英語)  
<http://app.autocad360.com/jsapi/v2/docs/index.html>

# JavaScript API の利点

カスタマイズ プログラムの  
集中管理が可能に



# JavaScript API のデバッグ環境

- WebInspector の利用が可能  
HTML コンテンツがアクティブなときに [F12] キーで表示

The screenshot shows the Autodesk Developer Tools interface. The title bar reads "Developer Tools - http://adndevblog.typepad.com/files/createcircle-2015.html". The address bar shows the URL "http://localhost:9308/devtools/devtools.html?ws=localhost:9308/devtools/page/bb543c17fc7a528239a3c6a20517478". The top menu bar includes Back, Forward, Reload, Stop, and a dropdown menu. Below the menu is a tab bar with Elements, Resources, Network, Sources (which is selected), Timeline, Profiles, Audits, and Console.

The main area displays the code for "createcircle-2015.html". The code is as follows:

```
11 var m_rad = 0.0;
12 var m_cpt = new Acad.Point3d(0.0, 0.0, 0.0);
13
14 function pointToString(pt) {
15     var ret =
16         pt.x.toString() + "," +
17         pt.y.toString() + "," +
18         pt.z.toString();
19     return ret;
20 }
21
22 function doubleToString(value) {
23     return value.toString();
24 }
25
26 function myCircle() {
27     try {
28         var ppo = new Acad.PromptPointOptions();
29         ppo.useBasePoint = false;
30         ppo.allowNone = false;
31         ppo.setMessageAndKeywords("\n中心点を入力", "");
32         Acad.Editor.getPrompt(ppo).then(onCompleteCpt, onError);
33     }
34     catch(ex) {
35         write(ex.message);
36     }
37 }
38
39 function onCompleteCpt(jsonPromptResult) {
40     try {
41         var resultObj = jsonPromptResult;
42         if (resultObj) {
43             try {
44                 m_cpt = resultObj.value;
45             }
46         }
47     }
48 }
49
50 Acad.Curve.createCircle(m_cpt, m_rad, Acad.CurveType.CenterPoint);
51
52 Acad.Editor.setObject(m_cpt, Acad.ObjectType.CenterPoint);
53 Acad.Editor.setObject(m_cpt, Acad.ObjectType.CenterPoint);
```

The code editor highlights line 28 and line 41. The right side of the interface features the "Breakpoints" panel, which lists several breakpoints. Two breakpoints are currently active, indicated by checked checkboxes: "createcircle-2015.html:28" and "createcircle-2015.html:41". Other listed breakpoints include "createcircle-2015.html:11", "createcircle-2015.html:14", "createcircle-2015.html:22", and "createcircle-2015.html:39". The panel also includes sections for "Watch Expressions", "Call Stack", "Scope Variables", "DOM Breakpoints", "XHR Breakpoints", "Event Listener Breakpoints", and "Workers".

# 他の新しい API

- 図面修復
  - ObjectARX と .NET API
  - Acad::ErrorStatus  
`AcApDocManager::appContextRecoverDocument(`  
`const ACHAR *pszDrawingName )`
  - void  
`DocumentCollection::AppContextRecoverDocument(`  
`string fileName )`
- 図面監査
  - ObjectARX と .NET API
  - Acad::ErrorStatus `acedAudit(AcDbDatabase* pDb,`  
`bool bFixErrors, bool bCmdLnEcho = false)`
  - static void DatabaseExtension.**Audit**(Database db,  
bool bFixErrors, bool bCmdLnEcho)

# 他の新しい API

- ‘ドキュメント毎’ データ属性
  - .NET のみ
  - アセンブリ コンテキスト内の PerDocumentClass 属性
  - 例)  
`[assembly: PerDocumentClass(typeof(MyClass1))]`
- システム変数イテレタ
  - ObjectARX と .NET API
    - ObjectARX : `AcEdSysVarIterator`
    - .NET : `SystemVariableEnumerator`
  - 変数名、データタイプ、範囲(適用可能な場合)、読み込み専用ステータス、保存場所

# 他の新しい API

- WPF カラー データ バインディング クラス
  - .NET API のみ
  - 識別可能なコレクションを独自の WPF コントロールにバインド
- AcDbPointCloudEx クラス
  - ObjectARX のみ
  - 新しい点群エンティティを表現



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 offerings and specifications at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.