

Visual Studio 2005 時代の 64 ビットアプリケーション開発

～ Itanium® 上での .NET Framework 2.0 の展開

マイクロソフト株式会社
エンタープライズ プラットホーム本部
エバンジェリスト
佐藤 直樹

developer days

Agenda

- Windows 64-bit 概要
- マネージコード開発(導入)
- マネージコード開発(詳細)
- コーディング上の留意点
- まとめ

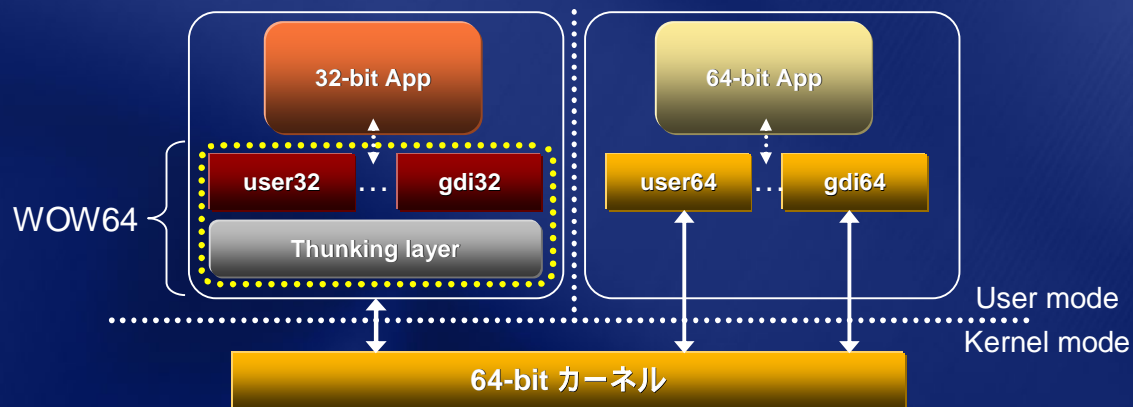
Windows 64-bit 概要

developer days
Microsoft Windows Vista

Windows 64-bit 概要

～ アプリケーションの観点から

- 64ビットアプリケーション
 - 64ビットシステムファイルを直接利用
- 32ビットアプリケーション
 - Windows on Windows 64 (WOW64) を経由して 64ビットシステムファイルを利用



developer days

Windows 64-bit 概要

～ 64/32ビット相互運用における留意点

- 同一プロセス内には 32ビットコードと 64ビットコードは同居できない
 - ライブラリロード時にエラー
 - COM インプロセスサーバ (DLL サーバ)
 - IME, IEプラグイン, シェル拡張, ActiveX コントロール

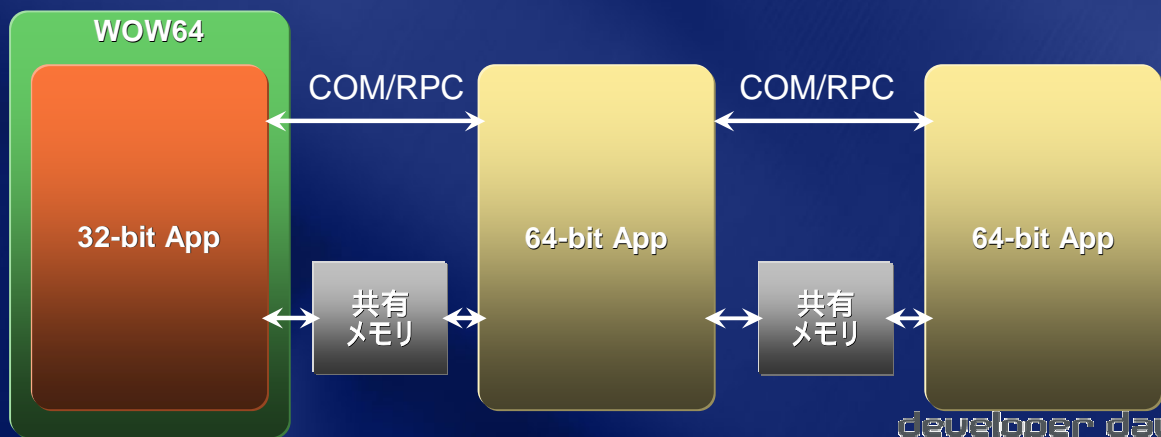


developer days

Windows 64-bit 概要

～ 64/32ビット相互運用における留意点

- アウトプロセス COM/RPC (EXE サーバ) を 64ビットと 32ビットプロセス間においてサポート
- 共有メモリ、イベント等のオブジェクトを共有
 - ミューテックス、セマフォ、ファイルハンドル、ウィンドウハンドル等



developer days

マネージコード開発(導入)

developer days

Visual Studio 2005

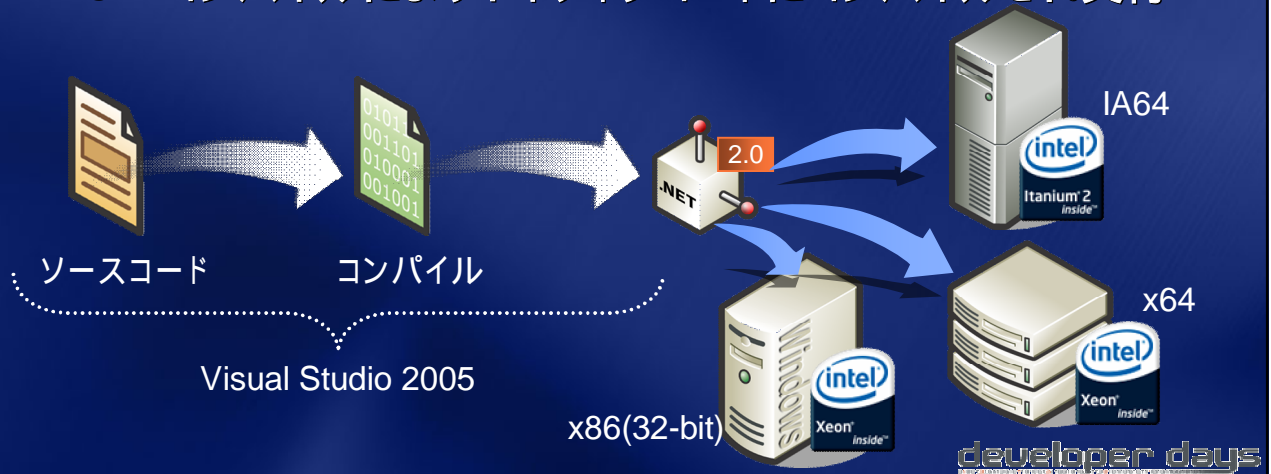
- 各言語の強化
- IDEの革新
- 開発ライフサイクルのサポート
- プラットホームとの統合
 - Windows CE、スマートフォンからデスクトップ、サーバ、64ビットサーバまで

Visual C++
Visual C#
Visual Basic
Visual J#



.NET Framework 2.0

- 管理コード、タイプセーフ、高パフォーマンス
- シングルバイナリ
 - 32ビット環境 (x86, WOW64)
 - 64ビット環境 (IA64 および x64)
- コンパイルオプションで特定プラットフォーム向けに作成
- JIT コンパイルによりネイティブコードにコンパイルされ実行



64ビット環境での .NET

- **.NET Framework 1.1**
 - WOW64 (32ビット) 上でのサポート
 - IPF では対話型クライアントをお勧め
- **.NET Framework 2.0**
 - 2つの64ビットCPUアーキテクチャ
 - IA64 : インテル® Itanium® プロセッサファミリー
 - x64 : インテル® EM64T , AMD AMD64
 - **WOW64 (32ビット)、32ビットCPUもサポート**
 - WOW64 : 64ビット向け再頒布可能パッケージに含む
 - 32ビット : x86向け再頒布可能パッケージ
 - **必須システム(64ビットプラットフォーム)**
 - Windows Server 2003 SP1 for Itanium-based Systems
 - Windows XP Professional x64 Edition
 - Windows Server 2003 x64 Editions

developer days

ターゲットプラットフォーム

- 開発言語別のサポート状況

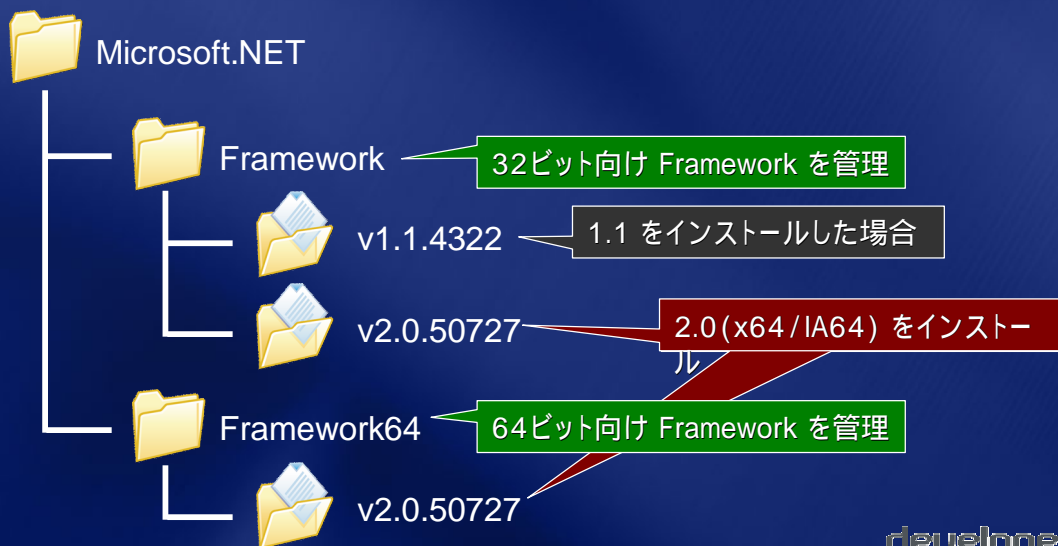
	マネージ		ネイティブ	
	32-bit	64-bit	32-bit	64-bit
Visual C++	✓	✓	✓	✓
Visual C#	✓	✓	✗	✗
Visual Basic	✓	✓	✗	✗
Visual J#	✓	✗	✗	✗

✓ : サポートする / ✗ : サポートしない

developer days

.NET Framework 2.0

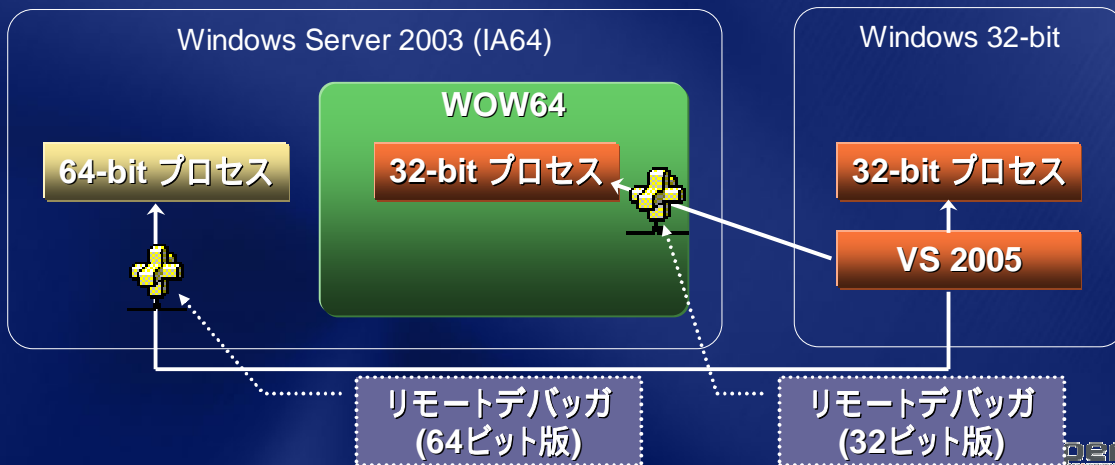
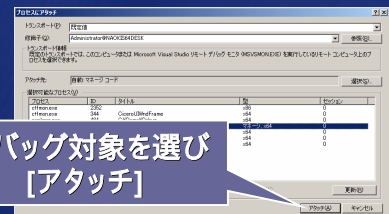
- WOW64 上で .NET Framework 2.0 を動作
 - .NET Framework 2.0(64ビット) には 32/64ビットの両方のランタイムが含まれる



developer days

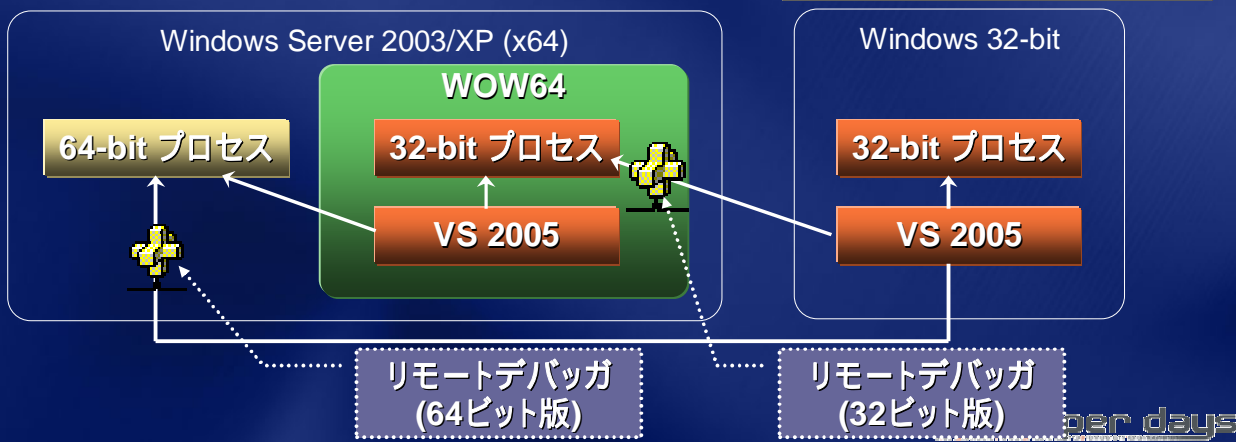
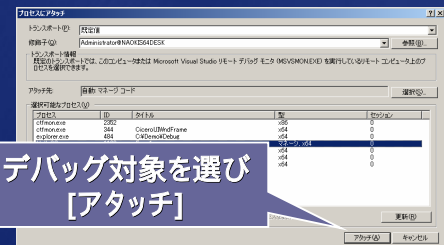
Visual Studio 2005

- IA64 をターゲットとするクロス開発
 - 32ビットOS上での開発
 - リモートデバッガによるデバッグ



Visual Studio 2005

- x64をターゲットとする クロス開発およびWOW64上
 - 32ビット OS、WOW64(x64) 上での開発をサポート
- ローカル、リモートでのデバッグ
 - WOW64 上の 32ビットプロセス
 - 64ビットプロセス



アプリケーション移行戦略

～ 64ビット化の道筋

- 64ビットへの移行のモデル
 - 32-bit OS上で現状維持 (32-bit)
 - WOW64 上でホスト (32-bit)
 - 一部 64-bit 化 (32-bit/64-bit 混在)
 - 64-bit 移植 (64-bit)
 - 再構築 (64-bit)
 - 業務の見直し、新業務への対応、新技術への対応
 - SOA, .NET Framework 対応



developer days

.NET Framework 1.1

～ 64ビット環境への移行の検討

- 1.1 ベースのアプリケーションを 64ビット環境で動作させる 4つの選択肢
 - を推奨。(場合によっては も)



マネージコード開発(詳細)

developer days

実行は 32ビット or 64ビット？

- 言語と実行プラットフォームとコンパイルオプションの関係
 - C#, Visual Basic : /platform オプション
 - C++ : ソリューションプラットフォームと /clr オプションの組み合わせ

	32ビット OS	64ビット OS		
	32ビット	WOW64	64ビット(x64)	64ビット(IA64)
Visual C++	/clr:safe /clr (Win32) /clr:pure (Win32)	/clr (Win32) /clr:pure (Win32)	/clr:safe /clr (x64) /clr:pure (x64)	/clr:safe /clr (IPF) /clr:pure (IPF)
Visual C#	anycpu x86	x86	anycpu x64	anycpu Itanium
Visual Basic	anycpu x86	x86	anycpu x64	anycpu Itanium
Visual J#	(x86)	(x86)	×	×

developer days

アプリケーションローディング

～ コンパイルオプションによる留意点

- 同一プロセス内に 32ビットコードと 64ビットコード、ターゲットマシンの違うコードは同居できない
 - アーキテクチャ ミスマッチ
 - ネイティブコードと同様の制約
 - /platform:anycpu, /clr:safe はロード



developer days

共有アセンブリ

- 共有アセンブリの格納先 **グローバルアセンブリキャッシュ (GAC) はアーキテクチャを認識**
 - 適切なアーキテクチャにインストールされる
 - i.e. x64 は IPF にインストールされない
 - アセンブリの検索順序:
 - 特定プロセッサ (x86, x64, IA64)
 - MSIL
 - Legacy (1.1)
- gacutil /lr**

WOW64
x86
MSIL
1.1

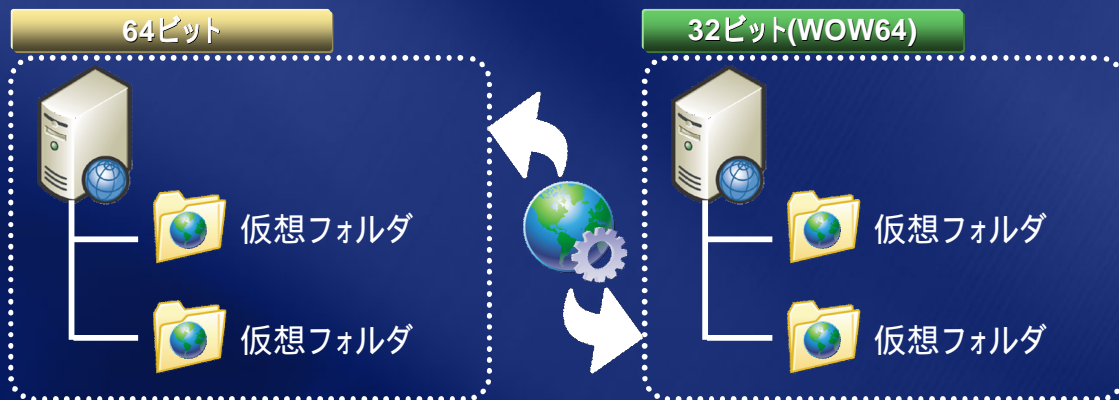
64ビット(IA64)
IA64
MSIL
1.1

Assembly Name	Version	Cult...	Public Key Token	Process...
System.DirectoryServ...	2.0.0.0		b03f5f7111d50a3a	MSIL
System.DirectoryServ...	2.0.0.0		b03f5f7111d50a3a	MSIL
System.Drawing	2.0.0.0		b03f5f7111d50a3a	MSIL
System.Drawing.Design	2.0.0.0		b03f5f7111d50a3a	MSIL
System.EnterpriseSer...	2.0.0.0		b03f5f7111d50a3a	x86
System.Management	2.0.0.0		b03f5f7111d50a3a	AMD64
System.Messaging	2.0.0.0		b03f5f7111d50a3a	MSIL
System.Runtime.Rem...	2.0.0.0		b77a5c561934e089	MSIL
System.Runtime.Seria...	2.0.0.0		b03f5f7111d50a3a	MSIL
System.Security	2.0.0.0		b03f5f7111d50a3a	MSIL
System.ServiceProce...	2.0.0.0		b03f5f7111d50a3a	MSIL



64ビット上でのIIS 6.0

- 64ビットもしくは32ビットのいずれかで動作
 - メタベース プロパティの値に従って32ビットもしくは64ビットワーカプロセスを開始
 - ASP.NET も同様に 64ビットもしくは32ビットのいずれかでの動作



developer days

64ビット上でのIIS 6.0

- 32ビットワーカークラスへの切り替え

```
cscript %SystemDrive%\inetpub\AdminScripts\adsutil.vbs set w3svc/AppPools/Enable32bitAppOnWin64 1
```

- 32ビット ASP.NET 1.1 のインストール

```
%SystemDrive%\Microsoft.net\Framework\v1.1.4322\aspnet_regiis -i
```

- 32ビット ASP.NET 2.0 のインストール

```
%SystemDrive%\Microsoft.net\Framework\v2.0.50727\aspnet_regiis -i -wow64 -enable
```

2.0 にマッピング

1.1 にマッピング



64ビット上でのIIS 6.0

- 64ビットワーカークラスへの切り替え

```
cscript %SystemDrive%\inetpub\AdminScripts\adsutil.vbs set w3svc/AppPools/Enable32bitAppOnWin64 0
```

- 64ビット ASP.NET 2.0 のインストール

```
%SystemDrive%\Microsoft.net\Framework64\v2.0.50727\aspnet_regiis -i -enable
```



developer days

コーディング上の留意点

developer days

コーディング上の留意点

- シリアル化
 - XML シリアル化は動作する
 - Web サービス
 - バイナリーシリアル化は注意が必要
 - ポインタサイズ System.IntPtr は 64ビットCLR環境ではサイズが違う
 - マネージコード間のアライメントは CLR によって処理されるので意識することはない
- 浮動小数点演算 (float/double)
 - 32ビット、64ビットで演算値が同一とは限らない
 - 64ビット化により境界部分での演算可能となる
 - 例: Sin(1e37)
- ポインタサイズ - System.IntPtr
 - 32ビット : 4バイト / 64ビット : 8バイト
- 環境変数
- Module.GetPEKind メソッド

developer days

ネイティブとの相互運用

- アライメント
- シグネチャ
 - DllImport, Declare, #include
 - System.IntPtr
 - 全てのプラットフォームでAPIが用意されていること
- マーシャリング
 - StructLayoutAttribute
 - Marshal.SizeOf
- COM
 - DLLサーバ : アーキテクチャの制約を受ける
 - EXEサーバ、COM+ サーバアプリケーション(サロゲートプロセス dllhost.exe) の利用

developer days

コード例

~ SequentialLayout 属性

マネージコード(クラス)

```
[StructLayout(LayoutKind.Sequential, Pack=1)]
public class Unicode1 {
    public byte by_minneg = unchecked((byte)-1);
    [MarshalAs(UnmanagedType.ByValArray, SizeConst=52)]
    public int[] padding = new int[13];
};
```

アンマネージコード(構造体)

```
#pragma pack(1)
typedef struct {
    BYTE by_minneg; // = (byte)-1;
    int padding[13];
} ASM_StructU1;
```

コード例

~ ExplicitLayout 属性

マネージコード(構造体)

```
[StructLayout(LayoutKind.Explicit)]
internal struct FooValue {
    [FieldOffset(0)] public int dwType;
    [FieldOffset(4)] public IntPtr pType;
    [FieldOffset(8)] public int typeValue;
}
```

要注意：
IntPtr はプラットフォームに依存
(4バイト or 8バイト)

フィールドオフセットでは他のフィールドサイズを考慮

developer days

まとめ

developer days
Microsoft

まとめ

- インテル® Itanium® プロセッサファミリーをサポートしている Windows Server 2003
そして .NET Framework 2.0
- Visual Studio 2005 により 64ビット環境に向けた .NET Framework アプリケーションの開発が可能
- プラットホーム限定するコンパイルオプションにより環境依存なコードとの相互連携

developer days

技術情報リソース

Windows 64 bit アプリケーション開発:

<http://www.microsoft.com/japan/msdn/windows/64bit/>

64-bit Windows (Development Guides):

<http://msdn.microsoft.com/library/en-us/dnanchor/html/64bitwindows.asp>

Microsoft .NET Framework SDK:

<http://msdn.microsoft.com/library/en-us/dnanchor/html/netfxanchor.asp>

Microsoft Platform SDK:

<http://msdn.microsoft.com/platformsdk/>

Microsoft[®]

Your potential. Our passion.[™]

© 2006 Microsoft Corporation. All rights reserved.

This presentation is for informational purposes only. Microsoft makes no warranties, express or implied, in this summary.