



**Future  
Technology Days**

# 64bitプラットフォームへの アプリケーション移行

マイクロソフト株式会社  
デベロッパー & プラットフォーム 統括本部  
井戸 文彦

# Agenda

- ◎ Windows 64bit 概要
  - ◎ 最新CPUのトレンド
- ◎ アプリケーションの動作環境
- ◎ アプリケーション移行
- ◎ .NETFramework ベース  
アプリケーションの移行
- ◎ コーディング上の留意点
- ◎ まとめ

The slide features a black background with decorative wavy lines in shades of blue and green at the top and bottom edges. The main title is centered in the middle of the slide.

# Windows 64bit 概要

# Windows 64bit 概要

Windows Server 2008 R2 は  
64bit バージョンのみのリ  
リースのみとなります。

## ◎ Windows 64bit のゴール

- 4GB 超のメモリ空間をフラットに利用する拡張性
- エンタープライズレベルの可用性
- 既存の Win32 バイナリとの互換性
- 既存の Win32 ソースコードとの互換性/移植性
- Microsoft Windows ファミリーと共通の操作性

## ◎ ターゲットプラットフォーム

- x64 – インテル® 64 , AMD AMD64
- IPF – インテル® Itanium® プロセッサ





# Windows 64bit 概要

## ～メモリ空間

一般的なメモリ制限	32bit	64bit
仮想アドレス空間全体	4 GB	16 TB
仮想アドレス空間 (32bit プロセス)	2GB 3GB (*1)	2GB 4GB (*2)
仮想アドレス空間 (64bit プロセス)	N/A	8 TB
ページプール	530 MB	128 GB
非ページプール	256 MB	128 GB
システムキャッシュ	860MB	1 TB

\*1) Boot.ini への /3GB スイッチ + リンカオプション /LARGEADDRESSAWARE

\*2) リンカオプション /LARGEADDRESSAWARE

# Windows 64bit 概要

## ～製品毎のCPU/メモリ空間の制限

物理メモリとCPU制限	32bit	64bit
Windows Vista※	4 GB / 1-2 CPUs	128 GB / 1-2 CPUs
Windows Server 2008 Standard	4 GB / 1-4 CPUs	32 GB / 1-4 CPUs
Windows Server 2008 Enterprise	64 GB / 1-8 CPUs	2 TB / 1-8 CPUs
Windows Server 2003 Datacenter	64 GB / 1-32 CPUs	2 TB / 1-64 CPUs

※ 64bitのHome Premier 16GB / Home Basic 8GB

# Windows 64bit 概要 ～注意点

## ◎ OS のバージョン情報

Windows Server 2008 R2 / Windows 7 x64 Editions → 6.1  
Windows Server 2008 / Windows Vista x64 Editions → 6.0

## ◎ Internet Explorer の User-Agent

IE(WOW64)

Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; WOW64; Trident/4.0)

IE(64bit)

Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Win64; x64; Trident/4.0)

## ◎ その他未サポート機能 (16bit、OS/2、etc...)

# 最新CPUのトレンド

インテルの  
最新64ビット・  
プラットフォーム

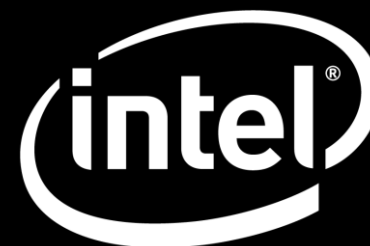
CPUテクノロジー  
トレンド  
－64ビットと  
マルチコア－

インテル株式会社  
清水 良直 様

日本AMD株式会社  
山野 洋幸 様



**Future  
Technology Days**



# インテルの最新64ビット・ プラットフォーム

インテル株式会社  
ソフトウェア技術部  
清水 良直

# インテルサーバー・プロセッサー製品

RISC/メインフレーム  
置き換え



インテル® Itanium® プロセッサー 9000系  
最高水準の柔軟性と信頼性を備えたメインフレーム・クラス・サーバー



大規模データ処理



インテル® Xeon® プロセッサー 7000系  
優れた 性能拡張性 と 信頼性

高密度ITインフラ



インテル® Xeon® プロセッサー 5000系  
優れた 性能密度 と 省電力性



エントリー・クラス



インテル® Xeon® プロセッサー 3000系  
高いコスト・パフォーマンスで基本的なサーバー要件に適合

最先端のテクノロジーと  
インテルのサーバー/ワークステーション向けプロセッサー

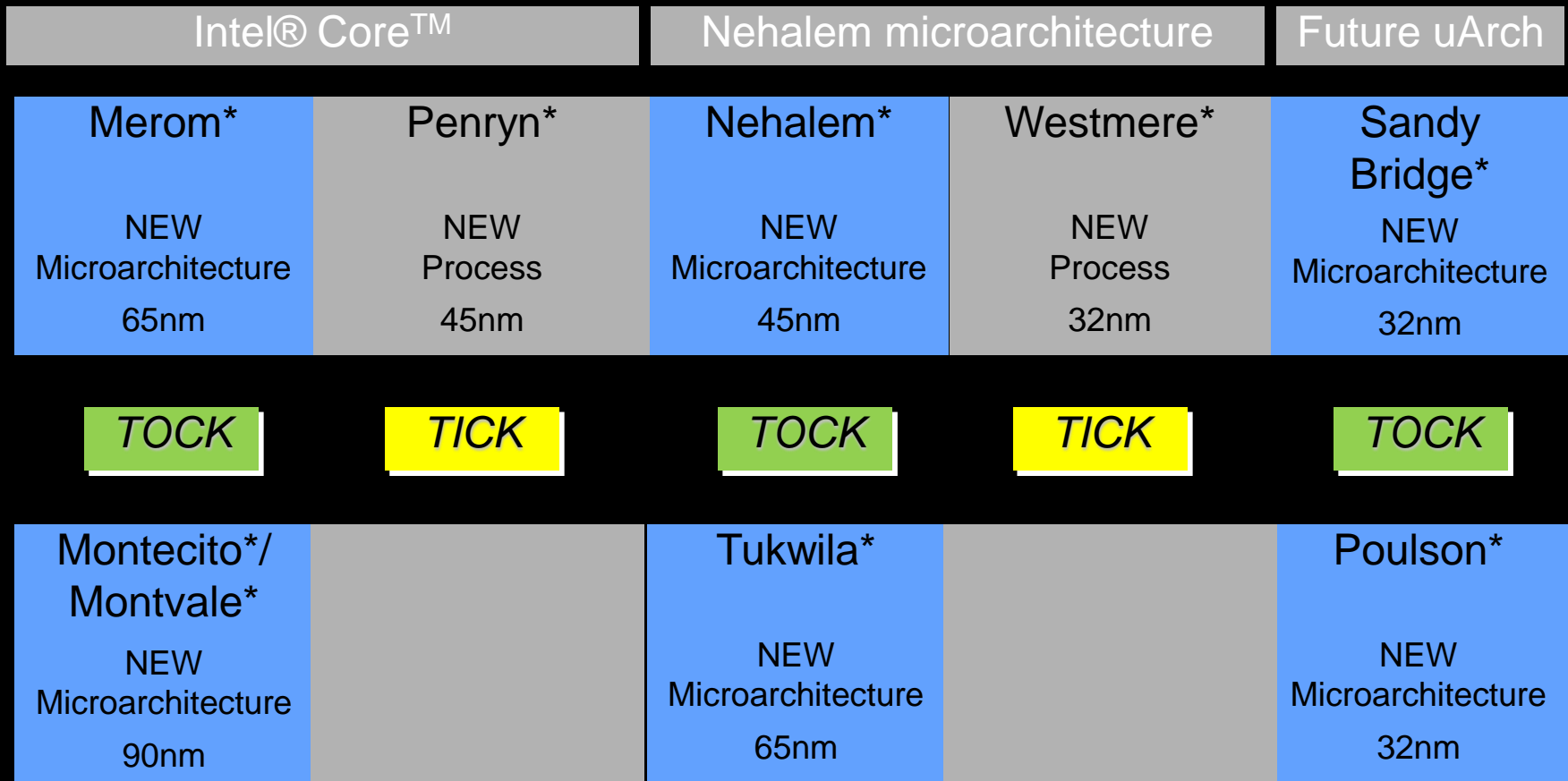


# リーダーシップを持続する製品開発



インテル製品は、予告なく仕様が変更される場合があります。 \*は開発コード名です  
本資料に記載されているすべての製品、日付、および数値は、現在の予想に基づくものです。

# Tick-Tock 開発モデルに基づく サーバー・プロセッサ・ロードマップ



予測

インテル製品は、予告なく仕様が変更される場合があります。 \*は開発コード名です  
本資料に記載されているすべての製品、日付、および数値は、現在の予想に基づくものです。



# 新しいインテル® マイクロアーキテクチャー

(開発コード名: Nehalem)

高い性能と優れた省電力性  
インテル® ハイパースレッディング・  
テクノロジー

インテル® ターボ・ブースト・テクノロジ  
ー

動的電源管理

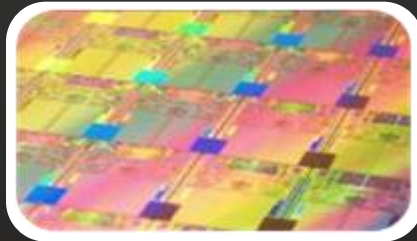
第2世代 インテル®  
バーチャライゼーション・テクノロジー

3倍のメモリー帯域幅  
業界最高水準の帯域幅を備えた  
インテル® QuickPath インターコネクト



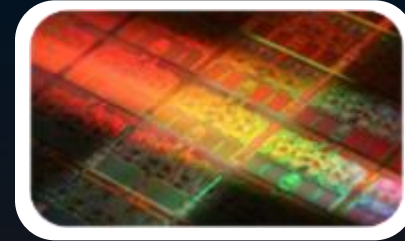
拡張性と省電力を考えた高性能プロセッサ・コア

# プロセッサー・コアと製品群



Penryn\*

45nm shrink, High-k process



Nehalem\*

Up to 8 cores, Integrated Memory Controller, Quick Path Technology, SSE4.2

サーバー

Bensley-Ht\*  
Stoakley-Ht\*  
Available

Dunnington\*  
6-core, SSE4.1

Nehalem-EX\*, Nehalem-EP\*  
Next gen VT & I/OAT, Turbo, DRS

Westmere-EX\*  
Westmere-EP\*

オフィスPC

Weybridge\*  
Available

McCreary\*  
2C/4C, SSE4.1  
VT, TXT

Piketon\*  
4C/8T, SSE4.2  
VT, TXT

ホームPC

Salt  
Creek\*  
Available

Boulder Creek\*  
2C/2T, 4C/4T  
SSE4.1

Tylersburg HEDT\*  
4C/8T  
SSE4.2

Kings Creek\*  
2C / 4T, 4C/8T  
SSE4.2

ノートPC

Santa Rosa\*  
Refresh  
Available

Montevina\*  
2C/2T, 4C/4T  
SSE4.1

Calpella\*  
2C / 4T, 4C/8T  
SSE4.2

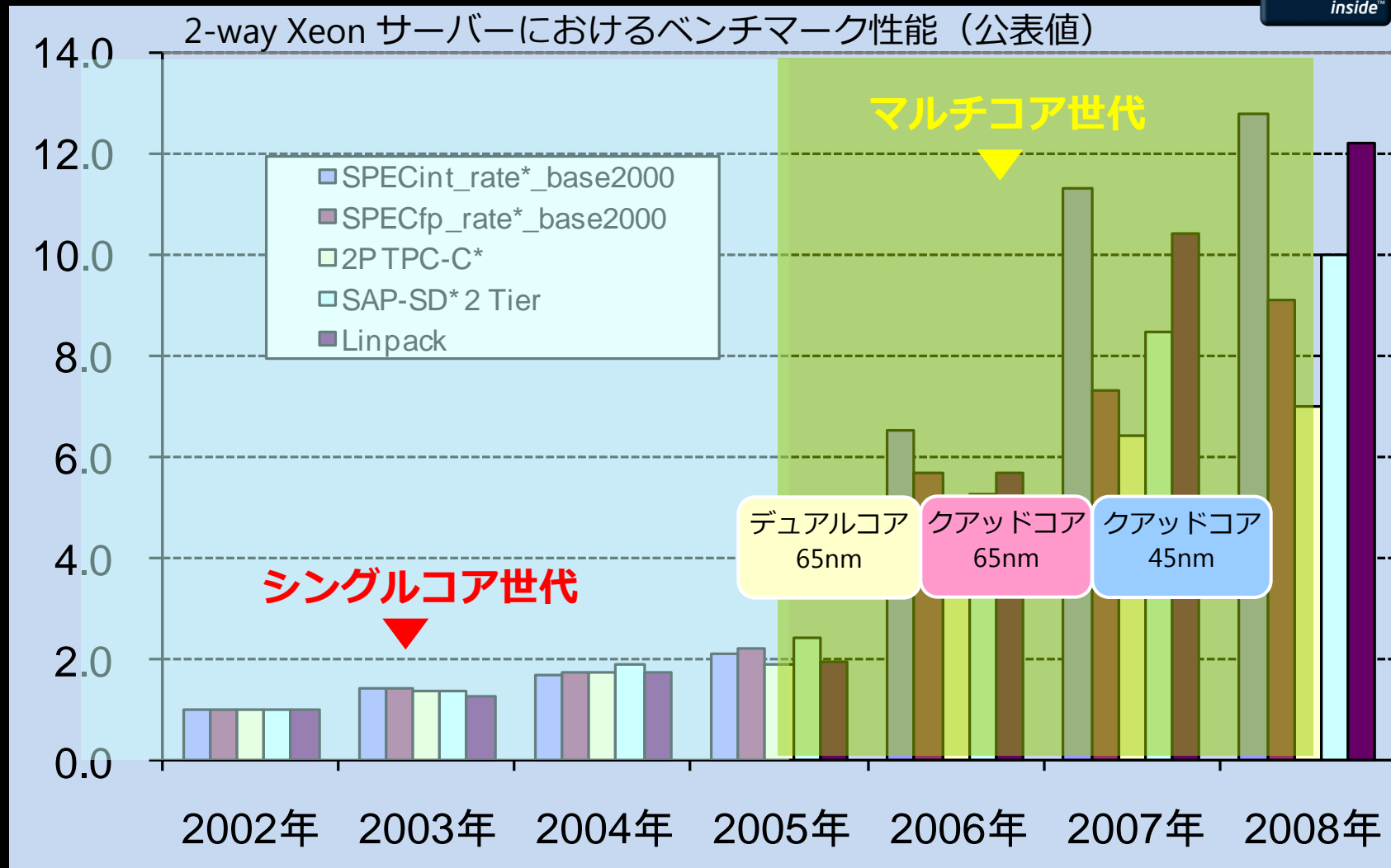
同一のプロセッサー・コアをノートPCからサーバーまで展開

インテル製品は、予告なく仕様が変更される場合があります。 \*は開発コード名です  
本資料に記載されているすべての製品、日付、および数値は、現在の予想に基づくものです。

# マルチコア化による性能向上

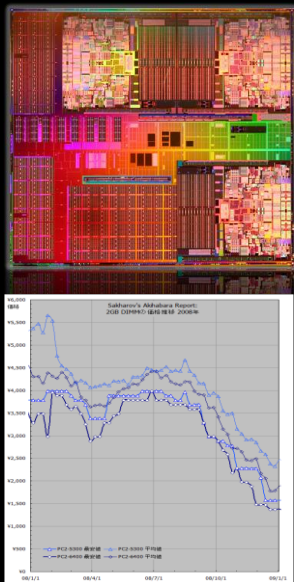


相対性能



性能に関するテストや評価は、特定のコンピューター・システム、コンポーネント、またはそれらを組み合わせて行ったものであり、このテストによるインテル製品の性能の概算の値を表しているものです。システム・ハードウェア、ソフトウェアの設計、構成などの違いにより、実際の性能は掲載された性能テストや評価とは異なる場合があります。システムやコンポーネントの購入を検討される場合は、ほかの情報も参考にして、パフォーマンスを総合的に評価することをお勧めします。インテル製品の性能評価についてさらに詳しい情報をお知りになりたい場合は、<http://www.intel.com/performance/resources/limits.htm>（英語）を参照してください。インテル製品は、予告なく仕様変更される場合があります。詳細は補足資料参照。

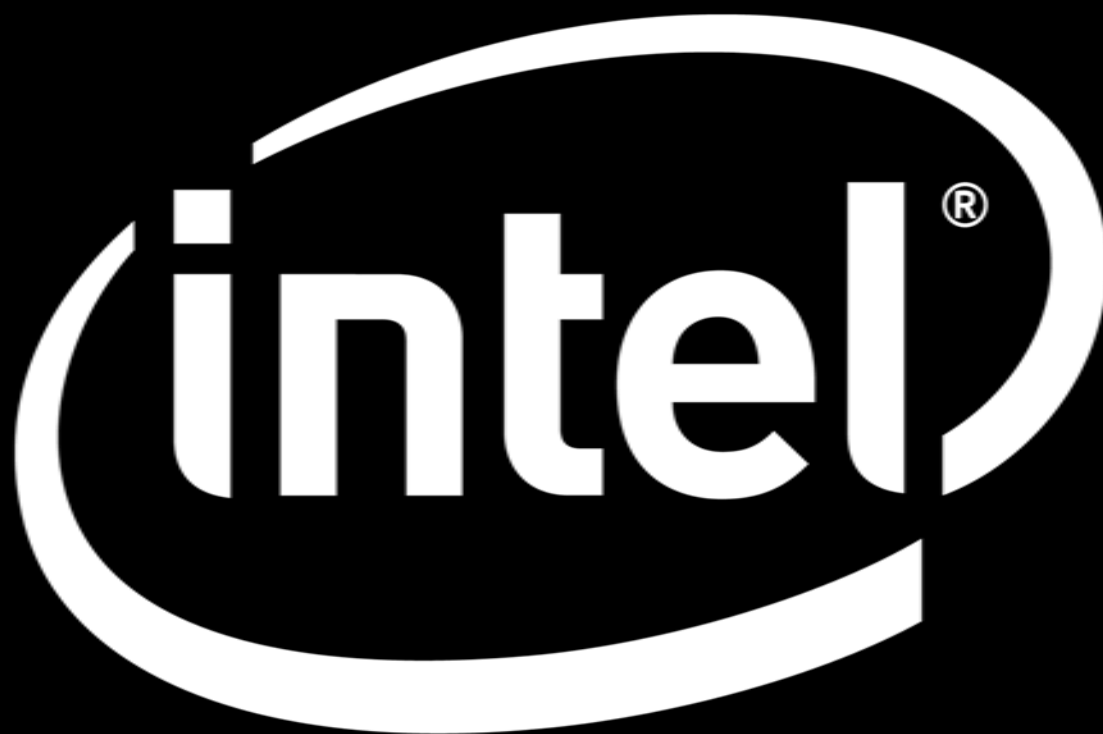
# 64ビット OS、アプリケーションの必要性



- マルチコア・プロセッサの性能を享受する
  - コアあたりのメモリー容量を確保
  - マルチタスクをマルチコアで高速実行
- 64ビットメモリー空間を使用したインメモリー処理
  - 価格が安くなったメモリーのメリットを生かす
  - データ量の増大に応じた高速処理
- 仮想化への対応

インテルのプロセッサー・ファミリーは 64ビット対応済み





# 著作権/法律に基づく表示

- 本資料に掲載されている情報は、インテル製品の概要説明を目的としたものです。製品に付属の売買契約書『Intel's Terms and conditions of Sales』に規定されている場合を除き、インテルはいかなる責を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商品性に関する保証、第三者の特許権、著作権、その他、知的所有権を侵害していないことへの保証を含む) に関しても一切責任を負わないものとします。インテル製品は、医療、救命、延命措置などの目的への使用を前提としたものではありません。
- インテル製品は、予告なく仕様が変更される場合があります。本資料に記載されているすべての製品、日付、および数値は、現在の予想に基づくものであり、計画以外の目的ではご利用になれません。
- 機能や命令の中に「予約済み」または「未定義」と記されているものがありますが、その機能が存在しない状態や何らかの特性を設計の前提にしてはなりません。これらの項目は、インテルが将来のために予約しているものです。インテルが将来これらの項目を定義したことにより、衝突が生じたり互換性が失われたりしても、インテルは一切責任を負わないものとします。
- 本資料に掲載されているインテル製品は、エラッタと呼ばれる設計上の不具合が含まれている可能性があり、公開されている仕様とは異なる動作をする場合があります。現在までに判明しているエラッタの情報については、インテルまでお問い合わせください。
- 本資料には、設計段階にある製品の情報が含まれています。本資料で提供される情報は、予告なしに変更されることがあります。本資料をもとに設計を行わないでください。製品を注文する前に、販売代理店まで最新の仕様をお問い合わせください。
- Intel、インテル、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Atom、Intel Atom Inside、Centrino、Centrino Inside、Centrino ロゴ、Intel Core、Core Inside、Celeron、Celeron Inside、Itanium、Itanium Inside、Xeon、Xeon Inside は、アメリカ合衆国およびその他の国における Intel Corporation の商標です。その他の社名、製品名などは、一般に各社の商標または登録商標です。
- ©2009 Intel Corporation 無断での引用、転載を禁じます。



# 補足資料: マルチコア化による性能向上

Year	Code Name	Brand Name	Launch	Xeon Freq	Cores/ CPU	L1 Cache	L2 Cache	L3 Cache	FSB	Chipset	SPECint_rate*_base2000	SPECfp_rate*_base2000	TPC-C* tpmC	SAP-SD* Users	Linpack GFlops
2001	Tualatin	Intel Pentium III 1.40GHz 512M L2	WW32	1.26	1	16K(I)+16K(D)	512KB	0	133	SW HE-SL	<u>12.2</u>	<u>6.5</u>	<u>17079</u>	<u>135</u>	-
2002	Prestonia	Intel Xeon 2.20 GHz 512M L2	April	2.20	1	12K(I)+8K(D)	512KB	0	400	E7500	15.7	13.1	<u>33768</u>	<u>215</u>	5.46
2002	Prestonia	Intel Xeon 2.80 GHz 512M L2	Sept	2.80	1	12K(I)+8K(D)	512KB	0	400	E7500	18.9	14.4	<u>39007</u>	<u>245</u>	6.72
2003	Prestonia	Intel Xeon 3.06 GHz 512M L2	March	3.06	1	12K(I)+8K(D)	512KB	0	533	E7501	<u>21.5</u>	<u>16.7</u>	<u>44943</u>	<u>290</u>	7.83
2003	Prestonia	Intel Xeon 3.06 GHz 1M L3	July	3.06	1	12K(I)+8K(D)	512KB	1MB	533	E7501	<u>26.0</u>	<u>20.4</u>	<u>52468</u>	<u>329</u>	8.16
2003	Prestonia	Intel Xeon 3.20 GHz 1M L3	Oct	3.20	1	12K(I)+8K(D)	512KB	1MB	533	E7501	<u>26.9</u>	<u>20.7</u>	<u>54097</u>	340	8.40
2004	Prestonia	Intel Xeon 3.20 GHz 2M L3	March	3.20	1	12K(I)+8K(D)	512KB	2MB	533	E7501	<u>30.7</u>	<u>24.0</u>	<u>60364</u>	<u>408</u>	9.79
2004	Nocona	Intel Xeon 3.60 GHz 1M L2	June	3.60	1	16K(I)+16K(D)	1MB	0	800	E7520	<u>32.1</u>	<u>25.2</u>	<u>68010</u>	<u>470</u>	11.60
2005	Irwindale	64-bit Intel Xeon 3.60 GHz 2M L2	Feb	3.60	1	12K(I)+16K(D)	2MB	0	800	E7520	<u>39.9</u>	<u>31.6</u>	<u>74298</u>	<u>597</u>	13.00
2005	Paxville DP	Dual-Core Intel Xeon 2.80 2x2M L2	Oct	2.80	2	12K(I)+16K(D)	2x2MB	0	800	E7520	<u>59.5</u>	<u>39.4</u>	87791	<u>788</u>	18.10
2006	Dempsey	Dual-Core Intel Xeon 5080	May	3.73	2	12K(I)+16K(D)	2x2MB	0	1066	5000p	<u>81.2</u>	<u>61.6</u>	<u>125954</u>	<u>1047</u>	22.10
2006	Woodcrest	Dual-Core Intel Xeon 5160	June	3.00	2	32K(I)+32K(D)	4MB	0	1333	5000p	<u>123.0</u>	<u>82.1</u>	<u>169360</u>	<u>1285</u>	38.30
2006	Clovertown	Quad-Core Intel Xeon X5355	Nov	2.66	4	32K(I)+32K(D)	2x4MB	0	1333	5000p	<u>200.0</u>	103.0	<u>240737</u>	<u>1870</u>	64.00
2007	Clovertown	Quad-Core Intel Xeon X5365	July	3.00	4	32K(I)+32K(D)	2x4MB	0	1333	5000p	214	105	251300	2080	70.00
2007	Harpertown	Quad-Core Intel Xeon X5460	Nov	3.16	4	32K(I)+32K(D)	2x6MB	0	1333	5000p	242	131	273666	2449	82

## Intel Pentium® III Microarchitecture

## Intel NetBurst® Microarchitecture

## Intel Core® Microarchitecture

### Note

Results are Published/Measured/Submitted/Approved as of Nov 12, 2007.

- Only selected data used in the charts
- Additional data provided in the table above for reference
- Information in **Bold** indicates major changes from one generation to other
- Intel launch dates and details at <http://www.intel.com/pressroom/kits/quickreffam.htm>

性能に関するテストや評価は、特定のコンピューター・システム、コンポーネント、またはそれらを組み合わせて行ったものであり、このテストによるインテル製品の性能の概算の値を表しているものです。システム・ハードウェア、ソフトウェアの設計、構成などの違いにより、実際の性能は掲載された性能テストや評価とは異なる場合があります。システムやコンポーネントの購入を検討される場合は、ほかの情報も参考にして、パフォーマンスを総合的に評価することをお勧めします。インテル製品の性能評価についてさらに詳しい情報をお知りになりたい場合は、<http://www.intel.com/performance/resources/limits.htm>（英語）を参照してください。



**Future  
Technology Days**

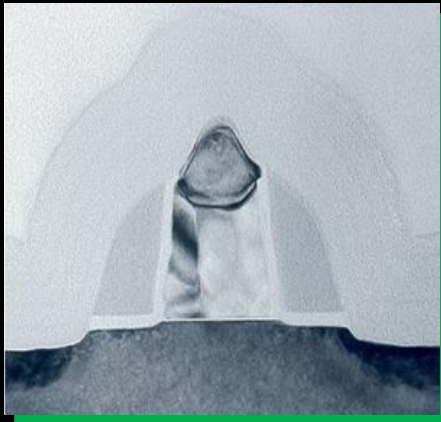
# CPUテクノロジートレンド －64ビットとマルチコア－

日本AMD株式会社  
マーケティング&ビジネス開発本部  
エンタープライズプロダクトマーケティング部  
山野 洋幸



# プロセステクノロジーの進化

Lg = 50 nm



90 nm generation

(2004)

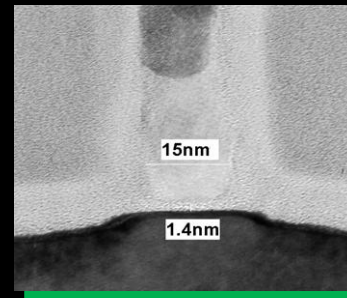
Lg = 35 nm



65 nm generation

(2005)

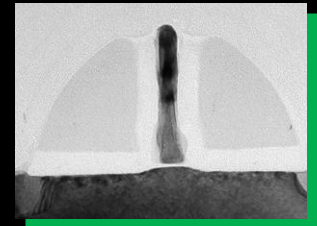
Lg = 15 nm



45 nm generation

(2007)

Lg = 13 nm



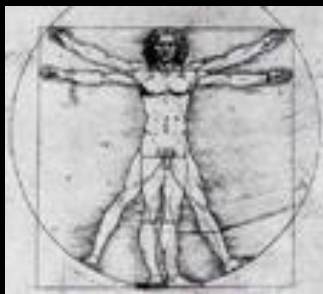
32 nm generation

(2009)

22 nm generation

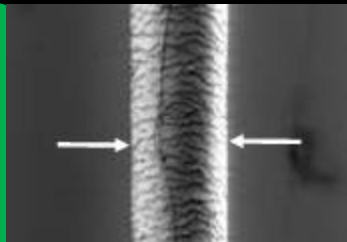
(2011)

# ミクロンからナノメートルへ

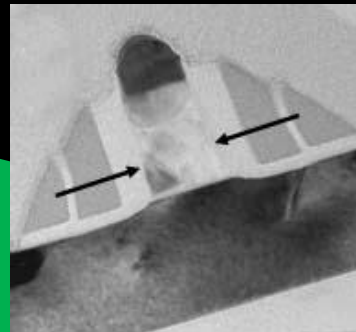


Man: ~2 m  
= 2,000,000,000 nm

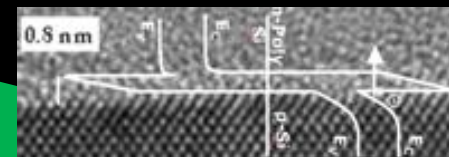
Human hair:  
60  $\mu\text{m}$  = 60,000 nm



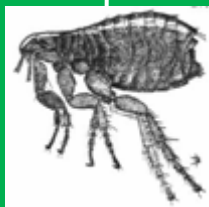
Transistor Gate: 40 nm



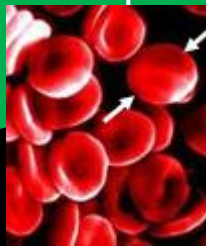
High-performance  
gateoxide: 0.8 nm



Flea: ~1 mm  
= 1,000,000 nm



Bloodcell:  
7.5  $\mu\text{m}$  = 7,500 nm



DNA: 3.4 nm



Si-atom: 0.22 nm



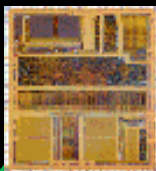
# AMD プロセッサの進化

**1991**



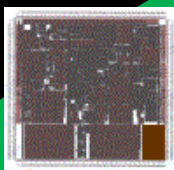
**Am386**  
275K Transistors  
40MHz  
800nm process

**1993**



**Am486**  
1.2M Transistors  
120MHz  
500nm process

**1997**



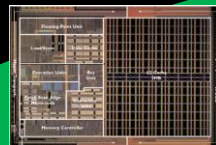
**K6**  
8.8M Transistors  
550MHz  
350-250nm process

**1999**



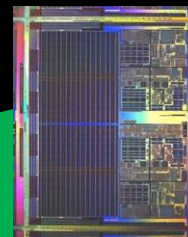
**K7 Athlon**  
22M Transistors  
1.5MHz  
250-180nm process

**2003**



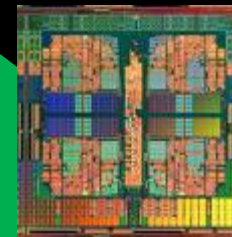
**K8 Opteron/Athlon64**  
110M Transistors  
2.5GHz  
130/90nm process

**2005**



**K8 Athlon64 Dual Core**  
230M Transistors  
3GHz  
90/65nm process

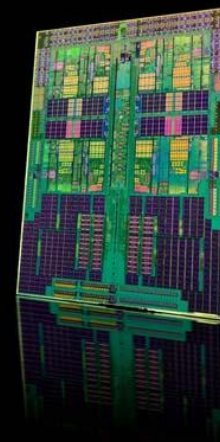
**2007**



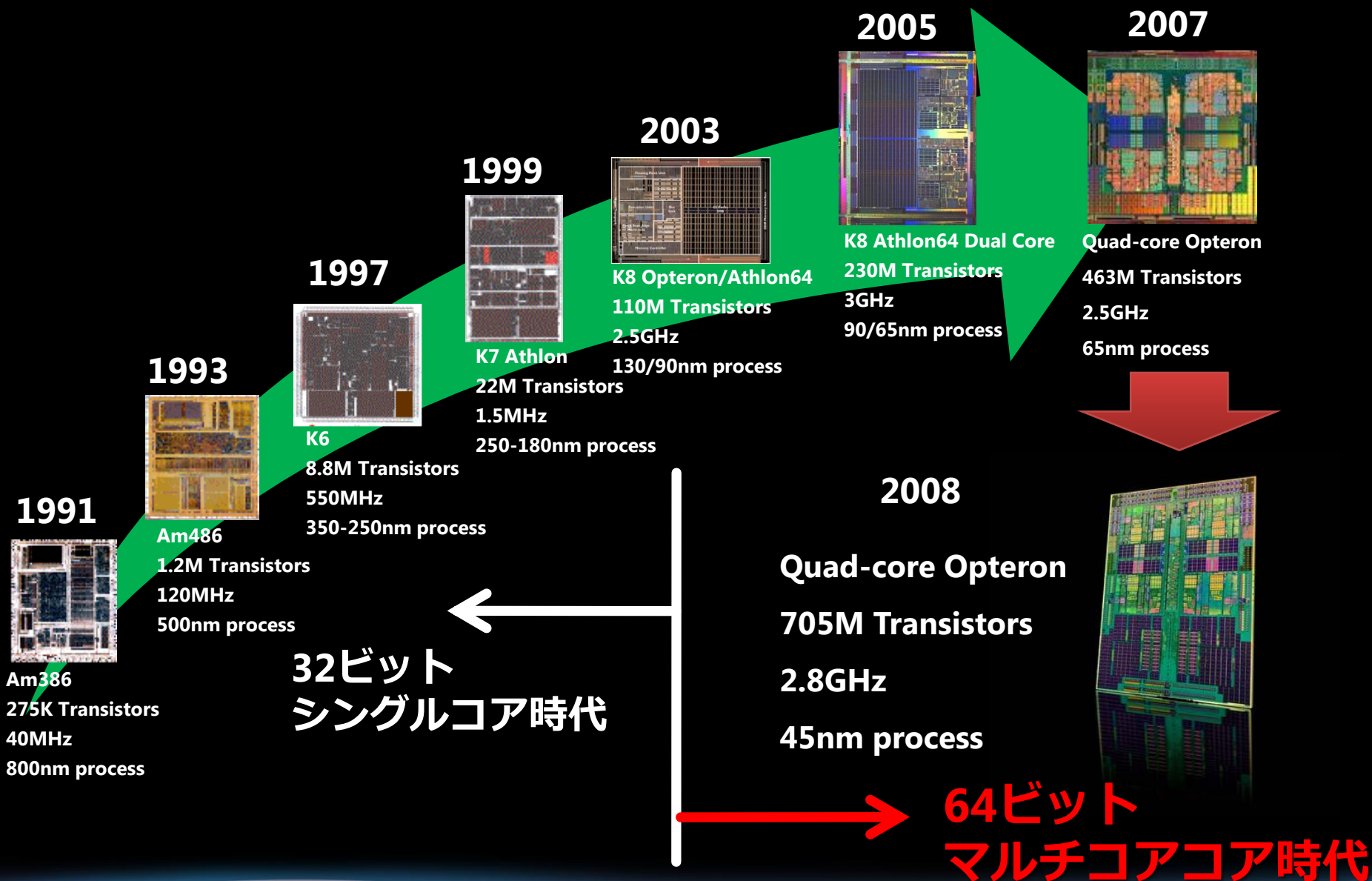
**Quad-core Opteron**  
463M Transistors  
2.5GHz  
65nm process

**2008**

**Quad-core Opteron**  
705M Transistors  
2.8GHz  
45nm process



# AMD プロセッサの進化





# CPUおよびPCサーバープラットフォーム に求められる要件

- コア数増加によって、パフォーマンスがリニアに伸びる  
マルチコアアーキテクチャ
- クラウドコンピューティングに対応した、クラスターおよび  
グリッド環境でのスケーラビリティ
- 仮想化環境での性能、スケーラビリティ
- ワット性能、省電力機能

ご静聴ありがとうございました

**hiroyuki.yamano@amd.com**

#### **Trademark Attribution**

(c) 2009 Advanced Micro Devices, Inc., AMD, the AMD Arrow logo, and combinations thereof, are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names used in this presentation are for identification purposes only and may be trademarks of their respective owners.

©2009 Advanced Micro Devices, Inc. All rights reserved.

64 bit の世界は  
既に皆様のお手元にあります

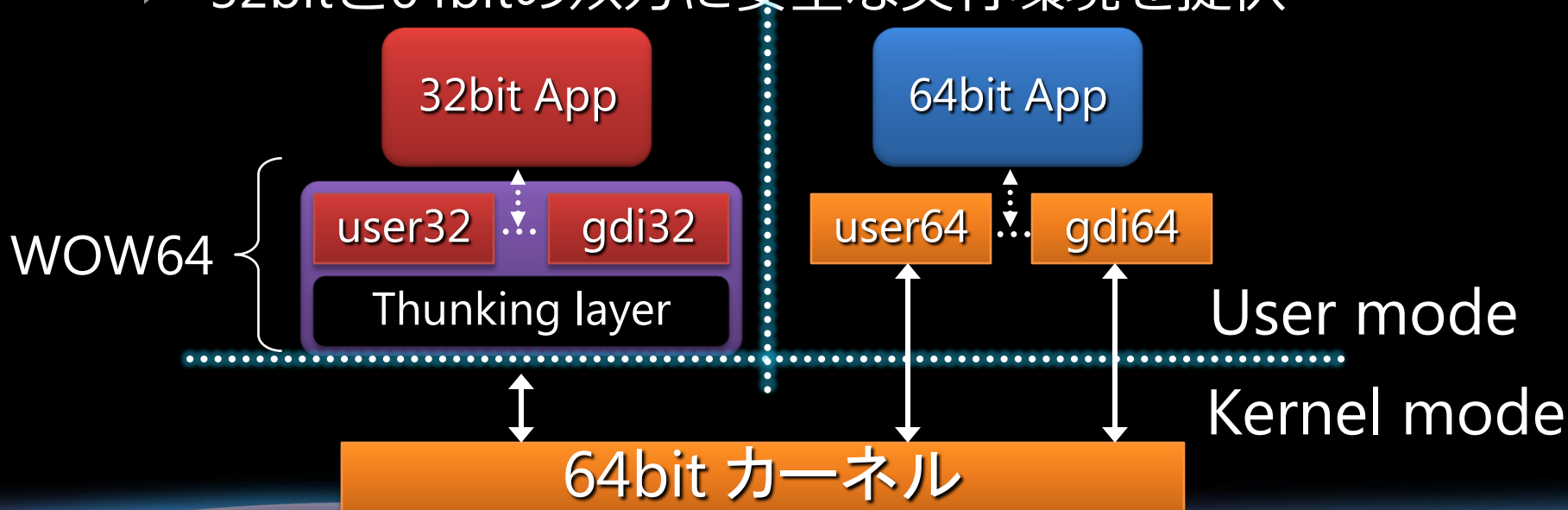


# 64 bit Windows における アプリケーションの動作環境



# アプリケーションの動作環境

- ◎ 64bitアプリケーションは64bitシステムファイルを直接利用する
- ◎ 32bitアプリケーションは Windows on Windows 64 (WOW64) を経由して 64bitシステムファイルを利用する
  - 32bitと64bitの双方に安全な実行環境を提供





*Demo*

Naitive と WOW 64 の動作



# Windows on Windows 64

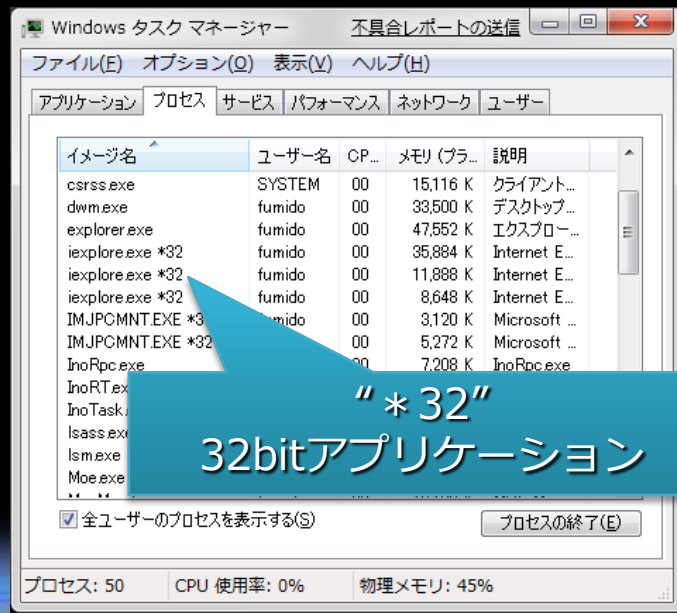
## WOW64

### ◎ WOW64

- Win32 x86 アプリケーションのバイナリーをそのまま動作させるためのエミュレーションレイヤである
- “\*32” が付与されたものが32bitアプリケーション
- WOW64 上での制御のため API が追加されている

### ◎ 主な機能

- システム コールのインターセプト
- コールバックのインターセプト
- 例外のディスパッチ
- レジストリ管理
  - レジストリ リダイレクター
  - レジストリ リフレクター
- システムファイル管理
  - ファイル システム リダイレクター



# WOW64の環境

◎ 32bit、64bitシステムを分離し、アプリケーションの共存と連携を実現

- システムファイル、プログラムファイル、レジストリ、環境変数



HKLM¥Software¥  
Wow6432Node

%windir%  
¥SysWow64

変数	値
ProgramFiles	X:¥Program Files (x86)
ProgramFiles(x86)	X:¥Program Files (x86)
ProgramW6432	X:¥Program Files
CommonProgramFiles	X:¥Program Files (x86)¥Common Files
CommonProgramFiles(x86)	X:¥Program Files (x86)¥Common Files
CommonProgramW6432	X:¥Program Files¥Common Files
PROCESSOR_ARCHITECTURE	X86
PROCESSOR_ARCHITEW6432	AMD64



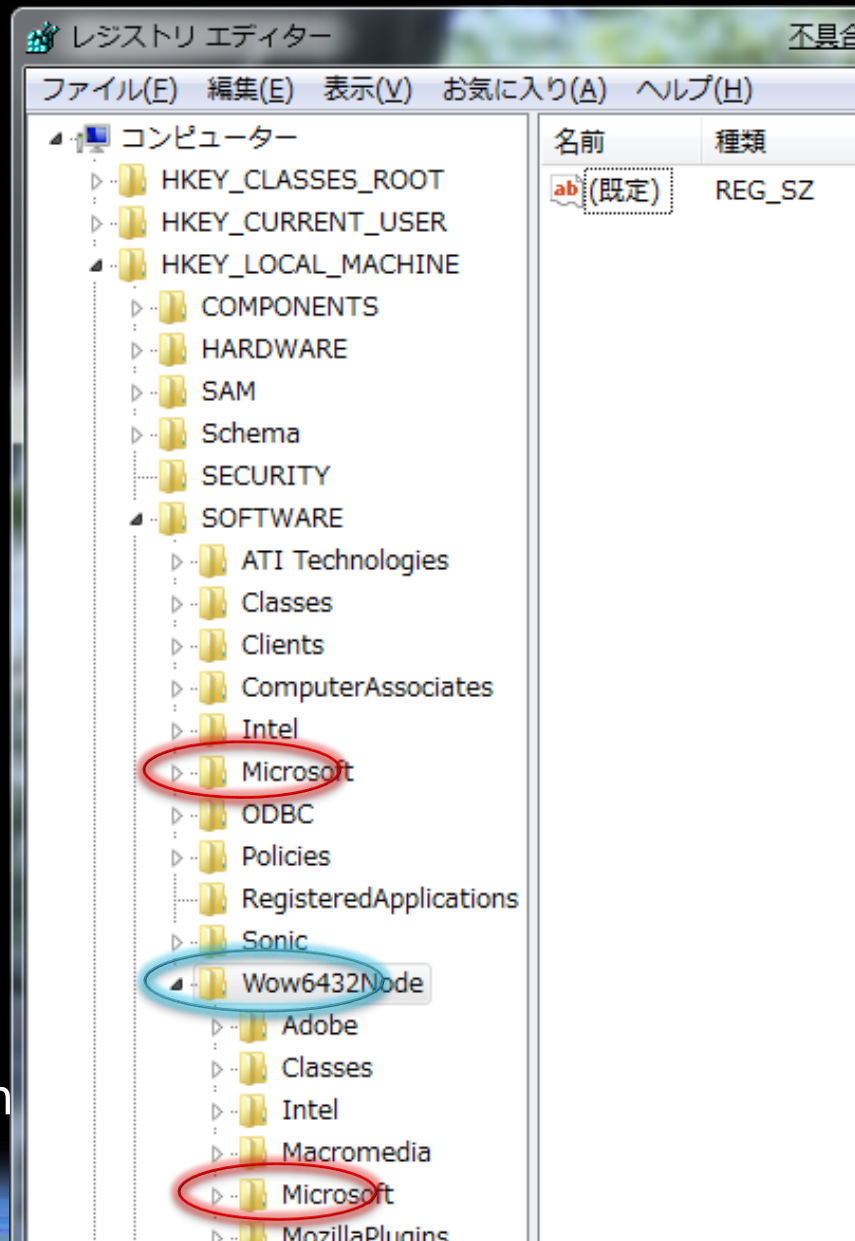
HKLM¥Software

%windir%  
¥System32

変数	値
ProgramFiles	X:¥Program Files
ProgramFiles(x86)	X:¥Program Files (x86)
CommonProgramFiles	X:¥Program Files¥Common Files
CommonProgramFiles(x86)	X:¥Program Files (x86)¥Common Files
PROCESSOR_ARCHITECTURE	AMD64

# WOW64の仕組み

- 64bit アプリケーションと WOW64 プロセスは 別々のレジストリビューを利用
  - WOW64 のための代替ビュー
- WOW64 プロセスがキーを オープン/作成したとき、 WOW6432Node が追加される
  - HKLM¥Software
  - HKCR
- システムファイルも同様に 別々のフォルダが用意される
  - 64bit : "%windir%¥system32"
  - WOW64 : "%windir%¥syswow64"
- 32bit レジストリ エディタの起動
  - %windir%¥syswow64¥regedit.exe -m



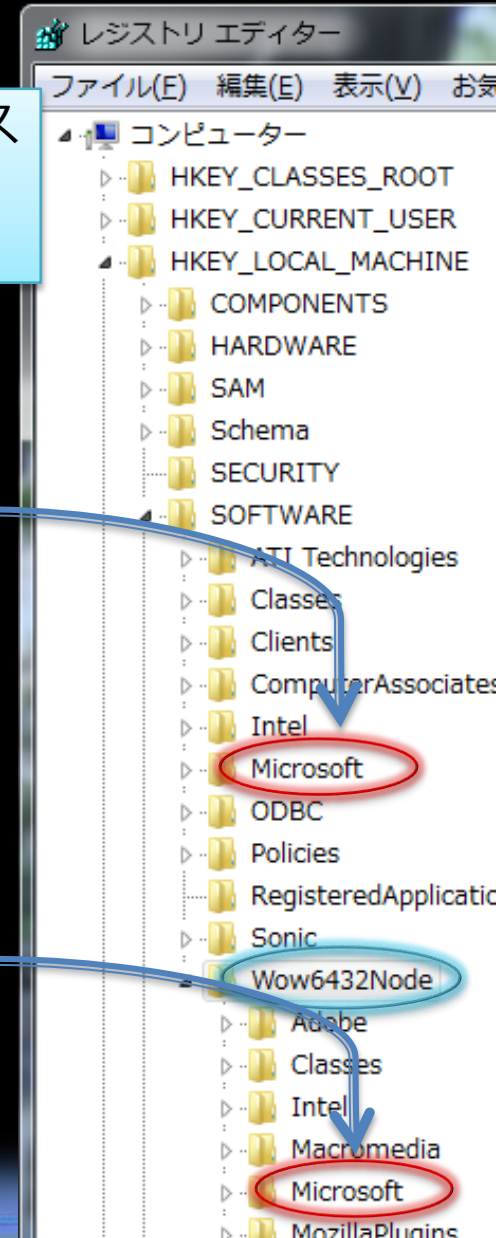
# レジストリ リダイレクター

- WOW64からレジストリへのアクセスはWOW6432Nodeにリダイレクトされる
  - HKEY\_LOCAL\_MACHINE¥Software
  - HKEY\_USERS¥\*¥Software¥Classes
  - HKEY\_USERS¥\*\_Classes※ \*はユーザーのSIDを示す
- RegOpenKeyEx()、RegCreateKeyEx()の変更は不要
- 32bit、64bit キーへの明示的なアクセスも可能
  - KEY\_WOW64\_64KEY :
    - 32bitおよび64bitアプリケーションによって64bitキーを確認する場合に使用
  - KEY\_WOW64\_32KEY :
    - 32bitおよび64bitアプリケーションによって32bitキーを確認する場合に使用

レジストリアクセス  
HKLM¥Software¥  
microsoft

64bit App

32bit App





# レジストリ リフレクター

- 二つのレジストリ ビュー間で、特定のレジストリ キーおよび値をリアルタイムで複製する
  - COM 等を利用した 64/32 bitアプリケーションの相互運用を実現するために実装された機能である
- レフレクト対象キー
  - HKEY\_LOCAL\_MACHINE¥Software¥Classes
  - HKEY\_LOCAL\_MACHINE¥Software¥Microsoft¥COM3
  - HKEY\_LOCAL\_MACHINE¥Software¥Microsoft¥EventSystem
  - HKEY\_LOCAL\_MACHINE¥Software¥Microsoft¥Ole
  - HKEY\_LOCAL\_MACHINE¥Software¥Microsoft¥Rpc
  - HKEY\_USERS¥\*¥Software¥Classes
  - HKEY\_USERS¥\*\_Classes

※ \*はユーザーのSIDを示す
- HKEY\_CLASSES\_ROOT¥CLSID 複製のルール
  - 64/32 bit のプロセス間呼び出しを可能にするために
    - LocalServer32 の CLSID は複製される
  - プロセス内では 64/32 bit の相互運用はできないため
    - InProcServer32 と InProcHandler32 は複製されない

# ファイルシステムリダイレクター

- 32bitプロセスからファイルレベルアクセスを  
"%windir%¥system32" から"%windir%¥syswow64"へ  
自動的にリダイレクションする
  - %windir%¥system32 がハードコードしている場合は問題となる
- 下記のディレクトリはリダイレクトの対象外となる
  - %windir%¥system32¥catroot
  - %windir%¥system32¥catroot2
  - %windir%¥system32¥drivers¥etc
  - %windir%¥system32¥logfiles
  - %windir%¥system32¥spool
- 32bit アプリケーション用のリダイレクト制御方法
  - Wow64EnableWow64FsRedirection() - FALSE/TRUE を使い制御
  - Wow64DisableWow64FsRedirection() – Revert と組合せて制御
  - Wow64RevertWow64FsRedirection()
- 64bitアプリケーション用の WOW64 フォルダの確認方法
  - GetSystemWOW64Directory()





*Demo*

リダイレクターの動作

# 動作環境のポイント

✓ 64bit OSのメリットを享受できる

- カーネルハンドルや Windows ハンドルが多くなる
- ただし、スレッドは64bitスタックを持つことになる

✓ アプリケーションのアドレス空間は2Gバイトとなる

- Large-Address-Aware でアプリケーションがビルドされると4Gバイトのアドレス空間となる

⊘ プロセス内で 64/32bit 相互運用はできない


- WOW64プロセスは64bit DLLをロードできない
- 異なるbitのLoadLibrary()は失敗する

⚠ Windows Server 2008 R2 CoreではWOW64が既定で無効となっている

# 動作環境のポイント - 2

 Active X や プラグイン を動作させる

- 32bit の IE を利用する
  - 相互運用の問題のため

 サードパーティ製 IMEを利用している

- 64bit版 IMEが提供されない場合は  
64bit アプリケーションから  
サードパーティ製 IMEが利用できない
  - MS-IMEは64/32bitが提供されている



# 64bit Windows への アプリケーション移行

# 動作環境を決める - Part 1

## .NET Framework ベース アプリケーション ※ 2.0 以降

64bit  
.NET

64bitで動作させる

- ・ 移行作業なし
- ・ ターゲット(AnyCPU, X64)

32bit  
.NET

32bitで動作させる

- ・ 移行作業なし
- ・ ターゲット(x86)

## C++のNative アプリケーション

64bit  
Native

64bitで動作させる

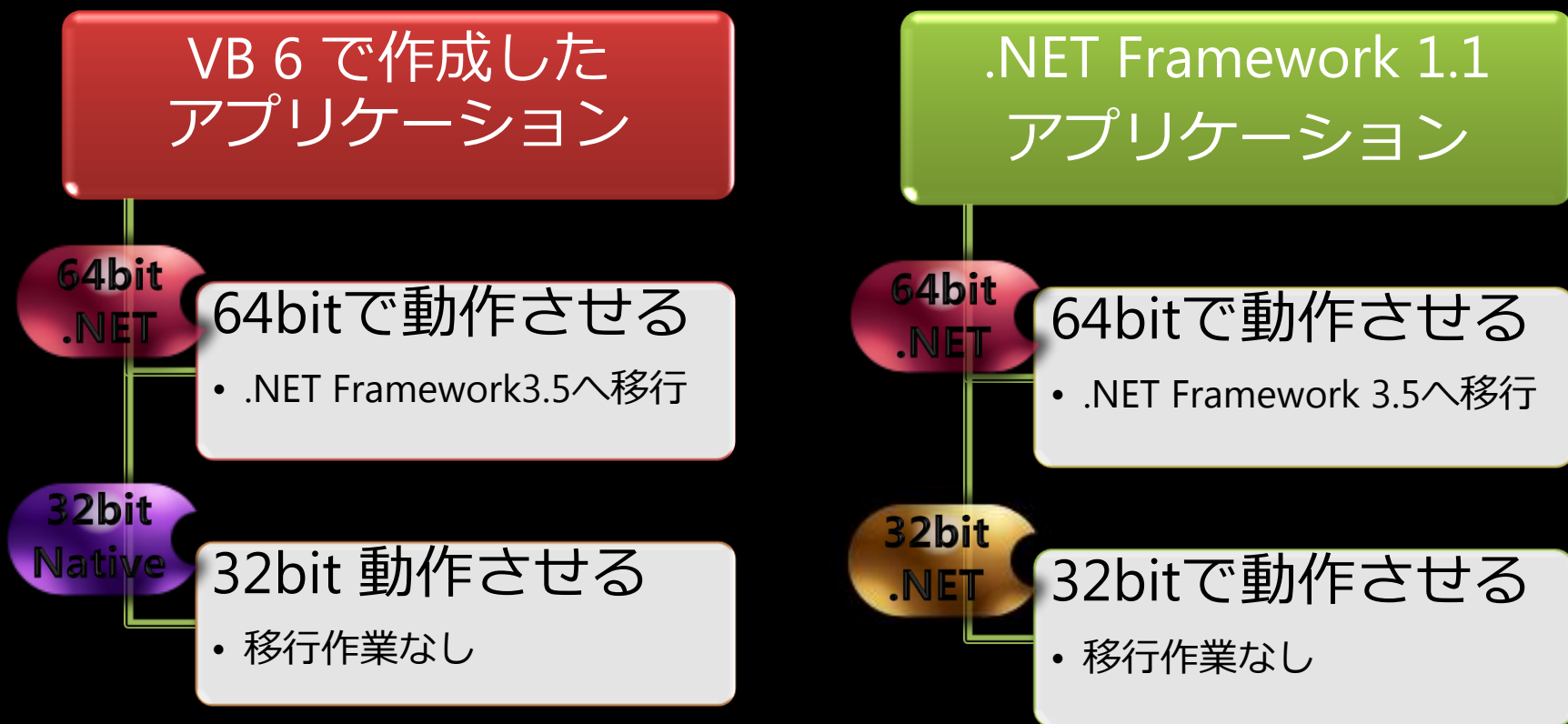
- ・ Win64へ移行作業が必要

32bit  
Native

32bit 動作させる

- ・ 移行作業なし

# 動作環境を決める – Part 2



※ Windows 7 と Windows Server R2 でのランタイムのサポートについて、製品リリースまでに情報を提供いたします。

VB6の情報 : <http://msdn.microsoft.com/en-us/vbrun/ms788708.aspx>

.NET Framework 1.1についてはサポートライフサイクル情報をご確認ください



# (参考) ドライバーアプリケーション

- 64bit への移行が必須となる
- カーネルモードは証明書が必須となる
- リソース

Designing for 64bit Windows

<http://www.microsoft.com/whdc/system/platform/64bit/64bitsystems.mspx>

32 ビットおよび 64 ビット プラットフォーム用の  
ドライバのインストール

[http://www.microsoft.com/japan/whdc/driver/install/32-64bit\\_install.mspx](http://www.microsoft.com/japan/whdc/driver/install/32-64bit_install.mspx)

# 64bit アプリケーション開発 ～32bitとの違い

64bit  
Native

## ◎ Windows プログラミング

- Win32 との同一API による単一のソースコード
- 64bit用に拡張 (ポインタ、データモデル、API)
  - ・ LONG32,INT32,DWORD32,LONG64,INT64,DWORD64
  - ・ INT\_PTR, LONG\_PTR,SSIZE\_T,UINT\_PTR,ULONG\_PTR,DWORD\_PTR

## ◎ ポインタのサイズ

- 32bit( $2^{32}$ )から64bit( $2^{64}$ )へ : 4バイト → 8バイト

API	データモデル	int	long	Pointer
Win32	ILP32	32	32	32
Win64	IL32P64	32	32	64

# 64bit アプリケーション開発 ～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 32bitコードと64bitコードは同居できない

- ＞ ライブラリロード時にエラーが発生する
- ＞ 該当するケース
  - ＞ 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



64bit  
.NET

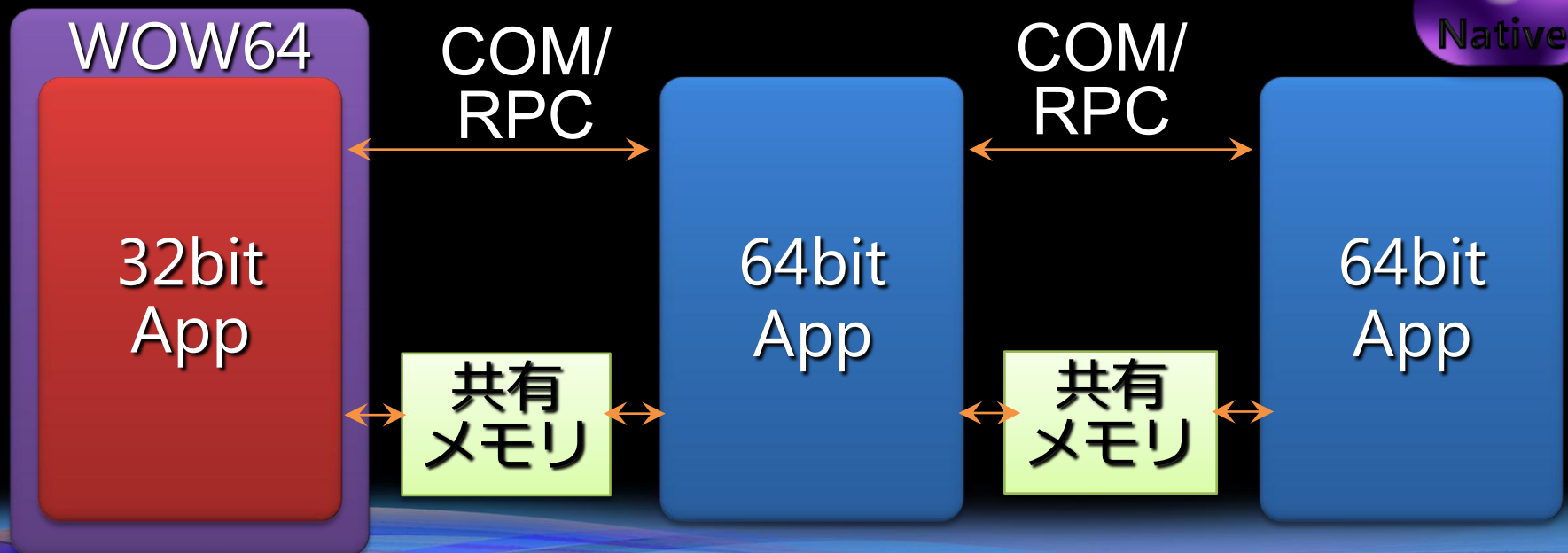
32bit  
.NET

64bit  
Native

32bit  
Native

# 64bit アプリケーション開発 ～64/32bit 相互運用における留意点

- ◎ アウトプロセス COM/RPC (EXE サーバ) により相互運用をサポートする
- ◎ 共有メモリ等のオブジェクトを共有する
  - 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



64bit  
.NET

32bit  
.NET

64bit  
Native

32bit  
Native

# 開発環境

- Visual Studio 2008

## ◎ 32ビットアプリケーション

- 32ビット Windows、WOW64 上での開発をサポート

## ◎ ローカル、リモートでのデバッグ

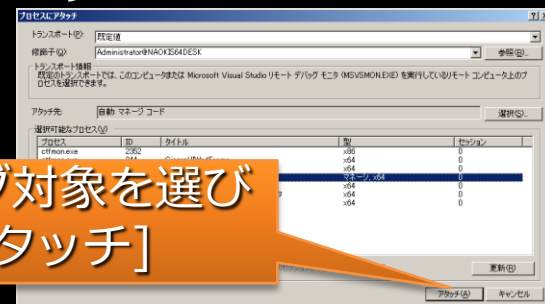
- WOW64 上の 32ビットプロセス
- 64ビットプロセス
- x64およびIPF

64bit  
.NET

64bit  
Native

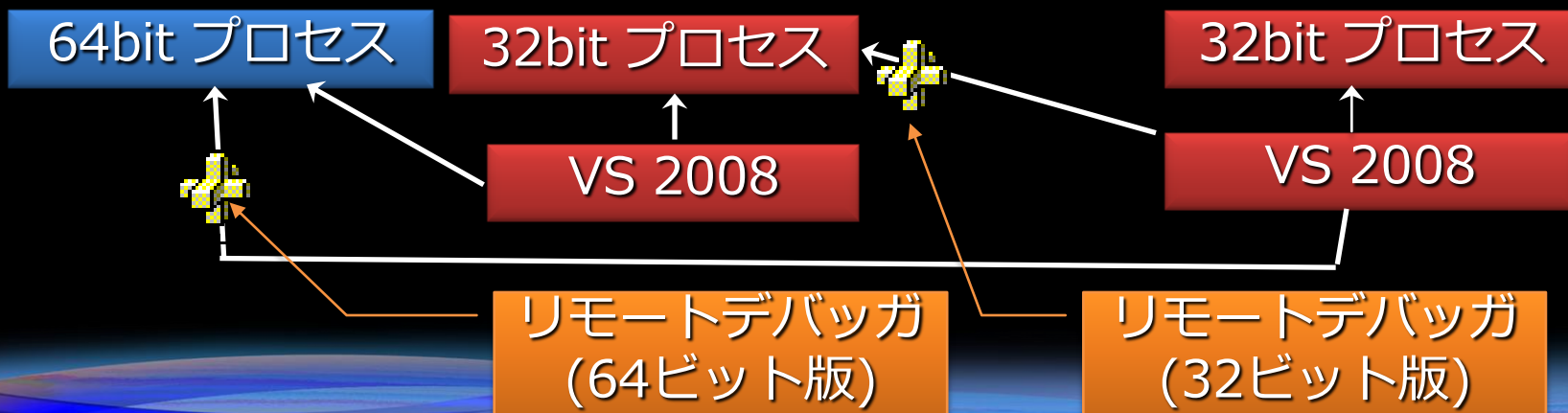
32bit  
.NET

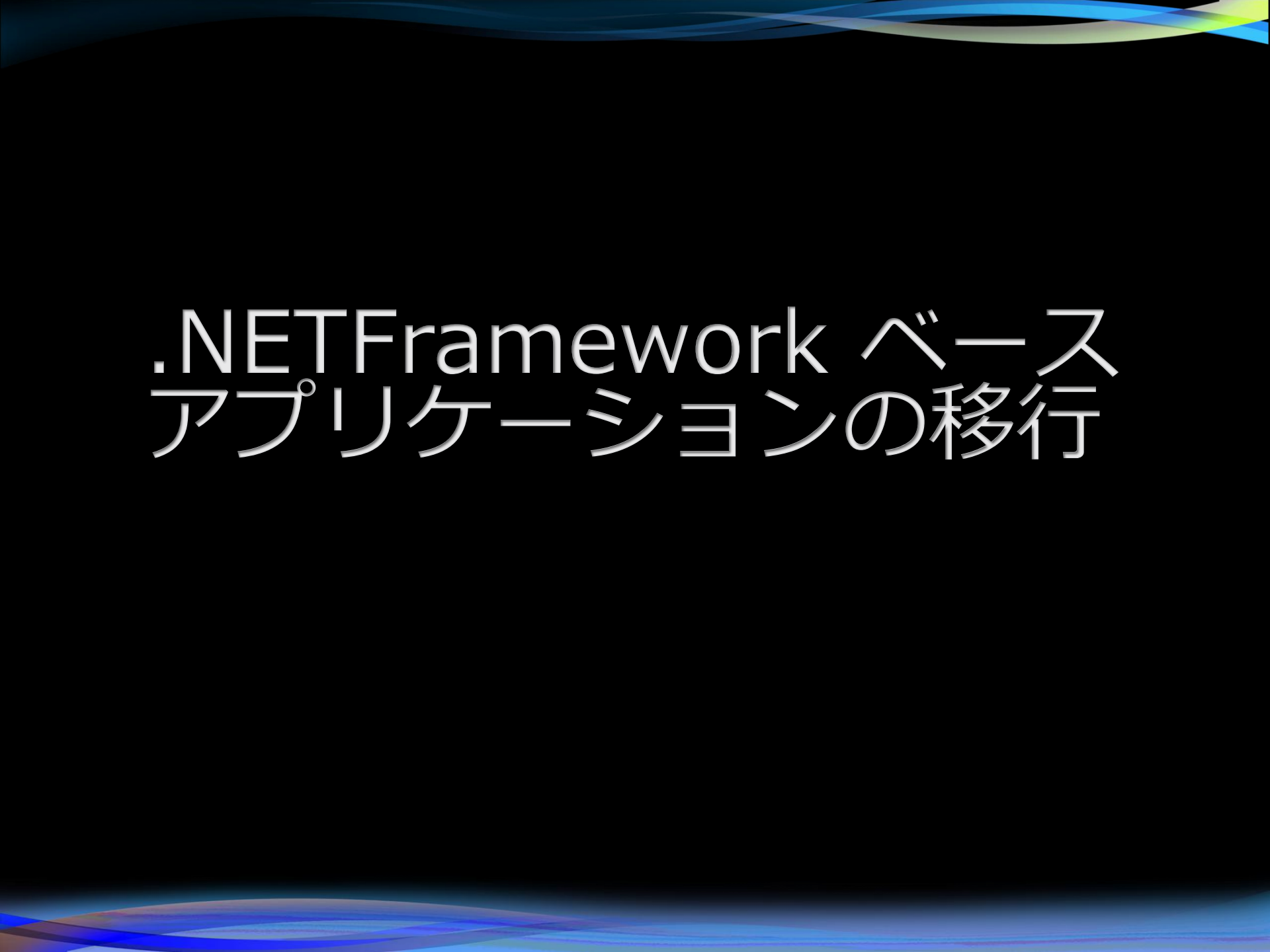
32bit  
Native



Windows 64bit

Windows 32bit





# .NETFramework ベース アプリケーションの移行



# 64 ビット環境の .NET Framework

64bit  
.NET

32bit  
.NET

## ◎ .NET Framework 3.5\*

- 2 つの 64 bit CPU アーキテクチャ
  - ・ x64 : インテル® 64, AMD AMD64,
  - ・ IPF : インテル® Itanium® プロセッサ
- WOW64 (32bit)、32bitCPUもサポート
  - ・ WOW64 : 64bit 向け再頒布可能パッケージ
  - ・ 32bit : x86向け再頒布可能パッケージ

※ .NET Framework 3.5 は.NET Framework 3.0と2.0を包括しています。  
ここから.NET Framework 3.5 とした場合は3.0と2.0が含まれていることとします。

# .NET Framework の動作

- 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
  - 32bit 環境 (x86, WOW64)
  - 64bit 環境 ( x64 およびIPF 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 コンパイルにより最適なネイティブコードで実行可能

64bit  
.NET

32bit  
.NET

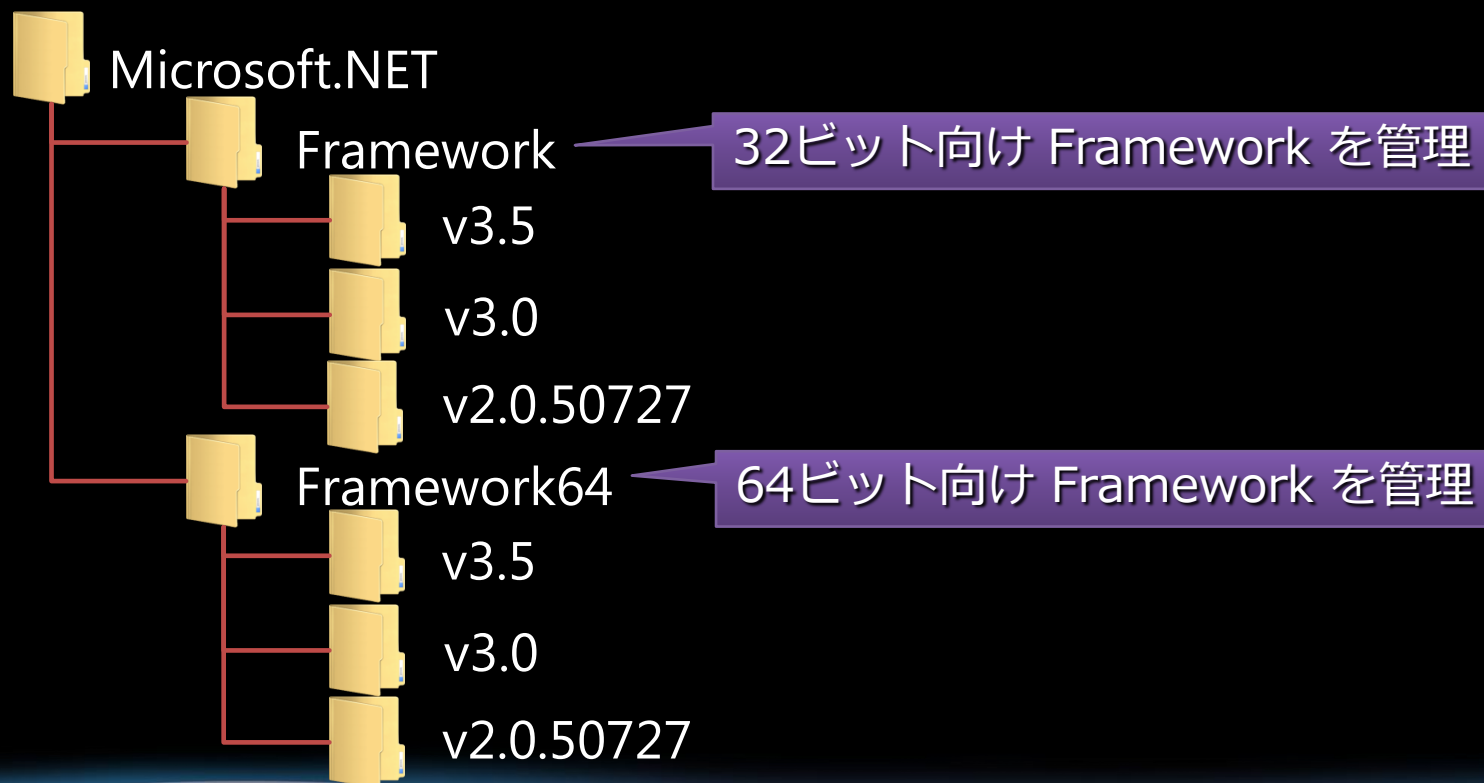


# .NET Framework の動作環境

64bit  
.NET

32bit  
.NET

- ◎ WOW64 上で .NET Framework を動作
  - > .NET Framework (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



# ターゲットプラットホーム

64bit  
.NET

64bit  
Native

32bit  
.NET

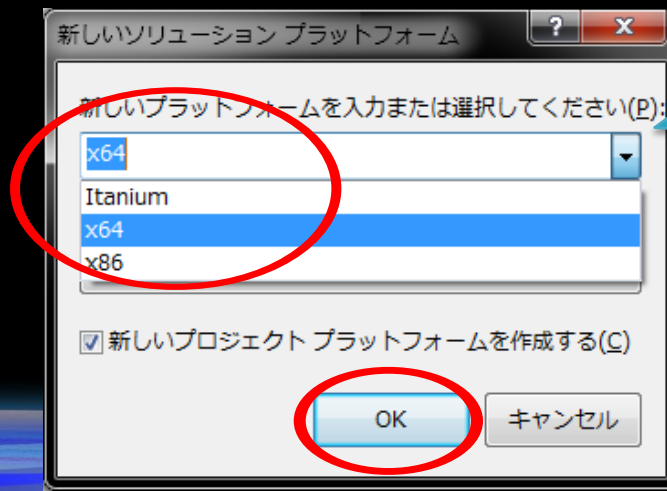
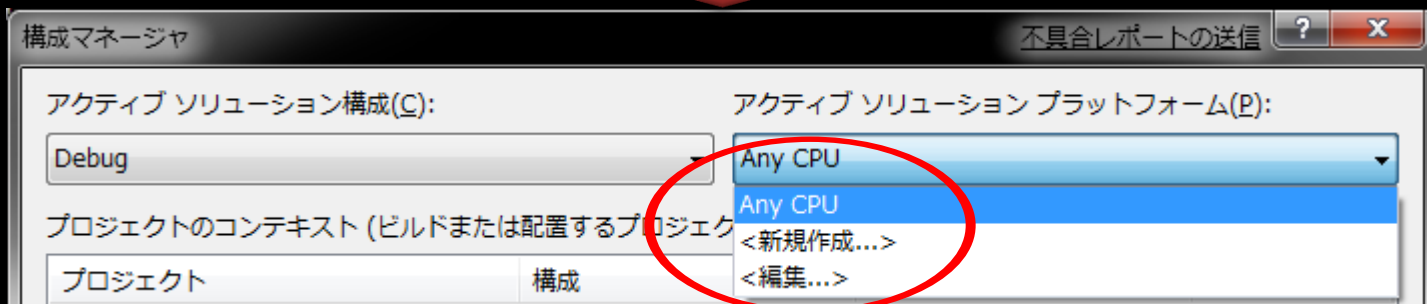
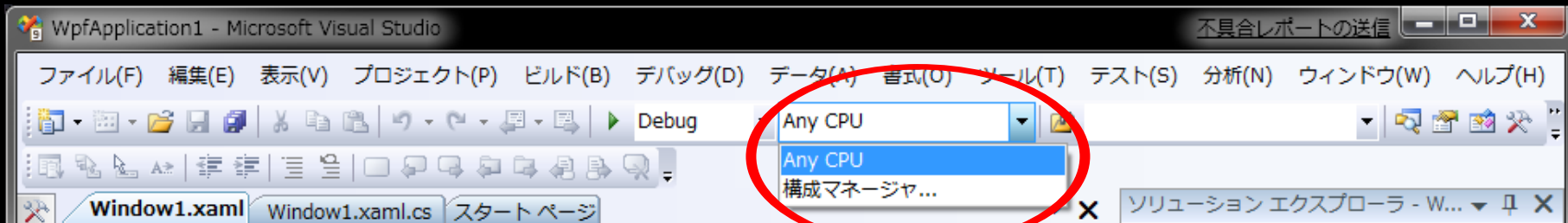
32bit  
Native

## ◎ 開発言語別のサポート状況

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

✓ : サポートする / ✗ : サポートしない

# ターゲットプラットフォームの選択



ソリューションプラットフォーム

- X86 : 32bit CPU
- X64 : x64(Intel 64 / AMD64)
- Itanium : Itanium
- Any CPU : 任意のCPU

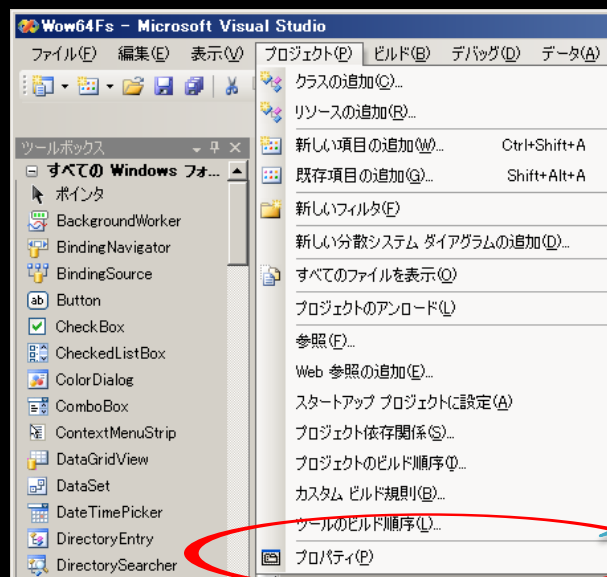
64bit  
.NET

32bit  
.NET

# ターゲットプラットフォームの選択 ～ C++/CLI

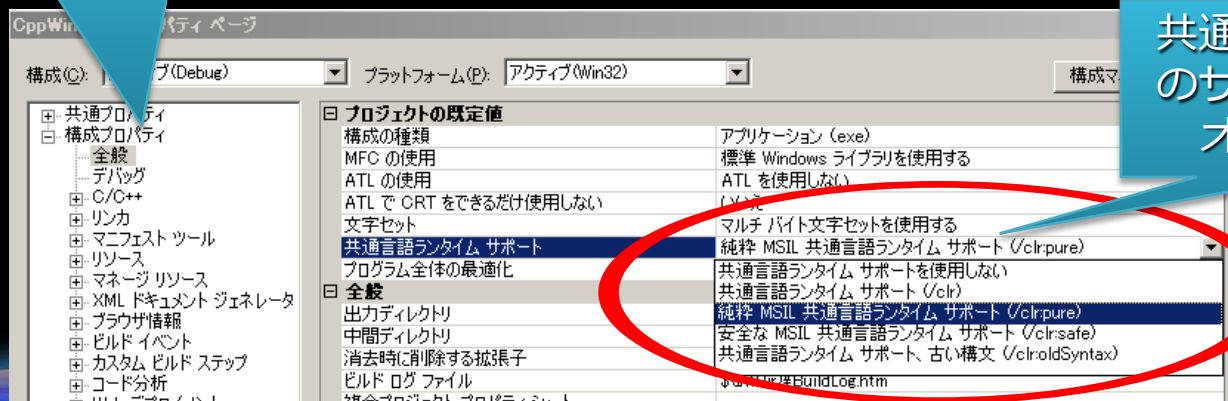
64bit  
.NET

32bit  
.NET



プロジェクトのプロパティ

構成プロパティ - 全般



共通言語ランタイム  
のサポートに関する  
オプション /clr

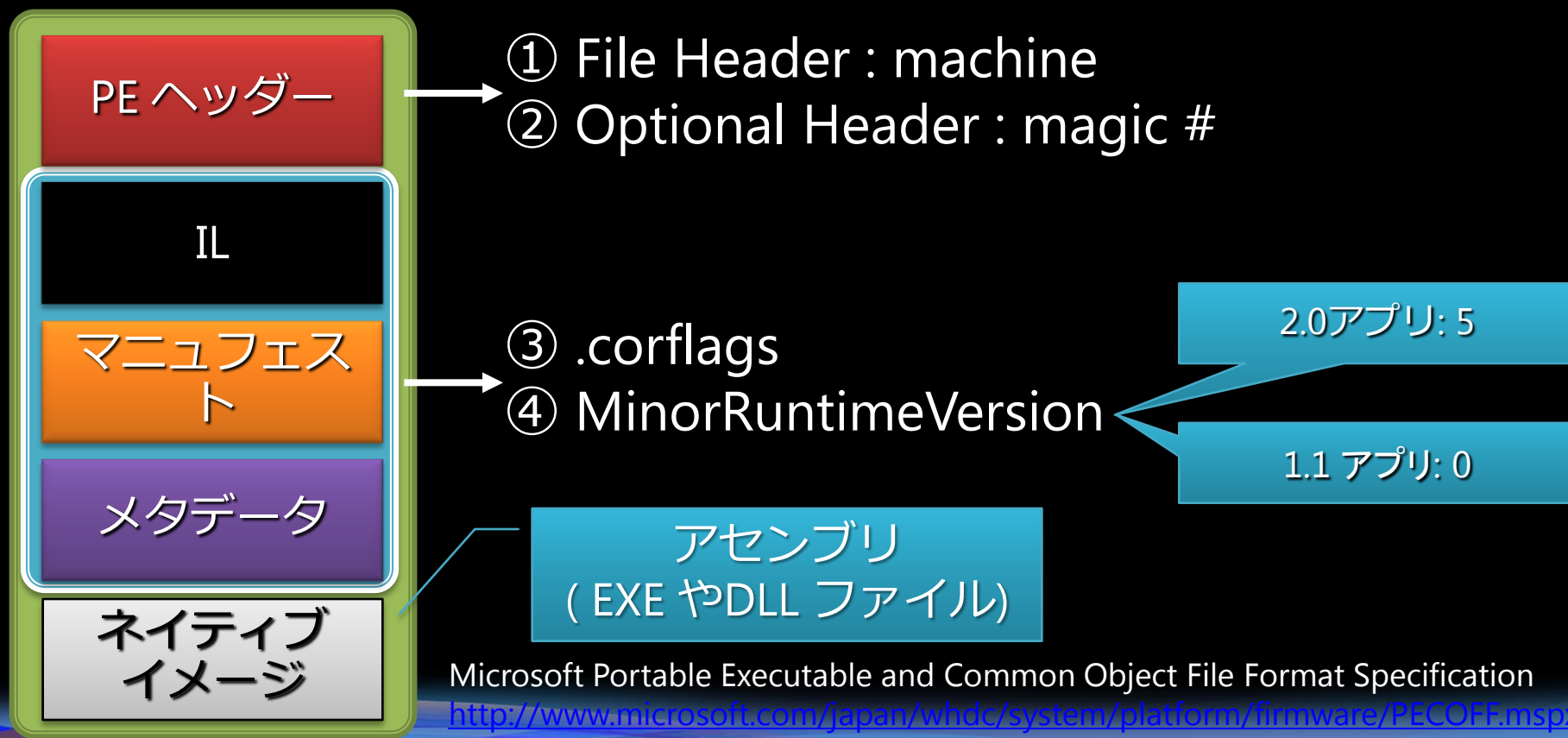


# 実行は 32bit or 64bit ?

64bit  
.NET

32bit  
.NET

- ◎ PEヘッダー、CLRヘッダーに記述により決定される

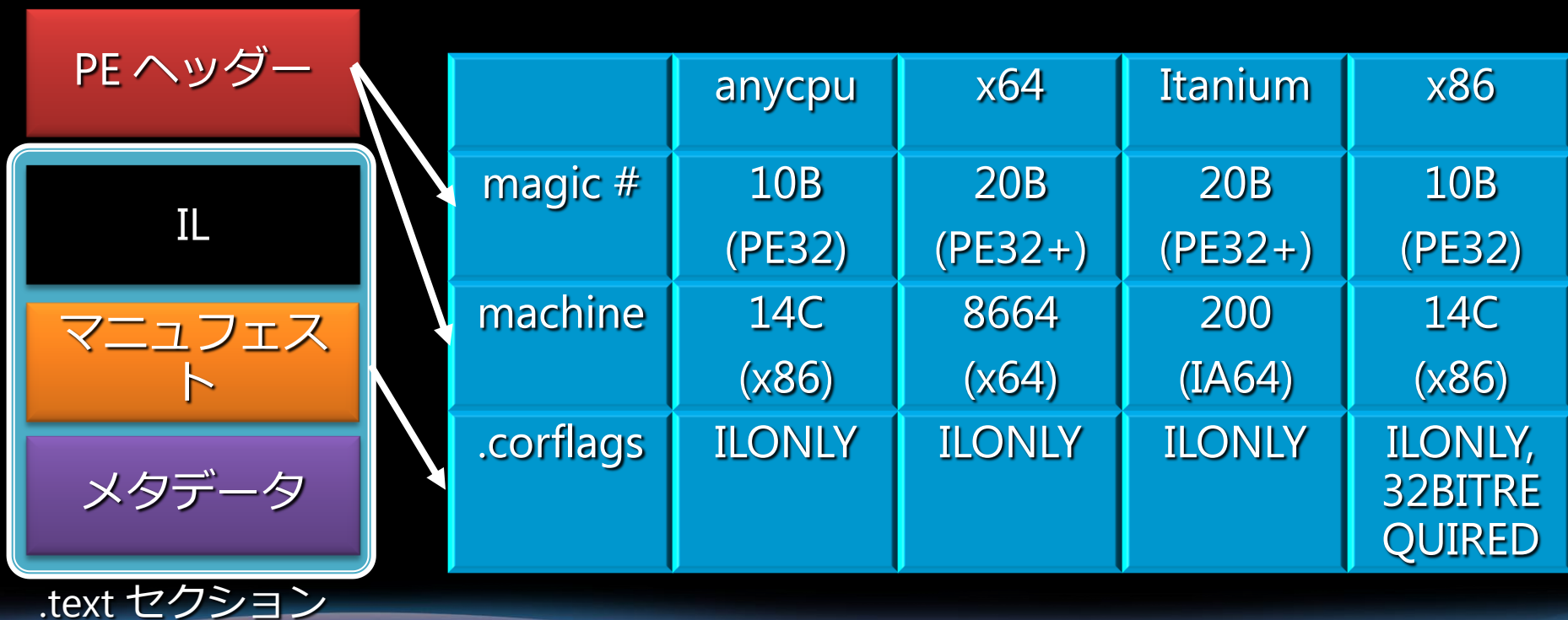


# アセンブリの構造

64bit  
.NET

32bit  
.NET

- ◎ コンパイルオプションはPEヘッダー、MSILにターゲットに関連する情報を付与
  - ◎ C#、VB でコンパイル /platform オプション



# アセンブリの構造

64bit  
.NET

32bit  
.NET

- ◎ コンパイルオプションはPEヘッダー、MSILにターゲットに関連する情報を付与
  - ◎ C++/CLI でコンパイル



※ ソリューションプラットフォームを x64  
に設定

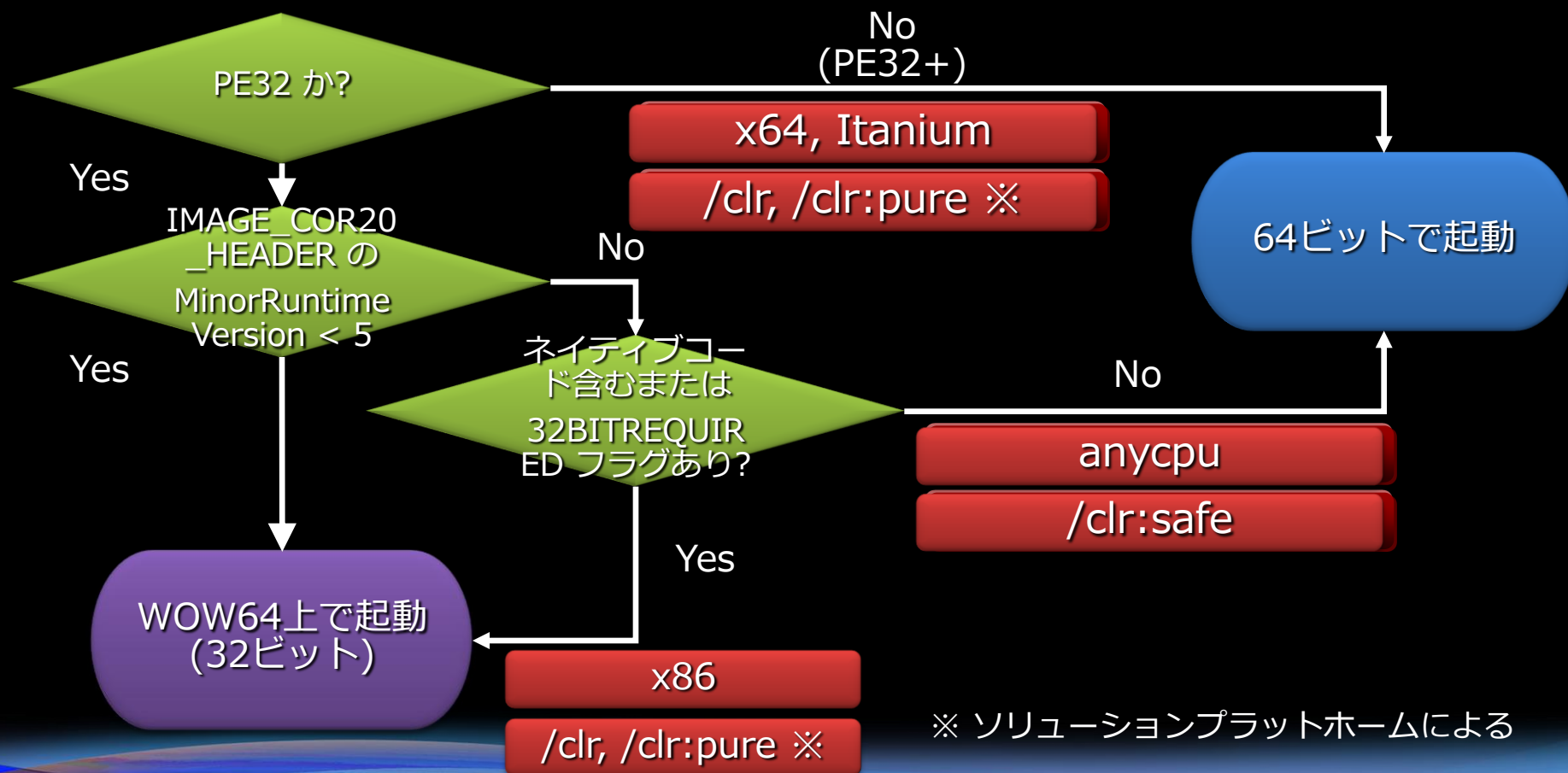
# OS ロードの動作

64bit  
.NET

32bit  
.NET

## ◎ OSロードがアプリケーションの起動を判別

- WOW64上での起動 or 64bit CLR での起動



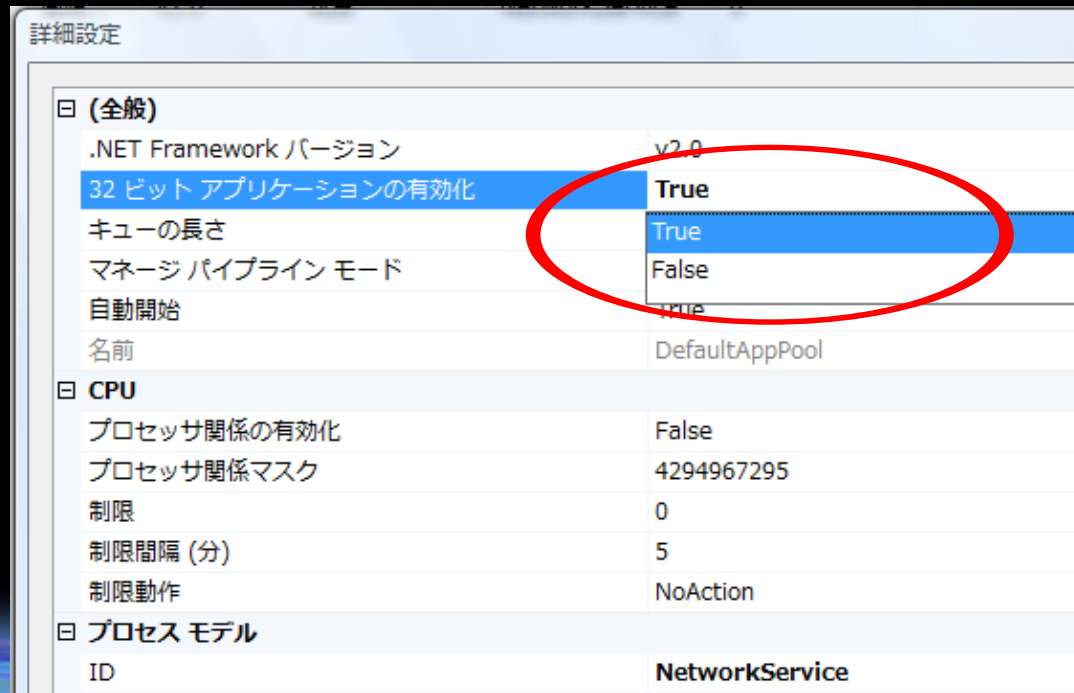
※ ソリューションプラットフォームによる

# 64bit Windows上での IIS 7.x

64bit  
.NET

32bit  
.NET

- ◎ アプリケーションプール毎に32bitか64bitを選択可能
  - 32bitアプリケーションの有効化 を設定します
  - False (Default) / True



# 64bit Windows 上での IIS 7.x

64bit  
.NET

32bit  
.NET

- ◎ IIS マネージャー 以外での設定方法
  - applicationHost.config での設定方法

```
<applicationPools>  
  <add name="MyAppPool32bit" autoStart="true" enable32BitAppOnWin64="true" />  
  <add name="MyAppPool64bit" autoStart="true" enable32BitAppOnWin64="false" />  
  <applicationPoolDefaults>  
    <processModel identityType="NetworkService" />  
  </applicationPoolDefaults>  
</applicationPools>
```

```
appcmd apppool set /apppool.name:MyAppPool32bit /enable32BitAppOnWin64:true  
appcmd apppool set /apppool.name:MyAppPool32bit /enable32BitAppOnWin64:false
```

※ default は false



64bit  
.NET

32bit  
.NET

# マネージド コード ～ここまでのまとめ

- ◎ ネイティブ、マネージコードの  
64/32bitアプリケーション開発が可能です
- ◎ マネージコード
  - 64/32bitはソースおよびバイナリ互換となります
    - ・ オプションで x86, x64, Itanium のターゲット指定可能です
  - C#、VB、C++/CLI
- ◎ 実行環境
  - コンパイルオプションにより32/64bit/WOW64 の動作が決定されます
  - 同一プロセスに別アーキテクチャのアセンブリロードは 不可となります
  - GAC の検索順序 特定プロセッサ -> MSIL -> 1.1
  - ASP.NET は アプリケーションプール毎に 32bit or 64 bit での動作 を指定できます

# コーディネーティングの留意点

# コーディングの留意点

64bit  
.NET

32bit  
.NET

64bit  
Native

32bit  
Native

## ◎ 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:  $\text{Sin}(1e37)$

# 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 (dllhost.exe) の利用

64bit  
.NET

32bit  
.NET

64bit  
Native

32bit  
Native

# コード例

## ～ SequentialLayout 属性

64bit  
.NET

32bit  
.NET

64bit  
Native

32bit  
Native

### 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

64bit  
.NET

32bit  
.NET

64bit  
Native

32bit  
Native



# コーディングの留意点 まとめ

64bit  
.NET

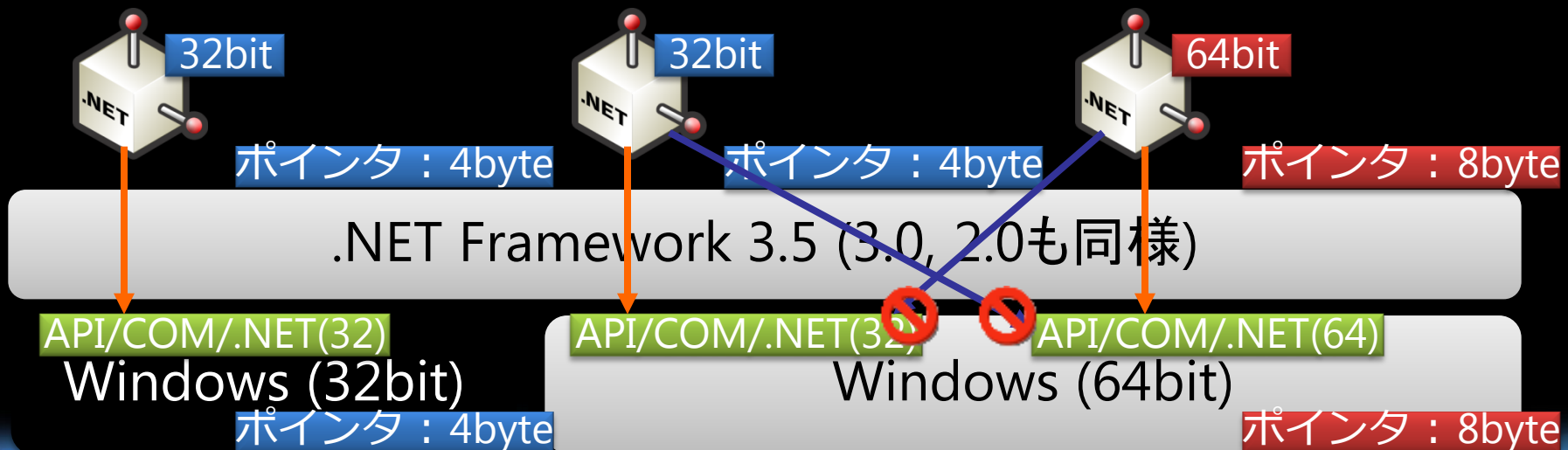
32bit  
.NET

64bit  
Native

32bit  
Native

- ◎ ポインタのサイズ - System.IntPtr
  - サイズの違に注意する
    - 32bit : 4バイト / 64bit : 8バイト
- ◎ OSの機能 - P/Invoke, COM introp
  - 64<->32 アーキテクチャ越えに注意する
  - データレイアウト(アライメント)が異なる

WOW64 (32bit)



The slide features a black background with decorative wavy lines in shades of blue and green at the top and bottom edges. The text 'まとめ' is centered in the upper half of the slide.

まとめ

# まとめ



Windows Server® 2008 R2  
Enterprise



Windows® 7

- ◎ Windows Server 2008 R2 では 64bit バージョンのみをリリースする
- ◎ 64bit Windows は 64/32 bit アプリケーションに安全な実行環境を提供している
- ◎ 動作プラットフォームを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の両立を検討する
- ◎ .NET Framework アプリケーションは 64bit 環境への対応を Framework の機能により吸収することが可能である

# 技術情報リソース

64 ビット アプリケーション:

<http://msdn.microsoft.com/ja-jp/library/ms241064.aspx>

Visual Studio 開発環境の 64 ビット サポート:

<http://msdn.microsoft.com/ja-jp/library/ms241066.aspx>

Programming Guide for 64bit Windows(英語):

[http://msdn.microsoft.com/en-us/library/bb427430\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb427430(VS.85).aspx)

WOW64 のベスト プラクティス:

[http://www.microsoft.com/japan/whdc/system/platform/64bit/wow64\\_bestprac.aspx](http://www.microsoft.com/japan/whdc/system/platform/64bit/wow64_bestprac.aspx)

Microsoft Windows SDK:

<http://msdn.microsoft.com/ja-jp/windows/bb980924.aspx>

.NET Framework 1.1/3.5 移行ホワイトペーパー :

[http://download.microsoft.com/download/5/5/b/55b34c74-2b8f-423c-94ad-4e22c0ae60f0/netFramework\\_migration\\_1.1\\_to\\_3.5.pdf](http://download.microsoft.com/download/5/5/b/55b34c74-2b8f-423c-94ad-4e22c0ae60f0/netFramework_migration_1.1_to_3.5.pdf)

# 技術情報リソース

Everything You Need To Know To Start Programming 64bit Windows Systems(英語):

<http://msdn.microsoft.com/en-us/magazine/cc300794.aspx>

Microsoft Portable Executable and Common Object File Format Specification(英語):

<http://www.microsoft.com/whdc/system/platform/firmware/PECOFF.mspx>

An In-Depth Look into the Win32 Portable Executable File Format(英語):

<http://msdn.microsoft.com/en-us/magazine/cc301805.aspx>

An In-Depth Look into the Win32 Portable Executable File Format, Part 2(英語):

<http://msdn.microsoft.com/en-us/magazine/cc301808.aspx>



参考：アプリケーションの  
ローディング



# 実行は 32bit or 64bit ?

- ◎ 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

64bit  
.NET

32bit  
.NET

64bit  
Native

32bit  
Native

	32bit OS	64bit OS		
	32bit	WOW64	64bit(x64)	64bit(IPF)
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

# アプリケーションローディング

- ◎ マネージコード自身はプロセッサアーキテクチャを認識しない
  - アセンブリ参照情報にはプロセッサアーキテクチャが含まれない

IL

マニフェスト

メタデータ

```
// Metadata version: v3.5
.assembly extern mscorlib
{
    .publickeytoken = (B7 7A 5C 56 19 34 E0 89 )
    .ver 2:0:0:0
}
.assembly extern PresentationFramework
{
    .publickeytoken = (31 BF 38 56 AD 36 4E 35)
    .ver 3:0:0:0
}
```

64bit  
.NET

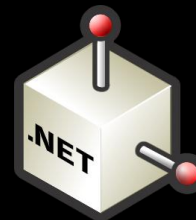
32bit  
.NET

# アプリケーションローディング

- ◎ プロセッサアーキテクチャに基づいた Assembly.Load は不可
  - ロード時に例外が発生

```
Assembly SampleAssembly;  
SampleAssembly = Assembly.LoadFrom( "c:¥¥Mydotnet.dll" );  
MethodInfo Method =  
    SampleAssembly.GetType()[0].GetMethod("MethodA");  
ParameterInfo[] Params = Method.GetParameters();
```

64bit windows



Mydotnet.dll x64

Mydotnet.dll x86

64bit  
.NET

32bit  
.NET

64bit  
Native

32bit  
Native

# 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 32bitコードと 64bit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



64bit  
.NET

32bit  
.NET

64bit  
Native

32bit  
Native

# アプリケーションローディング anyCPU

64bit  
.NET

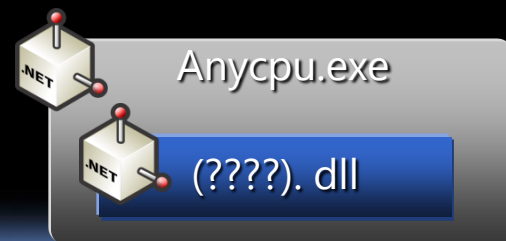
32bit  
.NET

- 複数のアセンブリでアプリケーションは構成されている  
場合が多い

		32bit OS		64bit OS	
		32bit	WOW64	64bit	
any.exe	any.dll	○	×	○	
	x86.dll	○	×	×	
	x64.dll	×	×	○	
	11.dll	○	×	○	

参照されているアセンブリ  
のイメージが生成されてい  
るプロセスと一致しないた  
め

OSローダが any.exe を読  
み込んだとき、64ビット  
のプロセスを生成するため





# アプリケーションローディング

## x64

64bit  
.NET

- 複数のアセンブリでアプリケーションは構成されていることが多い

		32bit OS	64bit OS	
		32bit	WOW64	64bit
x64.exe	any.dll	×	×	○
	x86.dll	×	×	×
	x64.dll	×	×	○
	11.dll	×	×	○

OSローダが PE32+ に対し64bitプロセスを生成できないため

OSローダが x64.exe を読み込んだとき、64bitのプロセスを生成するため

参照されているアセンブリのイメージが生成されているプロセスと一致しないため



Anycpu.exe



(????).dll



# アプリケーションローディング x86

32bit  
.NET

- 複数のアセンブリでアプリケーションは構成されていることが多い

		32bit OS	64bit OS	
		32bit	WOW64	64bit
x86.exe	any.dll	○	○	×
	x86.dll	○	○	×
	x64.dll	×	×	×
	11.dll	○	○	×

参照されているアセンブリのイメージが生成されているプロセスと一致しないため

OSロードが x86.exe を読み込んだとき、32bitのプロセスを生成するため



Anycpu.exe



(????).dll

# アプリケーションローディング

## GAC

64bit  
.NET

32bit  
.NET

- ◎ GAC (fusion.dll) はアーキテクチャを認識します
  - 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:
    - ・ 特定プロセッサー (x86, x64, IPF)
    - ・ MSIL
    - ・ Legacy (1.1)

WOW64  
① x86  
② MSIL  
③ 1.1

64bit(x64)  
① x64  
② MSIL  
③ 1.1

◎ gacutil /lr

アセンブリ名	バージョン	カル.	公開キー トークン	プロセッサのアーキテクチャ
MSBuild	3.5.0.0		b03f5f7f11d50a3a	x86
MSBuild	3.5.0.0		b03f5f7f11d50a3a	AMD64
mscorlib	2.0.0.0		b77a5c561934e089	x86
mscorlib	2.0.0.0		b77a5c561934e089	AMD64
mscorlib.resources	2.0.0.0	ja	b77a5c561934e089	MSIL
MSDATASRC	7.0.3300.0		b03f5f7f11d50a3a	
mscorlib	6.1.0.0		31bf3856ad364e35	x86
mscorlib	6.1.0.0		31bf3856ad364e35	AMD64
mscorlib	6.1.0.0		31bf3856ad364e35	x86
mscorlib	6.1.0.0		31bf3856ad364e35	AMD64
mscorlib	6.1.0.0		31bf3856ad364e35	MSIL





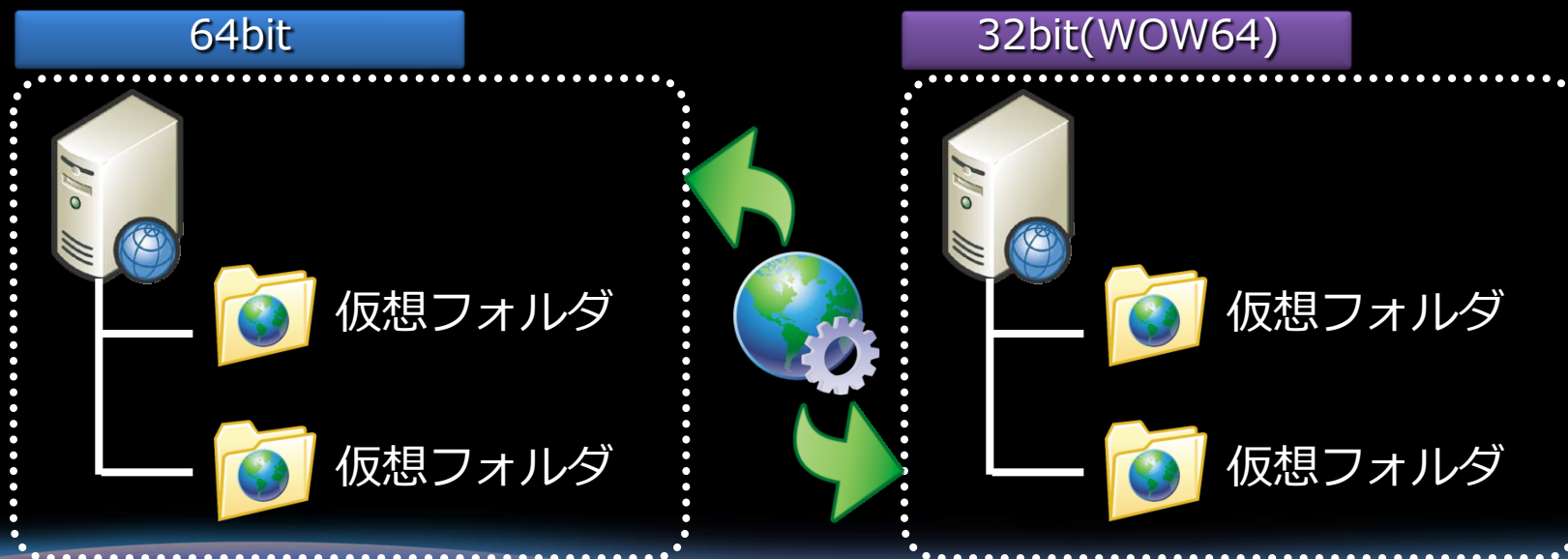
参考：IIS 6.0の設定

# 64bit上でのIIS 6.0

64bit  
.NET

32bit  
.NET

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



# 64bit上でのIIS 6.0

64bit  
.NET

32bit  
.NET

## ◎ 32bitワーカープロセスへの切り替え

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

### > 32bit ASP.NET 3.5 のインストール

```
%SystemDrive%\Microsoft.net\Framework\v3.5\aspnet_regiis -i -wow64 -enable
```

### > 32bit ASP.NET 2.0 のインストール

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

マッピング



32bit(WOW64)



仮想フォルダ

# 64bit上でのIIS 6.0

64bit  
.NET

32bit  
.NET

## ◎ 64bitワーカープロセスへの切り替え

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

### > 64bit ASP.NET 3.5 のインストール

```
%SystemDrive%\Microsoft.net\Framework64\v3.5\aspnet_regiis -i -enable
```

### > 64bit ASP.NET 2.0 のインストール

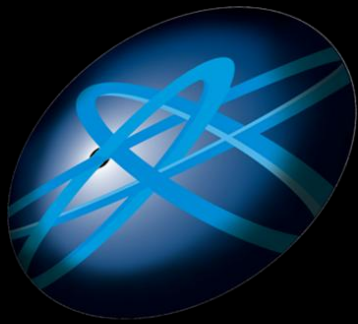
```
%SystemDrive%\Microsoft.net\Framework64\v2.0.50727\aspnet_regiis -i -enable
```







Windows® 7



# ***Future Technology Days***

***Technology Days***



**Microsoft®**

The Microsoft logo is centered on a black background. It features the word "Microsoft" in a white, bold, sans-serif font. The letters are closely spaced, and the registered trademark symbol (®) is positioned to the upper right of the final letter. The logo is flanked by decorative wavy lines in shades of blue and green at the top and bottom of the image.