本次课程内容包括

- IDE & Build System History
- Team Roles
- Data Flow
- Goals and best practice

收听本次课程需具备的条件

- 了解Windows CE/Mobile的目录安排
- 了解基本的源代码管理方式

Level 200

Brief History of Microsoft Platform Builder Integrated Development Environment (IDE)

- ## Before there was an IDE
  - Command-line based build system
  - Configure via environment variables (in BAT files)

- ## First IDE based on Visual Studio 6 Code
  - Parallel build system based on VS6 logic
  - Build results not quite the same as command-line

- ## Platform Builder 5.0, "Unified Build"
  - One build system – the command line
  - OS Designs (PBXML)
  - Subprojects (PBPXML)

Motivation for this Talk

- Encourage you to leverage the new build features
- Save you guesswork getting started
- Tell you what we intended

And...
- Illustrate the pain points so we can improve

Engineering Best Practices

- Source control used to manage files
  - Generated files/binaries are not checked in
- Integration centrally controlled
  - External components reviewed before accepting
  - Updates distributed to team via Source Control
- Reproducible build process
  - Regularly scheduled build-lab builds
  - Identical build results regardless of build machine
- Regularly scheduled test passes

Team Roles

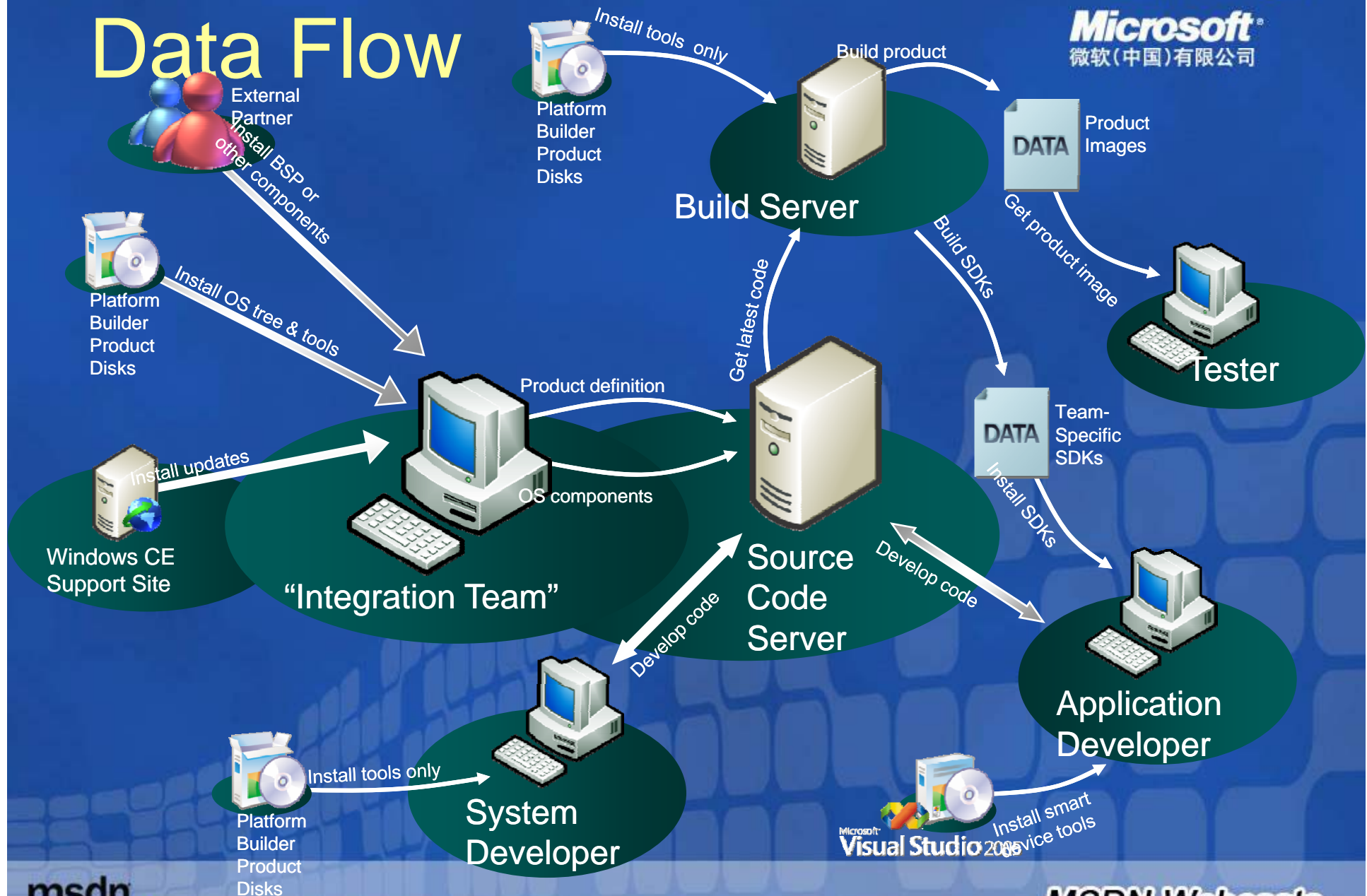- Integration team (or team lead) for the project
- Individual software developers
  - System-level developer
  - Application-level developer
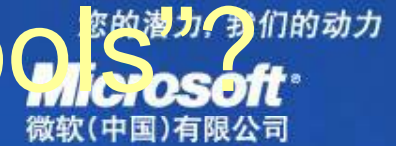- Testers
- A build lab

Windows CE Products

- Platform Builder "Tools"
  - OS Design and Visualization Tools
  - Kernel Debugger
  - Code and Resource Editors
  - Connectivity and remote tools
- Platform Builder "OS Tree"
  - Windows CE OS Code
  - Build System
- Visual Studio 2005
  (Standard, Professional, or Team System)
  - Application development tools for Windows CE

Who Should Install the "Tools"?
Platform Builder development tools

- Integration team
  - For defining and managing the product
  - For reviewing and accepting updates
- System-level developers
  - Coding and building
  - Kernel debugging
- Build lab
  - Needed to build OS designs

Pain point

msdn

MSDN Webcasts

Application Developer
Should you install Platform Builder?

- ## Visual Studio's smart device projects
  - Requires an SDK to be created ("rolled") by the OS Design team
  - CE Application development in C++ or C#
  - Does not require Platform Builder
  - Better application development design tools
  - Faster and easier development cycles, especially using the emulator

Pain point

- SDK generation and installation is manual and tedious
- Application software must be imported into the OS Design as binary files
  - NK.BIN build image cannot be created from source in one pass

Who Should Install the "OS Tree"?
Windows CE OS components and build system

- Integration team
  - The "gateway" between the outside world and the team's source code database

Bootstrapping the Project

- Install Platform Builder "Tools" and "OS Tree"
- Check OS Tree into source control
- Clone BSPs you expect to modify
- Define your product's features
  - OS Design wizard – Choose appropriate template
  - Choose one or more BSPs as appropriate
  - Customize the OS Design
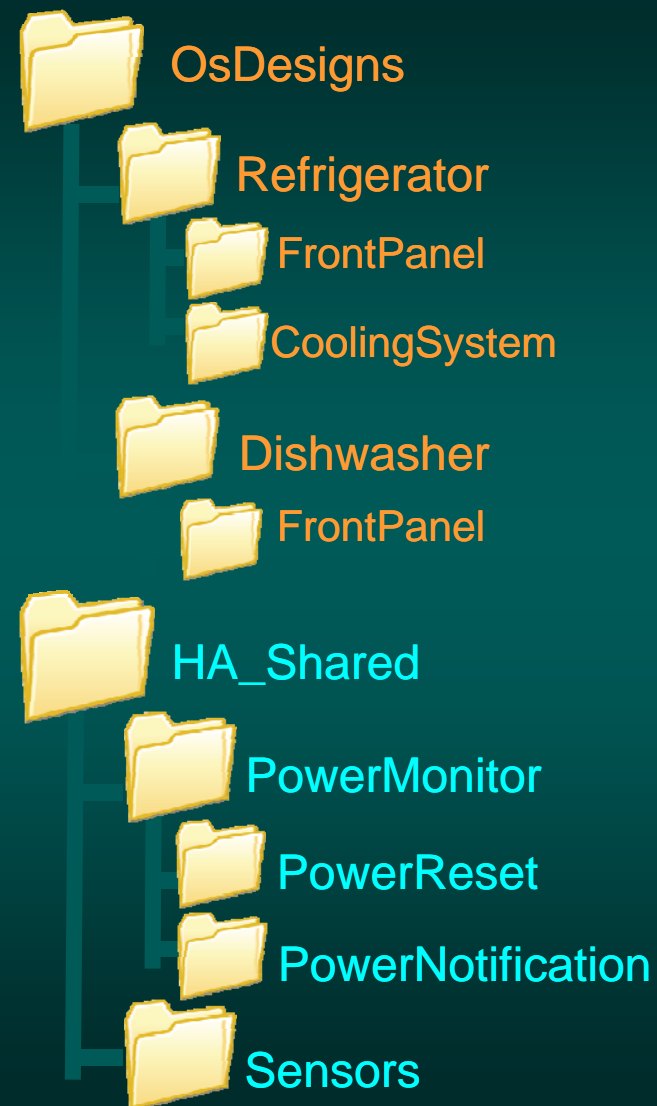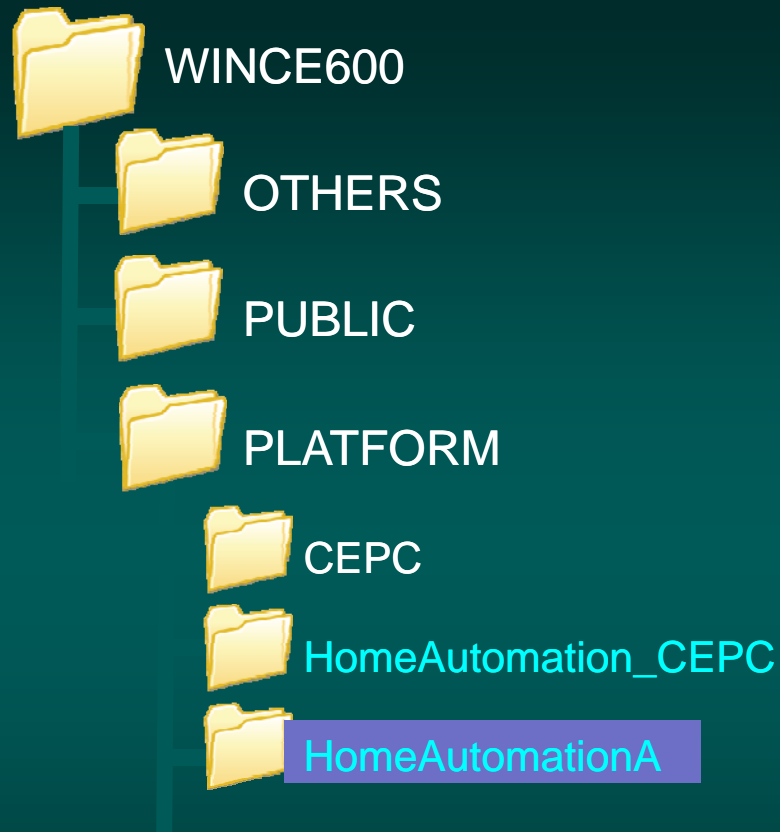 - Check OS Design into source control

Sample Source Code Structure
"Home Automation" company

📁 WINCE600
  📁 OTHERS
  📁 PUBLIC
  📁 PLATFORM
    📁 CEPC
    📁 HomeAutomation_CEPC
    📁 HomeAutomationA

📁 OsDesigns
  📁 Refrigerator
    📁 FrontPanel
    📁 CoolingSystem
  📁 Dishwasher
    📁 FrontPanel
  📁 HA_Shared
    📁 PowerMonitor
    📁 PowerReset
    📁 PowerNotification
    📁 Sensors

KEY:
MICROSOFT
Shared
Project-specific

Build-related Configuration Files

- OS Design configurations (PBXML)
  - Defines build configurations and product components
  - References to Subproject files (PBPXML)
- Subproject configurations (PBPXML)
  - Minimal metadata information
  - Sibling BIB and REG files to get components into OS Design
  - Sibling SOURCES & DIRS still the source of build information
- SOURCES files
  - Primary source of project build information
  - Used in the OS Tree and in Subprojects
- DIRS files
  - Hierarchical information

Smart Device Projects
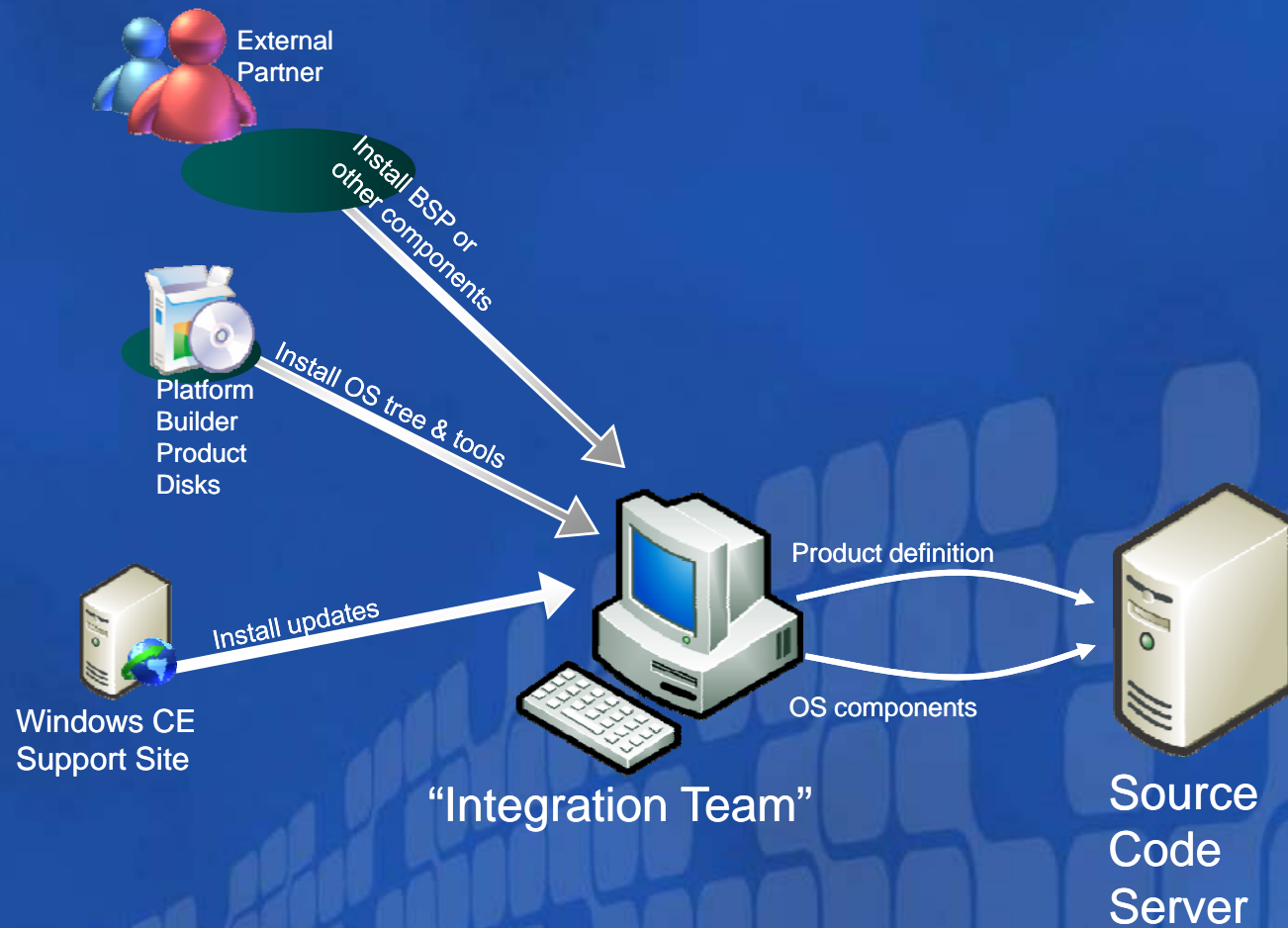## Integrating applications into the OS design

- Add files as pre-built binaries   Pain point
  - Build Smart Device project with latest internal SDK
  - Check in compiled binaries to OS Design
  - Add supporting PBPXML file, BIB, and REG files
  - Power Tool: CEFileWiz – Mike Hall's Blog

Installing Updates

- **Fundamental goals**
  - Get latest OS and driver fixes from Microsoft or BSP vendors
  - Have a process to review and accept changes into the project

- **Best practice for accepting an update**
  - Integration team installs the QFE onto a integration machine
    - QFE may come from Microsoft or BSP Provider
    - http://msdn.microsoft.com/embedded/downloads/ce/wince/default.aspx
  - Code changes are reviewed
  - One or more builds are generated on the integration machine
  - The resulting build images are tested for regressions
  - Accept/reject and check into the source code database
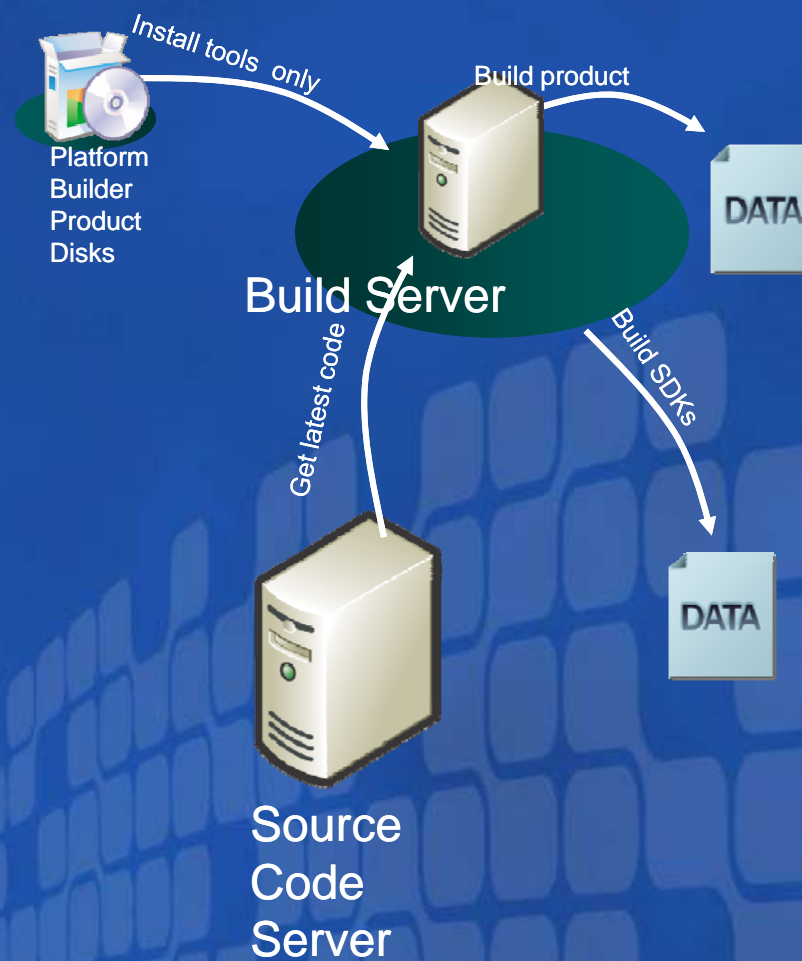
Installing Updates

Pain point

– QFE will overwrite any team changes to MS code

  • "Catalog Item Cloning" (5.0 & later) to prevent this scenario

– DIRS files may need merging

  • DIRS=* added in latest CE

Build Lab

Build Lab Setup

- Fundamental goals:
  - Have an automated process for building
  - Can be run from scripts
  - Install the minimum set of tools needed to build
- Lab machine initial setup
  - Windows XP Professional with latest Service Packs
  - A source code control client

Pain point   Install Platform Builder tools

- Required for building PBXML OS Designs and Subprojects
- Requires a "Tools" Install (including Visual Studio 2005 in latest Platform Builder)

Automated Build Script Sample
Outer script

```
REM Build Machine Should be setup per previous requirements

REM Set Global variables for this machine
Set WINCEROOT=c:\WINCE600
Set TMP=C:\Temp

REM Delete all previous files
rmdir %WINCEROOT% /s /q

REM Synchronize latest code and OS Tree
REM Use appropriate command for your source control client

REM Launch each configuration to build in a new environment
Call MyOSBuild.bat MyOSDesign1 "CEPC x86 Debug"
Call MyOSBuild.bat MyOSDesign1 "CEPC x86 Release"
Call MyOSBuild.bat MyOSDesign1 "Device Emulator ARMV4I Debug"
Call MyOSBuild.bat MyOSDesign1 "Device Emulator ARMV4I Release"
```

Automated Build Script Sample
Leveraging PbxmlUtils.exe

```
REM MyOsBuild.bat <OSDesign> <Configuration>
REM WINCEROOT and TMP must be set prior to calling this file

REM Make build environment variable settings local
Setlocal

REM Create a BAT file for setting build environment
rm %TMP%\%1.bat
Set PBWORKSPACE=%WINCEROOT%\OSDesigns\%1\%1.pbxml
Pbxmlutils.exe /config %2 /getbuildenv > %TMP%\%1.bat

REM Set build environment and do the build
Call %TMP%\%1.bat
Blddemo -q

endlocal
```
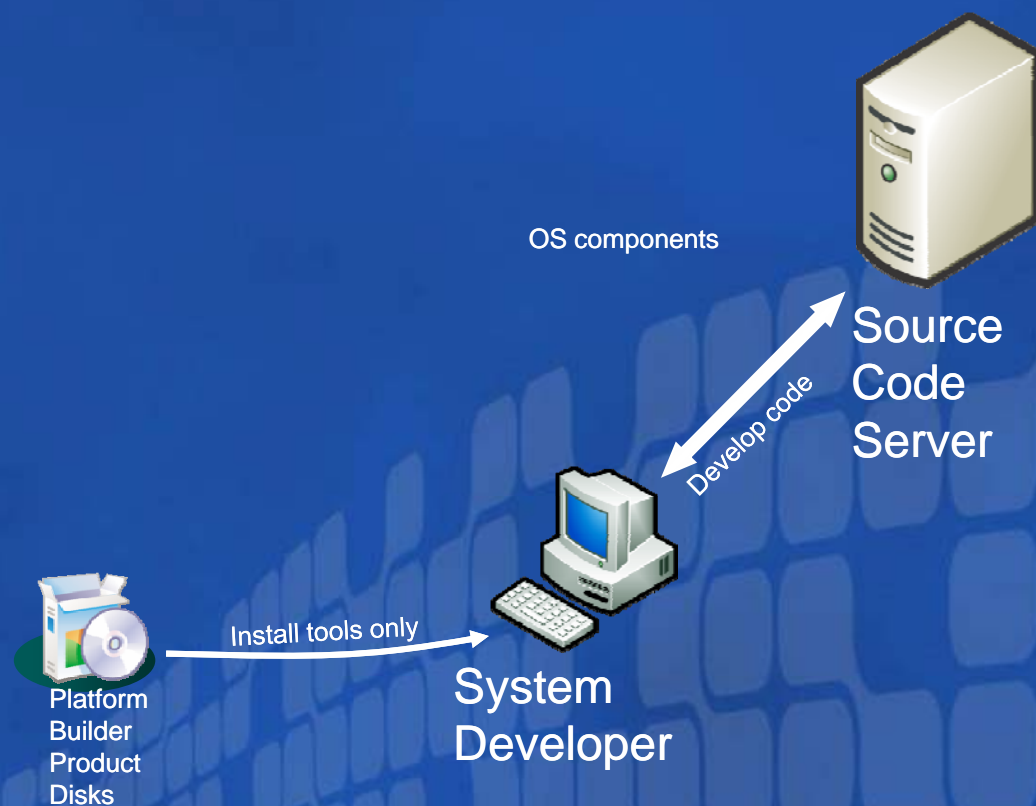
System Developer

System Developer Setup
Platform Builder Tools installation

Fundamental goals

- – Install the Platform Builder development tools on a workstation

Platform Builder installation

- – For versions after 5.0, install Visual Studio 2005 first
- – Choose "Tools Only" Installation of Platform Builder

OS Tree installation

- – Synchronize from your team source control database
- – Set the WINCEROOT to your OS Tree
    - Menu: Tools / Options / Build

Get OS Tree into Source Control

- Fundamental goals
  - Get all files and build tools that affect the build into source control
  - Ensure that all developers, testers, and build labs can produce identical results

- Platform Builder installation
  - Install "Tools" and "OS Components"
  - Install OS components to a clean location
  - Install only the CPUs for your final product
    - Alternate: If you have Application Developers, also install the ARM CPU used by the emulator

Get OS Tree into Source Control

- Optional: additional BSP installation
  - Files provided by a BSP provider for their custom hardware
  - The ARM emulator BSP for CE 5.0 is available at
    http://blogs.msdn.com/barrybo/archive/2006/03/15/552045.aspx
    - Allows you to create an application development SDK that can be used with Visual Studio 2005
- Do not build anything until you have added to source control!
- Check all OS tree files into Source Control
  - Everything under C:\WINCE600 even LIBs, DLLs, EXEs, etc.

Cloning a Board Support Package

- Fundamental goals
  - Copy BSP code so you can modify it for your needs but keep original intact

- BSP cloning
  - Menu: Platform / BSP Wizard… (5.0)
  - Streamlined tool planned for latest CE

Defining Your Product
OS Design creation

- Fundamental goals
  - Define settings for team-wide OS Designs for products you will ship
  - Define settings for emulation to enable application developers the to develop without hardware

- Creating a new OS Design
  - Run the "Platform Wizard" (5.0 and earlier)
  - In Visual Studio 2005 with Platform Builder: File / New / Project…
    - Project Type = Platform Builder

Defining Your Product
OS Design creation

- Choose BSPs for OS Design
  - Choose BSP for your hardware
  - Optional: Also choose Emulator to enable application development without hardware
  - Optional: Also choose CEPC if you want an alternative hardware platform
- Choose initial design template
  - A convenient starting point only
- Select optional components

Add OS Design to Source Control
Which files to add

MyDesign\MyDesign.pbxml

– XML file that defines the project and configurations

MyDesign\WINCE600 directory and all children

– A copy of files from Public\CEBASE

– Parameter file customization for your OS Design (BIB & REG)

MyDesign\MyDesign.sln

– Optional: Defines the Visual Studio "Solution"

Do Not Add: MyDesign.pbxml.user, MyDesign.suo

– These are user preference files

Do No Add: RelDir directory or children

– These are build result files

Defining an SDK

Fundamental goals

- – Create an SDK for your OS Design
- – Enables use of the simpler and less-cumbersome Visual Studio 2005 Smart Device development tools

Define an SDK based on your OS Design

- – 5.0 and earlier – See Help Topic
  - • "How to Create an SDK for a Custom OS Design"
- – Latest Windows CE
  - • Not yet in Beta – Simplifying the Process

Creating a Shared Code Hierarchy
PBCXML projects outside the OS Design

Fundamental goals
- Create a code tree to be shared between OS Designs

Create the root of the code tree
- Create the root directory for the project
- Create a text file called DIRS with Notepad
  - Add a single line "DIRS=" to the file and save it
- BIB, REG & DB files for hierarchy located in root
- From any OS Design:
  - Menu / Project / Add Existing Subproject…
  - Browse to location of DIRS file and select it
- You can now "Add New Subproject…" under this folder
  - Creates a PBPXML file for this hierarchy

Pain point

Checking in PBPXML Subprojects
Which files to add

Do not add the following files to source control
- The obj directory and children
- Build.dat, build.err, build.log, build.wrn
- Any files ending with ".user"

All other files should be added to source control
- Files with extensions BIB, REG, DAT, DB, BAT, PBPXML
- SOURCES or DIRS files
- And of course any code files

Application Developer Setup
Visual Studio 2005 with Smart Device support

Fundamental goals

– Install the Visual Studio 2005 application development tools on a workstation

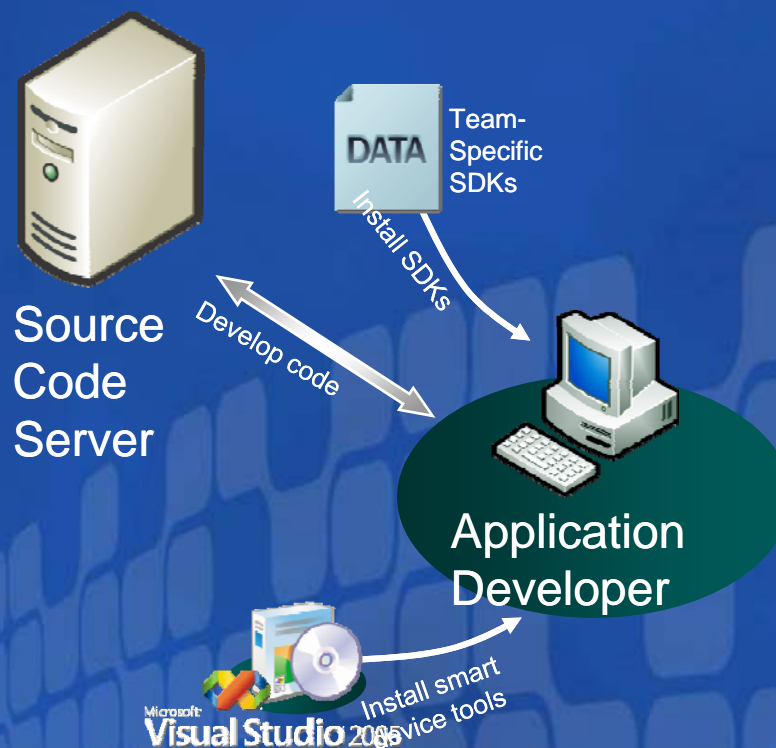– Be able to develop applications that will run on your product

Visual Studio installation

– Install Visual Studio 2005 Standard, Professional, or Team System

• Include Smart Device development support

– Install the latest OS Design SDK provided by your team

Application Developer

Source Code Server

DATA

Team-Specific SDKs

Install SDKs

Develop code

Application Developer

Microsoft Visual Studio 2005

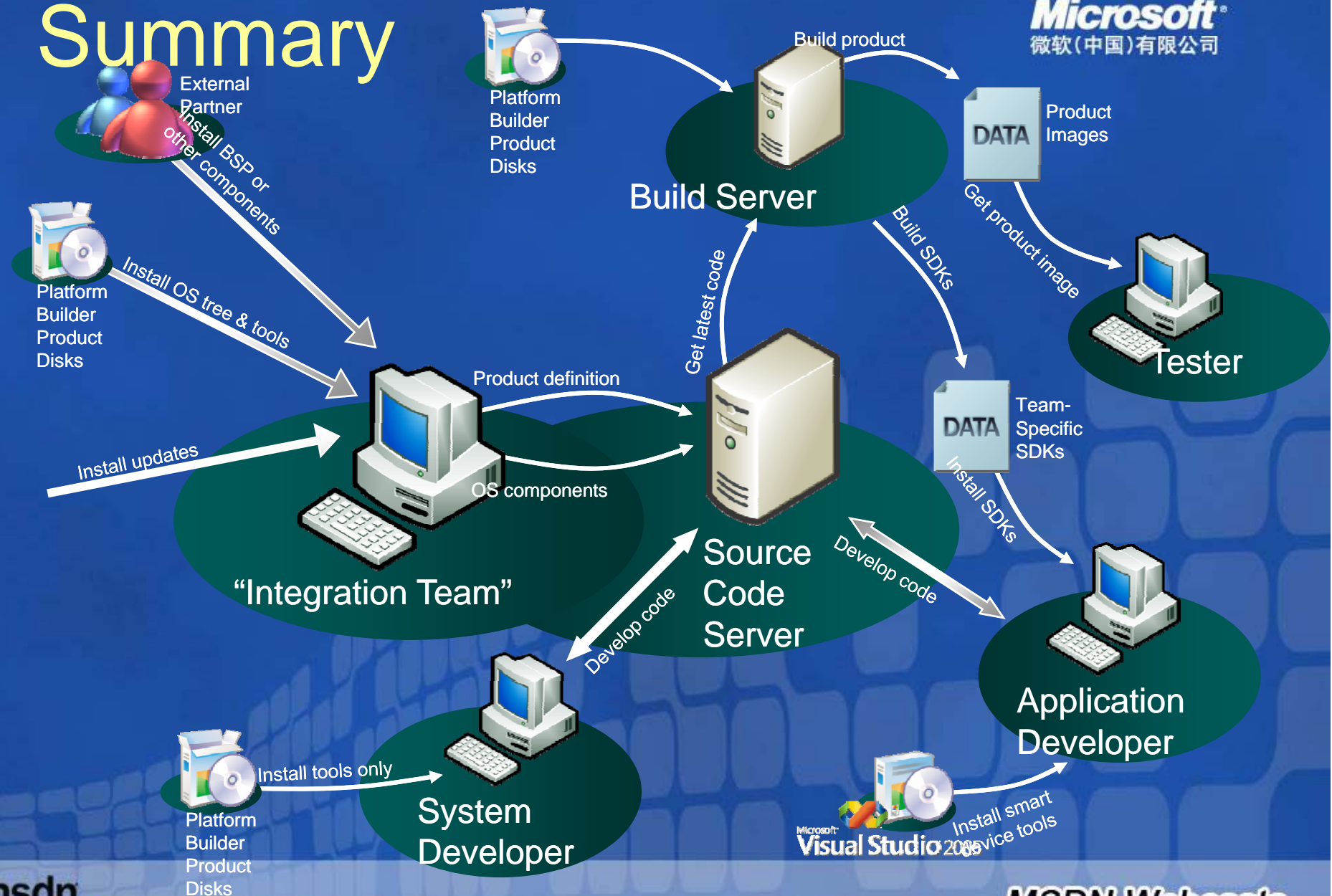Install smart device tools

Smart Device Projects
Advanced topic

- Multi-pass build required
  - Build OS Design
  - Export an SDK
  - Install SDK
  - Build the Smart Device Project
  - Add resulting binary to source control
  - Rebuild OS Design
- One-pass builds?

Pain point

Summary

External Partner

Install BSP or other components

Platform Builder Product Disks

Install OS tree & tools

Install updates

Product definition

OS components

"Integration Team"

Install tools only

Platform Builder Product Disks

System Developer

Build product

Platform Builder Product Disks

Build Server

Get latest code

Build SDKs

DATA Product Images

Get product image

Tester

Source Code Server

Develop code

DATA Team-Specific SDKs

Install SDKs

Develop code

Develop code

Application Developer

Install smart device tools

Visual Studio 2005

您的潜力，我们的动力

Microsoft®
微软(中国)有限公司

嵌入式开发资源

- **Windows Embedded中文官方网站**

  http://www.microsoft.com/china/windows/embedded

- **.NET Micro Framework**

  http://msdn2.microsoft.com/zh-cn/embedded/bb267253.aspx

- **Microsoft Robotics Studio**

  http://msdn2.microsoft.com/zh-cn/robotics/default.aspx

- 微软嵌入式开发者论坛

  http://forums.microsoft.com/china/default.aspx?ForumGroupID=493&SiteID=15

- 微软中国嵌入式开发者博客

  http://blogs.msdn.com/yunxu/

- **Mike Hall的博客**

  http://msdn2.microsoft.com/zh-cn/embedded/ Aa731228.aspx

微软启动新一轮"免费重考计划"

- 作为对微软认证学习者的支持与鼓励，微软公司于**2007**年**9**月 **15**日启动新一轮免费重考计划。
- 如果您以前曾参与过该项计划，那您或许已经了解到该计划将为您顺利通过微软认证考试带来有效的保障。在计划推行期内，如果您在考试前注册获得免费重考考试券，那当您首次考试未通过时，您将获得一次免费重考的机会。

msdn

MSDN Webcasts

获取更多**MSDN**资源

- **MSDN**中文网站
  http://msdn2.microsoft.com/zh-cn
- **MSDN**中文网络广播
  http:// www.microsoft.com/china/msdn/webcast
- **MSDN**免费中文速递邮件 **(MSDN Flash)**
  http://msdn2.microsoft.com/zh-cn/flash
- **MSDN**开发中心
  http://msdn2.microsoft.com/zh-cn/developercenters
- MSDN图书中心
  http://www.microsoft.com/china/msdn/book

Question & Answer

如需提出问题，请单击"提问"按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击"提问"按钮。

问题和解答（无问题）

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问

提问(A)　　删除(D)　　**问题管理器(Q)**

您的潜力, 我们的动力
Microsoft®
微软(中国)有限公司

Microsoft®
Your potential. Our passion.™

msdn

MSDN Webcasts